

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR MOHAMED ELYES BEN HAFSA

MODÉLISATION STATISTIQUE DES COLLECTIONS D'IMAGES EN
UTILISANT LES MODÈLES DE COPULES ET APPLICATION À
L'INDEXATION D'IMAGES

AUTOMNE 2011

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR:

M. Mohamed Lamine Kherfi, directeur de recherche

Département de mathématique et d'informatique de l'UQTR

M. Francois Meunier, juré

Département de mathématique et d'informatique à l'UQTR

M. Fathallah Nouboud, juré

Département de mathématique et d'informatique à l'UQTR

REMERCIEMENT

Tout d'abord, je remercie le Bon Dieu de m'avoir donné la force et la santé pour réaliser mes travaux à terme.

C'est avec le plus grand plaisir que j'exprime toute ma gratitude à mes directeurs de recherche, Monsieur Kherfi Mohamed Lamine et Monsieur Mesfioui Mhamed, pour l'encadrement qu'ils m'ont apporté durant la maîtrise. Leurs précieux conseils, leurs encouragements, leurs disponibilités et leur soutien moral ont été des facteurs très importants pour moi et ont joué un rôle crucial pour la réalisation de ce mémoire.

J'exprime aussi ma reconnaissance à toute ma famille pour leurs soutiens et leurs encouragements sans faille.

Je remercie également tous les professeurs ainsi que tous les membres du département de mathématique et d'informatique et en particulier Madame Chantal Guimond.

Table des matières

Remerciement.....	ii
Table des matières.....	iii
Table des figures.....	iv
Liste des abréviations.....	vi
Introduction générale.....	ix
1- Chapitre1 GÉNÉRALITÉ	1
1.1 Introduction	1
1.2 Différent mode de recherche d'images.....	2
1.2.1 Mode de communication entre l'utilisateur et le moteur de recherche	3
1.2.1.1 Recherche basée sur une requête	3
1.2.1.2 Recherche basée sur la navigation dans un catalogue	4
1.2.2 Type de recherche d'images.....	7
1.2.2.1 Recherche d'images basée sur le texte	7
1.2.2.2 Recherche d'images basée sur le contenu	11
1.3 Les étapes de la recherche d'images	14
1.3.1 Les caractéristiques des images	15
1.3.1.1 La couleur	15
1.3.1.1.1 L'histogramme de couleur	15
1.3.1.1.2 Les moments statistiques	17
1.3.1.2 La texture	17
1.3.1.3 La forme	18
1.3.2 Mesure de similarité	19
1.3.2.1 La distance euclidienne	19
1.3.2.2 La distance pondérée	20

1.3.2.3	La distance de Mahalanobis	20
1.3.3	L'indexation	21
1.4	Indexation	21
1.4.1	Qu'est ce que c'est que l'indexation	22
1.4.2	Les différentes étapes de la création d'un index	22
1.4.3	Les différentes techniques d'indexation.....	23
1.4.3.1	Indexation basée sur le texte.....	23
1.4.3.2	Indexation basée sur le contenu.....	24
1.5	Les domaines d'application.....	26
1.5.1	La criminologie	26
1.5.2	La recherche sur le Web.....	27
1.5.3	Le domaine médical	28
1.6	Conclusion.....	28
2-	Chapitre 2 ÉTAT DE L'ART	30
2.1	Introduction	30
2.2	Les techniques d'indexation.....	31
2.2.1	Les techniques d'arbre	31
2.2.1.1	KD-Tree.....	31
2.2.1.2	Le R-Tree	34
2.2.2	La technique des réseaux de neurones	36
2.2.2.1	Le neurone biologique.....	37
2.2.2.2	Le neurone formel	38
2.2.2.3	Les différentes méthodes de réseaux neurones utilisées pour la classification d'images	40
2.2.2.3.1	Les perceptrons multicouches.....	40
2.2.2.3.2	Les réseaux de Hopfield.....	42
2.2.3	La technique statistique.....	45
2.3	Quelques moteurs de recherche existants utilisant la méthode d'indexation	47
2.3.1	QBIC	47

2.3.2	Virage.....	48
2.3.3	VisualSeek	49
2.3.4	PhotoBook.....	49
2.4	Critique générale des méthodes d'indexation	50
2.5	Conclusion.....	52
3-	Chapitre1 NOTRE TRAVAIL.....	53
3.1	Introduction	53
3.2	La méthode de mixture de Gaussienne.....	54
3.2.1	Rappel statistique et principe de la méthode de classification basée sur le mélange gaussien	54
3.2.1.1	Distribution gaussienne multivariée	54
3.2.1.2	Distribution de mélange gaussien.....	55
3.2.1.3	Formule de Bayes	56
3.2.1.4	La méthode de Maximum de Vraisemblance	57
3.2.1.5	Le principe de la méthode de classification basée sur le mélange gaussien	57
3.2.1.6	L'algorithme de la méthode de classification basée sur le mélange gaussien	64
3.3	Critique de la GMM	65
3.4	Copules et quelques propriétés.....	69
3.4.1	Définition	71
3.4.1.1	Définition d'une copule.....	71
3.4.1.2	Quelques exemples de copules.....	72
3.4.1.2.1	Les bornes de Fréchet :	72
3.4.1.2.2	Copule d'indépendance.....	73
3.4.2	Copules Archimédiennes	73
3.4.2.1	Définition.....	73
3.4.2.2	Exemples de copules Archimédienne.....	74
3.4.3	Les tests d'adéquation	75
3.4.3.1	Description de l'algorithme de simulation d'une copule	76

3.4.3.2 Description de l'algorithme de simulation d'une copule Archimédienne	76
3.5 Notre proposition.....	77
3.5.1 Première étape	78
3.5.1.1 Extraction des caractéristiques de ces images :.....	79
3.5.1.2 Application du K-MEANS pour faire une classification initiale :	80
3.5.1.3 Test d'adéquation de Kolmogorov-Smirnov :.....	82
3.5.1.4 Détermination des lois marginales et calcul des fonctions de répartition et les fonctions de densité :	83
3.5.2 Deuxième étape	85
3.5.2.1 Détermination de la copule.....	85
3.5.2.2 Calcul des probabilités	87
3.6 Conclusion.....	90
4- Chapitre 4 EXPÉRIMENTATION	91
4.1 Introduction	91
4.2 Les caractéristiques utilisées	91
4.2.1 La couleur.....	92
4.2.1.1 L'espace de couleur RGB.....	93
4.2.1.2 L'espace de couleur HSV	95
4.2.2 La texture	97
4.2.2.1 La méthode de matrice de cooccurrence	99
4.3 Les mesures d'évaluation	103
4.4 Les méthodes étudiées (comparées)	104
4.4.1 Classification avec l'algorithme de K-MEANS.....	104
4.4.2 Classification avec l'algorithme de la Mixture de Gaussienne.....	107
4.4.3 Classification avec notre algorithme	110
4.5 Expérimentation et évaluation.....	111
4.6 Conclusion.....	112
CONCLUSION GÉNÉRALE.....	113

RÉFÉRENCE.....	115
----------------	-----

Table des figures

<i>Figure 1</i> : Schéma général expliquant la recherche d'image en utilisant la requête.....	4
<i>Figure 2</i> : Recherche basée sur la navigation dans un catalogue	6
<i>Figure 3</i> : Recherche d'images dans une base de données en utilisant une requête textuelle	9
<i>Figure 4</i> : Résultat de la recherche avec le mot « Chats ».....	9
<i>Figure 5</i> : Exemple d'une image qui montre les limites de la recherche d'image par le texte....	11
<i>Figure 6</i> : Recherche d'images dans une BDD en utilisant une image requête.....	12
<i>Figure 7</i> : Résultat de la recherche en utilisant une image exemple	13
<i>Figure 8</i> : Les différents histogrammes d'une image couleur	16
<i>Figure 9</i> : Exemple de base de données indexé.....	25
<i>Figure 10</i> : Exemple d'un k-d tree.....	33
<i>Figure 11</i> : Exemple d'un R-Tree.....	35
<i>Figure 12</i> : Schéma d'un neurone biologique [34].....	37
<i>Figure 13</i> : Schéma d'un neurone formel.....	39
<i>Figure 14</i> : Schéma d'un perceptron multicouche.....	41
<i>Figure 15</i> : Réseau de Hopfield.....	43
<i>Figure 16</i> : Loi de Hebb	45
<i>Figure 17</i> : Représentation d'une image dans un espace bidimensionnel	58
<i>Figure 18</i> : Exemple d'extraction des caractéristiques (dans notre cas c'est la texture)	79
<i>Figure 19</i> : Schéma d'une classification en utilisant l'algorithme K-MEANS	81
<i>Figure 20</i> : Indexation de notre base de données en utilisant l'algorithme K-MEANS	81
<i>Figure 21</i> : Détermination des lois marginales.....	83
<i>Figure 22</i> : Calcul des fonctions de répartition et des fonctions de densité	84
<i>Figure 23</i> : Spectre des ondes électromagnétiques.....	92
<i>Figure 24</i> : Spectre visible.....	93
<i>Figure 25</i> : L'espace de couleur RGB	94
<i>Figure 26</i> : Le cercle des couleurs.....	95
<i>Figure 27</i> : L'espace de couleur HSV	96
<i>Figure 28</i> : Quelques exemples de texture	98
<i>Figure 29</i> : Représentation des images avant la classification avec K-MEANS	105
<i>Figure 30</i> : Exemple de classes après application de l'algorithme de K-MEANS.....	106
<i>Figure 31</i> : Exemple de classes après application de l'algorithme GMM.....	109

Liste des abréviations

GMM	Gaussian Mixture Model (Mélange de mixture de Gaussienne)
CBIR	Content-Based Image Retrieval (Recherche d'images par le contenu)
RGB	Reed Green Blue (Rouge Vert Bleu)
HSV	Hue Saturation Value (Teinte Saturation Valeur)
UQTR	Université du Québec à Trois-Rivières
BDD	Base de données
PMC	Perceptron multicouche
QBIC	Query By Image Content
EM	Expectation Maximization (espérance-maximisation)

Chapitre 1

INTRODUCTION GÉNÉRALE

De nos jours, le nombre d'images personnelles où professionnelles ne cesse d'augmenter. Face à cette croissance des données multimédia, il devient de plus en plus difficile à un utilisateur de retrouver l'information recherchée d'où le besoin de développer des systèmes d'indexation qui permettent de retrouver l'information souhaitée le plus rapidement possible, de rendre leur localisation plus efficace et de les organiser de manière à rendre leur accès plus facile, c'est la raison pour laquelle nous nous sommes attaqués dans ce mémoire à l'indexation d'images.

Il existe plusieurs systèmes qui permettent de rechercher et d'indexer une base de données d'images que se soit par le contenu visuel (les caractéristiques des images de bas niveau : couleur, texture, forme) ou par le texte. L'indexation s'applique dans plusieurs domaines comme la recherche d'information visuelle sur le Web, la sécurité, la culture et les musées, l'organisation des collections personnelles d'images, etc. Malgré cela, ces systèmes n'arrivent pas à subvenir aux besoins de l'utilisateur.

Le pilier de l'indexation d'images est l'extraction des signatures (ou caractéristiques) qui représentent l'essentiel de l'image. Dès lors différents outils peuvent être appliqués afin de classifier cette banque de données à savoir la méthode des arbres, la méthode des réseaux de neurones, la méthode statistique et bien d'autres. Dans ce mémoire, nous avons emprunté le chemin qui mène à l'indexation d'images en appliquant la méthode statistique. En effet, dans la plupart des travaux existants, les chercheurs supposent que les signatures obéissent à des lois normales et ils se limitent à estimer les paramètres de ces lois en utilisant des modèles de mixtures de Gaussiennes (GMM).

Dans le premier chapitre de ce mémoire, nous donnerons un aperçu général sur le concept de la recherche et d'indexation d'images, de montrer la nécessité de développer des systèmes d'indexation et pour finir, nous verrons quelques domaines d'application.

Dans le deuxième chapitre intitulé « État de l'art » nous verrons les différentes méthodes d'indexation d'images ainsi qu'une présentation de quelques systèmes existants.

Dans le troisième chapitre, nous allons tout d'abord faire une description détaillée de la méthode qui utilise la mixture de gaussienne vu que nous nous sommes basé dessus, ensuite, nous allons critiquer cette dernière en montrant que les signatures ne suivent pas

que des lois normales, enfin, nous allons décrire notre proposition avec les différentes étapes que nous allons suivre.

Dans le quatrième chapitre, nous allons en premier lieu faire une description détaillée des caractéristiques d'images que nous allons utiliser, en second lieu, Nous avons présenté des algorithmes de classification classiques à savoir l'algorithme K-MEANS et l'algorithme de mixture de gaussiennes. Des résultats de ces derniers algorithmes ont été présentés. Toutefois, la programmation de l'algorithme basé sur les copules archimédiennes est très compliquée du fait que la densité de la copule de Frank retenue était très compliquée à calculer. Nous avons utilisé des approximations de cette dernière, mais, cela ne nous a pas permis d'évaluer notre système à partir de l'algorithme proposé. Malgré l'absence de résultats, nous pouvons dire que notre proposition est très prometteuse vu que les lois marginales ne suivent pas uniquement des lois normales mais suivent des lois gamma, uniforme, etc. Enfin nous complétons notre mémoire avec une conclusion générale.

1- Chapitre2

GÉNÉRALITÉ

1.1 Introduction

L'image est une représentation d'êtres, d'objets ou de scènes par la photographie, le dessin, etc. Elle peut être à une ou plusieurs dimensions. En plus de cela, la démocratisation de l'informatique et l'évolution des réseaux de transmission, en particulier l'internet qui amène à partager et à manipuler un flot impressionnant d'images et de données multimédia, ont donné à toute personne la possibilité d'accéder à un nombre incalculable d'images ce qui n'était pas possible auparavant. C'est pour cette raison que l'image est devenue l'un des principaux supports de l'information. Aujourd'hui, avec l'évolution technologique nous participons à un accroissement phénoménal du nombre de données multimédia que nous trouvons soit dans les ordinateurs personnels, soit sur internet et ceci est dû essentiellement à la faiblesse des prix des supports de stockage ainsi que les dispositifs qui permettent leur acquisition (disque dur externe, les clés USB, les cartes mémoire des appareils photo et des caméras vidéo, les téléphones cellulaires, les webcams, etc). Cette abondance de données n'a pas que des répercussions positives, mais aussi négatives. En effet une fois ces images stockées, le problème est de pouvoir les retrouver le plus simplement et le plus rapidement possible et d'en faciliter l'accès. Tout ça a donné naissance à un nouveau domaine qui est la recherche et l'indexation d'images par le contenu. Ce domaine-là a pour objectif le développement d'outils qui permettent de retrouver les images, en se basant sur leur contenu visuel (couleur, texture, formes, etc.) Ainsi, l'utilisateur pourra accéder aux images recherchées soit suite à la formulation d'une requête, soit par le biais d'un catalogue où il navigue. La majorité des systèmes existants aujourd'hui utilisent les

annotations qui accompagnent les images pour pouvoir indexer ces images. Cette technique est utilisée, entre autres, par « *Google image* » et « *Yahoo image* », mais elle a certaines limitations vu que les annotations sont différentes d'une personne à une autre, aussi les annotations se font dans différentes langues. Pour pallier à ces problèmes, les chercheurs ont eu l'idée de créer des systèmes qui se basent sur le contenu même des images. Ces systèmes s'inscrivent dans un nouveau domaine, qui est en pleine expansion, et qui s'appelle la recherche et l'indexation d'images par le contenu ou Content-Based Image Retrieval (CBIR) en anglais.

Dans ce chapitre, nous allons parler en premier lieu des différents modes de recherche d'images. En second lieu, nous donnerons les différentes étapes de la recherche d'images. En troisième lieu, nous allons présenter l'indexation d'image un peu plus en détail vu que notre recherche porte sur cet aspect. En quatrième lieu, nous fournirons quelques domaines d'application de la CBIR. Finalement, nous présentons une conclusion du chapitre.

1.2 Différent modes de recherche d'images

On peut catégoriser la recherche d'images selon différents critères :

- La façon dont l'utilisateur communique avec le moteur, on peut la catégoriser en recherche basée sur une requête vs. Navigation dans un catalogue. Ces deux modes seront présentés dans la section 1.1.
- Les attributs utilisés lors de la recherche, on peut la classifier en recherche basée sur le texte entourant les images vs recherche basée sur le contenu des images. Ces deux types de recherche seront présentés dans la section 1.2.

1.2.1 Mode de communication entre l'utilisateur et le moteur de recherche

La recherche d'images est un domaine qui est en pleine expansion et qui est une véritable clé technologique pour les services multimédias et leur large diffusion vers le grand public. Mais dans ce domaine il existe deux façons de faire la recherche d'images. Le premier est celui de la formulation de la requête, le deuxième est celui de la navigation dans la base de données. Dans la section 1.1.1, nous présenterons le premier mode de recherche, puis dans la section 1.1.2 nous expliquerons le deuxième mode.

1.2.1.1 *Recherche basée sur une requête*

D'une manière générale, pour faire une recherche d'images, l'utilisateur commence par saisir une requête, ensuite le moteur de recherche compare la requête avec les données qu'il a dans sa collection d'images. Une fois les résultats obtenus, le moteur les affichent à l'utilisateur via l'interface. La figure 1 illustre cette démarche :

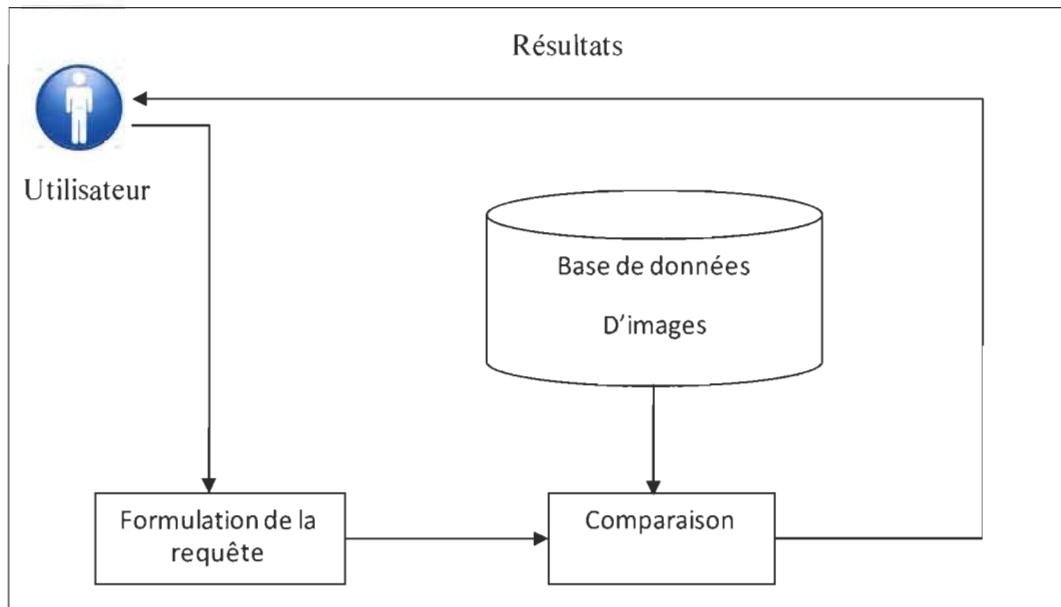


Figure 1 : Schéma général expliquant la recherche d'image en utilisant la requête.

Nous distinguons principalement deux types de requêtes à savoir une recherche qui est faite avec une requête textuelle et une autre qui est faite avec une image exemple. Dans la section 1.2, nous allons décrire ces deux principales techniques.

1.2.1.2 Recherche basée sur la navigation dans un catalogue

Dans ce deuxième mode de recherche d'images, le système permet à l'utilisateur via une interface, de naviguer dans la base de données en toute liberté. En effet, l'utilisateur peut commencer à faire sa recherche à partir de la racine du catalogue ou par un nœud donné et de naviguer facilement en parcourant ce thème en allant soit en profondeur pour parcourir les sous thèmes s'ils existent, soit en allant parcourir d'autres thèmes connexes, et de bien préciser ses besoins s'il n'a pas une idée bien précise de ce qu'il doit chercher comme image et d'identifier ce qu'il recherche avec exactitude tout ça via

une interface graphique. Mais pour pouvoir effectuer cette méthode de recherche d'images, il faut créer un catalogue. Pour sa création, il s'agit d'organiser les images de la base de données au sein d'une structure hiérarchique. Pour cela, nous allons utiliser les mesures de similarité qui permettent de classifier toutes les images dont les caractéristiques se ressemblent et de les regrouper au sein d'une même classe. Cette technique permet de gagner du temps puisque l'utilisateur ne sera plus obligé de parcourir toute la base de données, aussi elle permet de résoudre le problème de la page zéro.

Durant la section précédente nous avons présenté une description des différentes méthodes de recherche d'images en l'occurrence la recherche par la requête et la navigation dans une banque de données. Cependant pour obtenir de meilleurs résultats dans un temps raisonnable, la solution optimale consiste à organiser les images au sein d'une structure hiérarchique généralement appelée indexe comme représenter dans la figure 2. Dans ce qui suit, nous allons décrire la notion d'indexation des images.

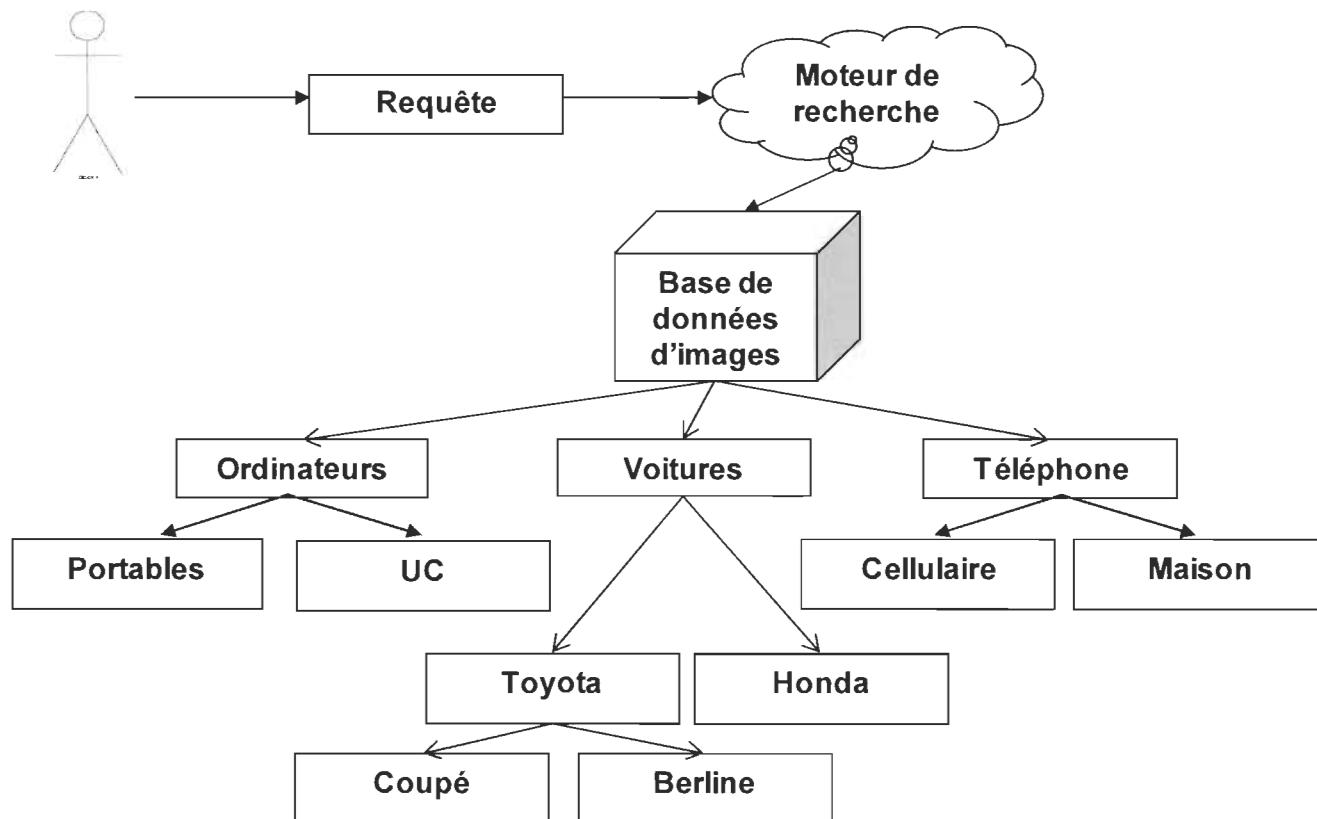


Figure 2 : Recherche basée sur la navigation dans un catalogue.

1.2.2 Type de recherche d'images

Dans cette section, nous allons voir les types de recherche d'images à savoir la recherche d'images par une requête textuelle et la recherche d'images basée sur le contenu.

1.2.2.1 *Recherche d'images basée sur le texte*

La formulation de la requête est une étape de communication qui se déroule entre l'usager et le moteur de recherche. Dans notre cas, la requête formulée par l'utilisateur est textuelle. En effet, cette requête peut être soit un mot clé, un ensemble de mots clés ou une phrase décrivant ce que l'utilisateur recherche comme image. Après cela, le moteur parcourt la base de données à la recherche d'images décrites avec ces mêmes mots-clés. Dès l'obtention des résultats, le moteur les affiche à l'usager. Toutefois sans une représentation textuelle de ces images, cette méthode ne fonctionnera pas, il faut donc que les images soient décrites à l'avance. Cette description provient en général de deux sources :

- ✓ Elle provient du texte qui entoure les images comme le texte contenu dans la ou les pages Web où se trouve l'image. En effet le moteur de recherche collecte les images sur les sites Web en explorant tous les liens qu'il rencontre et il récupère les ressources qu'il juge pertinentes. Ensuite, il extrait les mots qui entourent les images et qu'il considère signifiants, et les conservent dans sa banque de données.
- ✓ Elle provient de l'annotation. L'annotation est une démarche par laquelle on assigne du texte à une image. En fait, il existe trois sortes d'annotation.

- i) La première est celle qu'on appelle l'annotation manuelle. Comme son nom l'indique, cette annotation se réalise manuellement par un être humain et dont l'objectif est d'attribuer à chaque image un mot-clé ou un ensemble de mots-clés. Notons que pour une grande collection d'images cette technique devient très difficile, voire impossible à accomplir par un être humain.
- ii) La deuxième est appelée l'annotation automatique. Elle consiste à attribuer automatiquement des mots-clés à une image via un système informatique. Cette méthode d'annotation vise à diminuer la charge de travail nécessaire dans le cas de l'annotation manuelle. Mais dans le cas d'images complexes, cette technique devient inefficace d'où ses limites.
- iii) La troisième et dernière est appelée annotation semi-automatique. C'est la combinaison de l'annotation manuelle et de l'annotation automatique. Dans cette technique, l'intervention d'un annotateur humain est obligatoire puisqu'il sera chargé d'annoter une image donnée et d'effectuer une propagation des mots-clés au reste des images de la base de données considérées comme étant similaires.

Pour retrouver les images à l'aide du texte, le moteur de recherche a besoin de mesurer la similarité entre le texte de la requête et celui qui accompagne les images de la base de données. En général, on applique l'appariement (*matching en anglais*) pour examiner la présence ou non d'images semblables au texte de la requête. Dans la figure 3, nous donnons un schéma explicatif sur la recherche d'images par le texte.

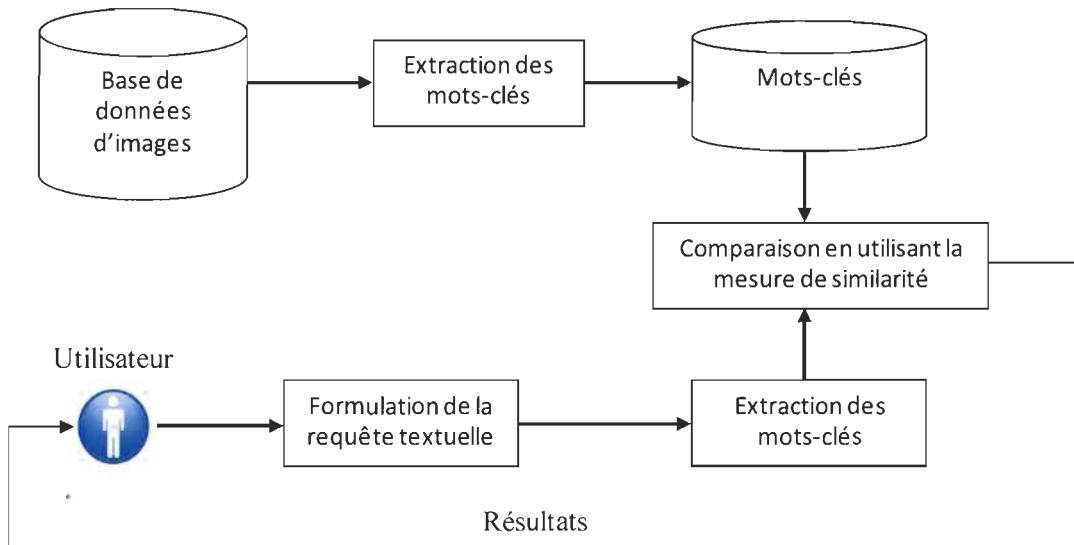


Figure 3 : Recherche d'images dans une base de données en utilisant une requête textuelle.

Voici par ailleurs un exemple de recherche d'images sur *Google Image* par le mot-clé « chat » (figure 4) :



Figure 4 : Résultat de la recherche avec le mot « Chats ».

Notons que cette façon de rechercher des images en utilisant une requête textuelle est avantageuse puisqu'elle :

- ✓ Permet de laisser la libre expression à l'utilisateur pour la saisie de sa requête et l'expression des besoins.
- ✓ Permet de mieux canaliser les concepts sémantiques qui sont liés aux images.

Ceci étant, la recherche utilisant cette technique rencontre des défis à surmonter comme :

- ✓ L'annotation pour la même image. En effet, plusieurs annotateurs peuvent annoter une image de plusieurs façons différentes.
- ✓ Les différentes langues d'annotations des images. En effet, les annotateurs peuvent parler des langues différentes et l'annotation d'une image va différer d'un annotateur à un autre puisque chacun d'entre eux va l'annoter dans sa langue.
- ✓ Un utilisateur peut vouloir rechercher des images qui ressemblent à l'image de la figure 5, alors il sera dans l'incapacité de décrire avec exactitude la forme de la maison, sa couleur, son contenu, etc. Cela montre une des raisons pour laquelle la recherche d'images par le texte est très limitée.



Figure 5 : Exemple d'une image qui montre les limites de la recherche d'image par le texte.

Vu les inconvénients de cette méthode, cela nous amène à une autre technique que nous allons voir dans la section suivante qui est la recherche d'images par le contenu et cette dernière permet de dépasser les frontières de la recherche textuelle.

1.2.2.2 *Recherche d'images basée sur le contenu*

Maintenant que nous avons vu les limites de la recherche par le texte, nous allons voir dans cette section la deuxième méthode de recherche qui utilise l'exemple comme requête. Cette méthode est présentée par la figure 6. Cette technique de recherche d'images est connue sous le nom de recherche d'images par le contenu ou *Content-based image retrieval* en anglais. Cette approche utilise la même démarche décrite précédemment. Le principe de cette méthode est que l'utilisateur commence par saisir sa requête. Dans le cadre de ce modèle de recherche, l'idée est de permettre à l'utilisateur

de formuler la requête soit par une image, soit par dessin, soit un croquis, soit par des icônes, etc. Mais la technique de recherche utilisant l'image comme requête est sans doute celle qui a connue le plus de réussite. Aussitôt, l'étape de la saisie est terminée, le moteur parcourt la base de données à la recherche d'images qui correspondent à cette requête. Une fois les résultats obtenus, le système les affichent à l'utilisateur. Cette recherche par l'exemple se fait à travers ce qu'on appelle les caractéristiques qui représentent d'une manière appropriée le contenu de l'image. Ces signatures sont représentées par des valeurs numériques [1] en sachant qu'elles sont plus objectives que les descripteurs textuels. Dans ce qui suit, nous allons faire une brève description de ces signatures.

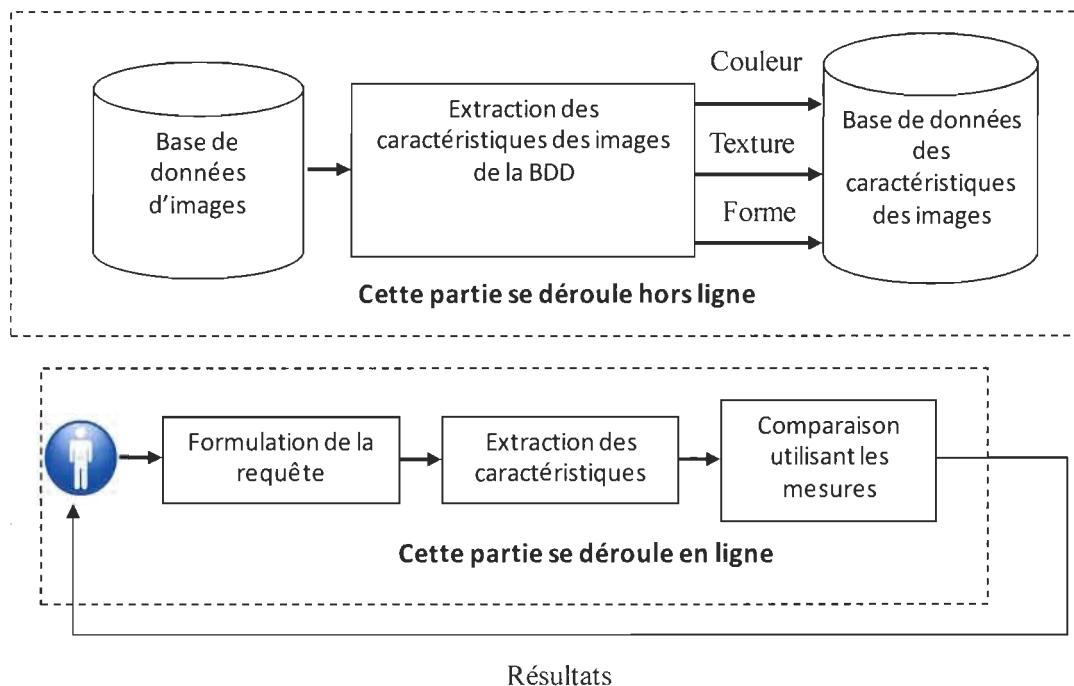


Figure 6 : Recherche d'images dans une BDD en utilisant une image requête

La figure 7 ci-dessous montre le résultat d'une recherche d'images par le contenu:

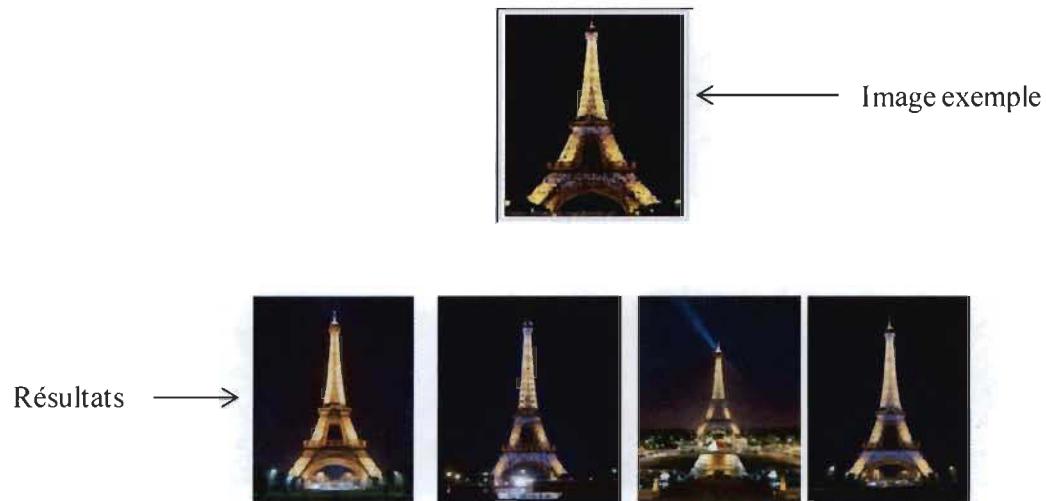


Figure 7 : Résultat de la recherche en utilisant une image exemple

Sachant que cette méthode de recherche par le contenu comporte certains avantages par rapport au texte comme :

- ✓ Elle n'engendre pas de problème même si la base de données ne comporte aucun texte.
- ✓ Elle est applicable aux images qui sont complexes au niveau des scènes, des objets, etc.
- ✓ Elle a un niveau de raffinement bien meilleur que celui du texte.

- ✓ Elle est plus efficace que la recherche textuelle puisqu'une image est plus descriptive que le texte (une image vaut mille mots).

Ceci étant, elle renferme des inconvénients et des défis à relever par exemple :

- ✓ L'extraction des différentes caractéristiques de l'image.
- ✓ Le fossé sémantique.
- ✓ Le problème de la page zéro.
- ✓ La classification des images de la banque de données en des classes à partir des caractéristiques des images.

Nous venons de présenter les deux modes de recherche d'images à savoir celle qui est basée sur le texte et celle qui est basée sur le contenu. Dans la section suivante, nous allons voir les différentes étapes de la recherche d'images par le contenu.

1.3 Les étapes de la recherche d'images

Pour faire une recherche d'images basée sur le contenu, nous avons besoin d'extraire les caractéristiques des images de la base de données ainsi que celle de la requête. Une fois ces caractéristiques extraites, on va faire la comparaison entre les signatures de l'image requête avec celles de la collection d'images. Pour faire cette comparaison, nous avons besoin des mesures de similarité. Afin que la recherche soit plus rapide et plus efficace, la collection d'images doit être indexée. Dans cette partie du chapitre, nous allons présenter en premier lieu les caractéristiques des images. En second lieu, nous

donnerons quelques mesures de similarités et en troisième lieu, nous allons parler brièvement de l'indexation.

1.3.1 Les caractéristiques des images

Les signatures visuelles les plus utilisées pour la description du contenu des images sont généralement les suivantes :

1.3.1.1 *La couleur*

La couleur est l'une des caractéristiques les plus utilisées pour l'indexation et la recherche d'images. Il existe plusieurs types de couleurs par exemple le dispositif HSV, RGB, L*a*b, etc. Mais le plus utilisé est le RGB. Il existe plusieurs attributs pour caractériser la couleur, on peut citer les histogrammes, les moments de couleurs, etc. [2].

1.3.1.1.1 L'histogramme de couleur

Un histogramme de couleurs permet de représenter la distribution de probabilité des couleurs dans une image. En effet, l'histogramme est produit en découplant en premier lieu les bandes de couleur utilisé dans un certain nombre de cases, en second lieu en comptant dans chaque case le nombre de pixels. La figure 8 ci-après, illustre un exemple d'une image avec ses différents histogrammes pour chaque bande de couleur.

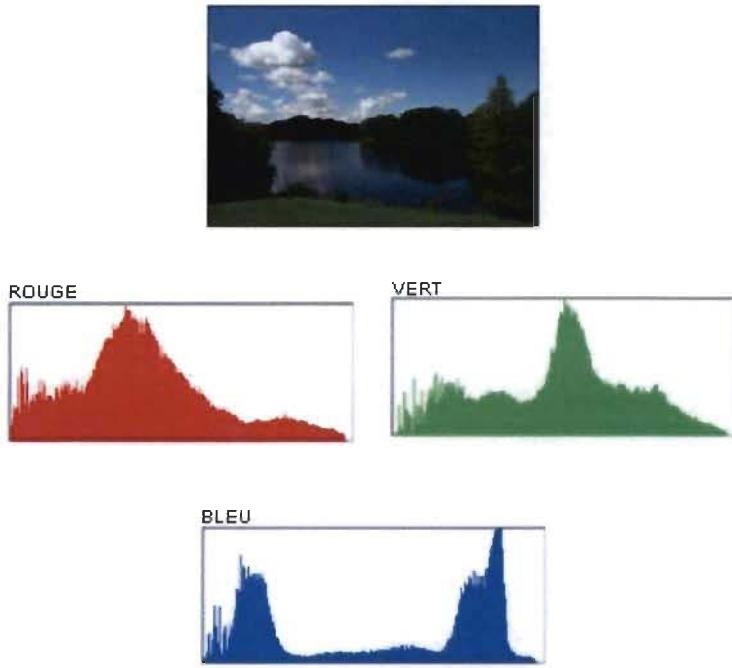


Figure 8 : Les différents histogrammes d'une image couleur RGB.

L'une des techniques les plus utilisées est l'intersection des histogrammes [2, 3, 4]. Lorsqu'on va effectuer une recherche d'images par le contenu, on prend l'histogramme de l'image requête et on le compare avec les histogrammes de la base de données. Mais pour effectuer cette comparaison soit on calcule la distance euclidienne soit en applique une mesure de similarité. Les histogrammes présentent certains avantages comme leur facilité et la rapidité de les calculer. Ils présentent certains inconvénients comme leur sensibilité au changement de luminosité et leur absence d'information spatiale. Ainsi pour remédier à ces inconvénients, plusieurs démarches ont été réalisées comme ce que proposent d'une part Stricker et Dimai [5] en découplant l'image en cinq blocs de zone d'intérêt et d'une autre part Y. Gong, G. Proietti et C. Faloutsos [6] qui utilisent l'espace de couleur HVC (Hue, Value, Chroma) puisqu'il est plus proche de la perception humaine et donne des résultats plus précis.

1.3.1.1.2 Les moments statistiques

L'histogramme utilise la distribution complète de la couleur ce qui engendre une énorme quantité de données à stocker et par conséquent qui prend plus de temps et de mémoire. Pour résoudre cet inconvénient au lieu de calculer la distribution complète de l'histogramme on calcule seulement des dominantes caractéristiques de couleurs tels que l'espérance, la variance et d'autres moments. On peut calculer l'espérance, la variance, les moments sur chaque composante couleur par la formule suivante :

$$E = \frac{1}{N} \sum_{j=1}^N P_j \quad (1.1)$$

$$\delta = \left(\frac{1}{N} \sum_{j=1}^N (P_j - E)^2 \right)^{\frac{1}{2}} \quad (1.2)$$

$$S = \left(\frac{1}{N} \sum_{j=1}^N (P_j - E)^3 \right)^{\frac{1}{3}} \quad (1.3)$$

1.3.1.2 La texture

La texture est étudiée depuis maintenant une vingtaine d'années. En effet, suite aux nombreuses études qui ont été réalisées, la texture est devenue l'une des caractéristiques qui est de plus en plus utilisée dans la recherche et l'indexation d'image par le contenu. D'après Tuceryan et Jain [7], il existe quatre approches pour analyser la texture qui sont l'approche statistique, l'approche par modélisation, l'approche géométrique et l'approche spectrale. Pour la première approche (statistique) Haralick [8], propose une analyse des textures en utilisant les matrices de cooccurrences. Pour la seconde (par modélisation), elle inclut la décomposition de Wold [9] qui caractérise la direction et la

périodicité. La troisième en l'occurrence l'approche géométrique, on trouve celle de Tamura [10] qui décrit toutes les textures possibles suivant les concepts qui correspondent à la vision de l'être humaine avec six propriétés proposées : granularité, contraste, direction prédominante, linéarité, régularité et rugosité. Enfin, la dernière approche contient les ondelettes avec les filtres de Gabor qui permettent la capture des directions principales et des fréquences [11] [12].

1.3.1.3 La forme

La forme est aussi l'une des principales primitives visuelles de base appliquée dans la recherche et l'indexation d'images. Cette primitive peut être définie comme étant à peu près un descripteur du contenu de l'image, mais elle devrait être invariante à la translation, rotation et aux changements de l'échelle lorsque la disposition des objets dans l'image ne sont pas connus à l'avance. Suivant cette idée, Hu [13] a proposé dans son article sept moments invariants aux changements de l'échelle, aux translations et aux rotations. Par conséquent, il y a eu l'émergence de plusieurs applications de recherche d'images par le contenu en utilisant la signature de la forme [14] [15] [16].

Maintenant que nous venons de présenter comment on procède pour faire une recherche d'image en utilisant l'exemple ainsi que les caractéristiques utilisées dans cette même recherche, nous allons aborder dans le point suivant les techniques de similarité qui permettent de comparer les signatures de la requête avec celles de la base de données d'images.

1.3.2 Mesure de similarité

Les mesures de similarité ne permettent pas de juger si une image ressemble à une autre, mais elles permettent de mesurer une distance mathématique et de déterminer si elles sont équivalentes ou indépendantes. Donc dans notre cas, ces mesures comparent les distances des caractéristiques de l'image de la requête avec celles de la base de données. Il en existe plusieurs de ces mesures qui font cette comparaison, mais les plus utilisées dans la recherche d'images sont :

1.3.2.1 *La distance euclidienne*

La distance euclidienne entre deux histogrammes de la couleur h et g est calculée pour chaque bande de couleur. La distance euclidienne est définie comme la racine de la somme des carrés des différences des coordonnées et est calculée comme suit :

$$d(h, g) = \left(\sum_{j=1}^P (h_j - g_j)^2 \right)^{\frac{1}{2}} \quad (1.4)$$

où P est le nombre de cases de l'histogramme et h et g sont deux vecteurs de caractéristiques de l'histogramme de la couleur de l'image à comparer.

1.3.2.2 La distance pondérée

La distance pondérée entre deux histogrammes de la couleur h et g est calculée pour chaque bande de couleur comme suit :

$$d(h, g) = \left(\sum_{j=1}^P M_j (h_j - g_j)^2 \right)^{\frac{1}{2}} \quad (1.5)$$

où h et g sont deux vecteurs de caractéristiques de l'histogramme de couleurs de l'image à comparer et le coefficient M_j pondèrent l'influence de la $j^{\text{ème}}$ variable.

1.3.2.3 La distance de Mahalanobis

La distance pondérée entre deux histogrammes de la couleur h et g est calculée pour chaque bande de couleur comme suit :

$$d(h, g) = \left((h - g)^T K^{-1} (h - g) \right)^{\frac{1}{2}} \quad (1.6)$$

Où h et g sont deux vecteurs de caractéristiques de l'histogramme de couleur de l'image à comparer et K est la matrice de variance-covariance.

1.3.3 L'indexation

Durant ces dernières années, il y a une explosion phénoménale de la quantité de données multimédias et cela est dû à la diminution des prix des appareils de captures et des supports de stockage (ordinateurs, webcams, disques durs de grande capacité, téléphones cellulaires, appareils photo, caméra, etc.). Avec l'évolution des réseaux de transmission de données comme l'internet et l'augmentation de la capacité des disques de stockage, donne la possibilité de sauvegarder un nombre incalculable d'images dans nos machines. Face à l'abondance de ces données, il devient très difficile à un utilisateur de retrouver les informations souhaitées le plus simplement et le plus rapidement possible. L'index permet de diminuer le temps de recherche puisque le moteur n'aura plus à parcourir toutes les classes de la banque de données, mais il se limitera plutôt aux classes les plus vraisemblables.

Durant cette section nous avons vu les différentes étapes de la recherche d'images. Dans ce qui suit, nous allons parler de l'indexation d'images, mais d'une façon détaillée vu que notre recherche se base sur l'indexation.

1.4 Indexation

Dans cette partie du chapitre, nous allons tout d'abord définir l'indexation, ensuite nous donnerons les différentes étapes de l'indexation et enfin, nous aborderons les différentes techniques d'indexation.

1.4.1 Qu'est ce que c'est que l'indexation

L'indexation est la représentation et l'organisation d'une manière efficace le contenu des documents d'une base de données sous la forme d'un arbre généalogique afin d'en faciliter l'accès et l'utilisation.

1.4.2 Les différentes étapes de la création d'un index

Pour créer un index, il y a un certain nombre d'étapes à effectuer qui sont :

- ✓ L'extraction des caractéristiques (appelées aussi signatures, attributs ou descripteurs) qui vont servir à trouver et classifier les images de la banque de données.
- ✓ La classification des images à partir des signatures extraites afin d'accélérer la recherche.
- ✓ La création d'une interface qui va permettre à l'utilisateur de communiquer et de naviguer dans la base de données. Aussi, l'utilisateur aura besoin de cette interface pour saisir sa requête et de lui afficher les résultats de la recherche.

Mais il y a différentes méthodes pour indexer une base de données d'images. Le paragraphe ci-dessous fera l'objet d'une étude détaillée de ces méthodes.

1.4.3 Les différentes techniques d'indexation

Dans la majorité des systèmes existants, on utilise soit l'indexation textuelle soit l'indexation basée uniquement sur le contenu.

1.4.3.1 Indexation basée sur le texte

Pour une indexation textuelle, il y a deux façons de procéder : soit une indexation textuelle manuelle, soit une indexation textuelle automatique. Pour l'indexation textuelle manuelle, dans la plupart du temps elle est faite par une personne qu'on appelle un « *iconographe* ». Il est chargé d'attribuer un certain nombre de mots-clés à chaque image et de classifier et d'indexer chacune de ces images de la banque de données en les associant à différentes catégories. Elle présente certains avantages puisqu'elle donne la possibilité de décrire exactement les besoins de l'utilisateur, mais cette technique présente différents problèmes dont le plus majeur est celui du choix des termes pour décrire les images en l'occurrence l'annotation. En effet, toute personne qui va annoter manuellement les images et toute personne qui va rechercher des images ne va pas forcément utiliser le même terme et la même langue pour décrire le même concept puisqu'une image peut avoir plusieurs sens et contenir plusieurs thèmes. Et lorsqu'on a une grande banque de données d'images, ce mode d'indexation devient fastidieux, voire impossible à réaliser, il est donc indispensable de posséder des outils qui permettent d'annoter les images d'une manière pertinente. Cela nous ramène à la deuxième façon qui est l'indexation textuelle automatique. Elle consiste à assigner des mots clés à une image au moyen d'un système informatique sans l'intervention de l'iconographe ou de toute autre personne. Cette approche n'est exécutable que si les images sont associées à du texte comme c'est le cas pour les images sur Internet, dans les manuels techniques, dans les encyclopédies, les catalogues de vente, etc. Cette technique arrive à résoudre les

inconvénients de l'indexation manuelle tels que le gain du temps, la diminution de la charge de travail nécessaire, etc. Ceci étant, elle présente aussi certains défis à réaliser comme le manque de précision puisque les algorithmes disponibles actuellement ne permettent pas d'extraire la sémantique de l'image avec exactitude. Pour palier aux lacunes de l'indexation textuelle, il y a eu l'émergence d'une approche d'indexation qui est basée sur le contenu des images. Dans ce qui suit, nous allons décrire cette technique.

1.4.3.2 Indexation basée sur le contenu

L'indexation basée sur le contenu, consiste à extraire toutes les signatures (couleurs, formes, textures, etc.) numériques de toutes les images de la BD, de les organiser d'une façon à optimiser la recherche. Dans la majorité des cas, on utilise une structure hiérarchique pour organiser et faciliter l'accès à la base de données. Cette structure est basée sur les classes c'est-à-dire que les images dont leurs caractéristiques sont presque identiques sont classées dans une même classe et les classes semblables sont regroupées au sein d'une même classe et ainsi de suite jusqu'à ce qu'on obtienne la structure arborescente de l'index. Cette structure donne aussi la possibilité à l'utilisateur de naviguer plus facilement dans l'index. La création d'un index permet non seulement de gagner beaucoup plus de temps lorsqu'on fait une recherche par une image exemple puisque le moteur de recherche n'est plus obligé de parcourir toute la banque de données, mais il se limitera plutôt aux classes les plus vraisemblables, mais aussi de naviguer plus facilement dans l'index dans le cas où l'utilisateur n'arrive pas à trouver ce qu'il recherche avec exactitude. La figure 9 représente un exemple d'une base de données d'images indexées.

La base de données d'images



Base de données indexée



Figure 9 : Exemple de base de données indexée.

Nous venons de voir les deux approches pour indexer une collection d'images, mais pour une meilleure indexation, on peut combiner le texte et le contenu ce qui va permettre de diminuer le bruit dans chacune des classes de la base de données.

Après que nous ayons fait une description sur l'indexation et ses différentes méthodes, nous allons présenter dans la section suivante les différents domaines d'applications qui utilisent l'indexation et la recherche d'images par le contenu.

1.5 Les domaines d'application

Ces dernières années, il y a eu l'identification d'un grand nombre d'applications qui utilisent le système de recherche d'images par le contenu et ça a touché plusieurs domaines comme la criminologie, la médecine, la recherche sur le web et bien d'autres.

1.5.1 La criminologie

En général, en criminologie, un nombre incalculable de preuves visuelles sont conservées dans les bases de données de la police comme les empreintes digitales, les empreintes de chaussures, les photos de suspects, les photos des scènes de crime, les photos des preuves recueillies sur les lieux du crime, etc. À chaque fois qu'il y a un crime, les scientifiques de la police comparent les données de la scène du crime avec les données qu'ils ont dans leur banque de données.

Les systèmes qui sont basés sur l'appariement automatique d'empreintes digitales ont été créés dans les années 80 [18] et ils sont utilisés en majorité par les forces de police dans le monde entier comme le FBI. De nos jours, il existe dans le commerce quelques systèmes d'identification automatique d'empreintes digitales. On peut prendre l'exemple du système AFIX (<http://www.afix.com/>) qui a été développé à Pittsburgh au Kansas ou celui de The Finger (<http://www.east-shore.com/>) développé à New York. Ces deux systèmes ont la capacité d'aller chercher des empreintes dans une grande banque de données [17].

1.5.2 La recherche sur le Web

Ces dernières années, les données (images) qu'on peut trouver sur le web augmentent de jour en jour ce qui amène à une nécessité qui est d'avoir des moteurs de recherche d'images par le contenu. C'est devenu un besoin évident. Pour essayer de résoudre ce problème, il y a eu l'apparition de plusieurs systèmes qui ont été démontrés, parmi eux Atlas WISE. Atlas WISE a été développé au laboratoire de mathématique et d'informatique à l'université de Sherbrooke [33]. Son mode de fonctionnement se résume à rassembler toutes les images à partir des sites les plus connus comme Google et Yahoo et en même temps extraire le texte des images plus exactement les mots-clés pertinents qui sont estimés à partir de leurs légendes, les titres de page, et le texte qui l'entoure. Une fois ces données (images et texte) sont extraites, elles vont être stockées jusqu'à qu'elles soient digérées, mais une fois après elles vont être supprimées et ça permettra de laisser de l'espace de stockage néanmoins Atlas WISE garde en mémoire les liens pour une utilisation ultérieure. Les caractéristiques d'images qu'utilise Atlas WISE sont les histogrammes de couleur et les mots clés. Mais la particularité d'Atlas WISE c'est qu'on peut faire la recherche non seulement avec une requête d'images exemplaires, mais aussi avec une requête d'images de contre exemple. En 1998, il y a eu l'apparition de deux moteurs de recherche qui proposent la méthode de recherche d'images par le contenu en l'occurrence le Surfer Yahoo Image qui utilise la technologie d'Excalibur et AltaVista AV Photo Finder qui utilise à son tour la technologie Virage qu'on a vue au point précédent. Ces deux moteurs proposent une méthode de recherche assez limitée vu le nombre de fonctionnalités qu'ils offrent par rapport aux autres systèmes comme WebSEEk. Mais l'avantage de ces moteurs c'est qu'ils utilisent une autre méthode supplémentaire, mais très utile pour localiser les images désirées sur Internet.

1.5.3 Le domaine médical

Le domaine médical est en train de se moderniser d'année en année ce qui a entraîné une énorme explosion surtout dans les techniques de diagnostic comme la radiologie, l'histopathologie, et la tomodensitométrie et cela a engendré l'émergence d'un très grand nombre d'images médicales qui sont désormais stockées par la plupart des hôpitaux. Bien que les systèmes d'imagerie médicale sont de plus en plus exigés, il y a un intérêt croissant pour l'utilisation de techniques de recherche d'images par le contenu pour faciliter le diagnostic par l'identification des cas similaires passés [17]. Les travaux déjà existants fournissant des fonctionnalités de base. Toutefois, les systèmes expérimentaux de recherche d'images par le contenu commencent à avoir un grand impact, par exemple le système I2C qui a été créé à l'Université de Crète [19] et qui permet de récupérer des images en 2 dimensions comme les images radiologiques. Aussi la 3-D du système neurologique de recherche d'images en cours d'élaboration au Carnegie-Mellon University [20], développé dans le but d'aider le personnel médical dans le diagnostic des tumeurs cérébrales.

1.6 Conclusion

Dans ce premier chapitre, nous avons vu les concepts généraux utilisés dans la recherche et l'indexation d'images. Nous avons vu, par exemple, la recherche des images par le texte et par le contenu en utilisant des mesures de similarité et cela était motivé par le nombre croissant des données multimédias à cause de la baisse des prix appareils qui permettent leurs captures. Nous avons également présenté la notion d'indexation d'une grande collection d'images basée sur le texte ou sur le contenu puisque sans une méthode appropriée de stockage et d'extraction des caractéristiques il devient très difficile d'exploiter ces données.

Dans le prochain chapitre, nous discutons différentes techniques d'indexation existantes et nous décrirons quelques systèmes réalisés dans ce domaine.

2- Chapitre 2

2 ÉTAT DE L'ART

3.1 Introduction

Dans le premier chapitre, nous avons parlé de la nécessité de la recherche d'images ainsi que des différentes tâches que doit accomplir un moteur de recherche d'images, en l'occurrence : l'extraction des caractéristiques, la mesure de la similarité et l'indexation. Cette dernière constitue justement l'objet principal de notre recherche. Par conséquent, nous allons consacrer le présent chapitre à discuter de l'indexation et des différentes approches qui ont été utilisées pour l'accomplir.

L'indexation consiste à organiser une grande collection d'images en une structure qui peut être hiérarchique. Pour pouvoir accomplir cette tâche d'indexation, nous avons besoin dans un premier temps d'extraire les caractéristiques des images. Une fois ces caractéristiques extraites, elles seront utilisées pour classifier les images de la collection en un certain nombre de classes où les images de la même classe sont relativement semblables. Parfois l'index est hiérarchique, ce qui veut dire que les classes entre lesquelles il existe une certaine ressemblance sont regroupées dans des classes plus grandes et ainsi de suite.

Différentes méthodes mathématiques ont été appliquées pour indexer les collections d'images. En effet, une fois les caractéristiques extraites, la collection d'images peut être vue comme un grand ensemble de données dans un espace multidimensionnel. Dès lors, différents techniques d'indexation peuvent être appliquées comme la technique à base d'arbres, la technique des réseaux de neurones ou la technique statistique.

Dans la section 2.2 de ce chapitre, nous allons présenter certaines de ces techniques en détail. Dans la section 3, nous allons présenter quelques moteurs de recherche existants qui utilisent la méthode d'indexation. Dans la section 4, nous allons critiquer les techniques qui ont été présentées dans la section 2. Et dans la section 5, nous allons conclure ce chapitre.

3.2 Les techniques d'indexation

Dans ce qui suit, nous allons faire une description de quelques techniques d'indexation qui ont été utilisées pour indexer une base de données d'images.

3.2.1 Les techniques d'arbre

Les techniques d'indexation multidimensionnelles ont commencé à voir le jour vers la fin des années 70. En effet, ces techniques ont commencé à émerger avec l'apparition des structures utilisant des arbres hiérarchiques tels que le k-d tree [21] et les nombreuses variétés du R-tree [22]. Dans ce qui suit nous allons décrire deux méthodes de classification qui sont le kd-Tree et le R-trees.

3.2.1.1 *KD-Tree*

Le kd-Tree, abréviation de « k-dimensional tree », est un arbre qui permet de faire une recherche binaire d'ordre k. Il permet de structurer les données en fonction de leur répartition dans l'espace. Pour la construction de l'arbre k-d tree, Bentley et al [21] ont repris l'idée du système d'indexation du Binary-tree qui consiste à ordonner les valeurs

de l'index d'une manière linéaire. Le Binary-tree divise le volume englobant en k -dimensions en le coupant en deux sous-volumes par un plan de l'espace et en répétant la même démarche sur ces deux nouveaux sous volumes.

Les plans séparateurs peuvent être choisis en fonction de la répartition des données pour que le volume soit d'une grande taille là où il n'y a pas une grande concentration de données et vice versa [23]. Par contre pour le kd-Tree, les plans séparateurs sont toujours choisis d'une manière à ce que leur normal soit un des axes du système de coordonnées de l'espace (plans toujours perpendiculaires aux axes « voir figure 10 »).

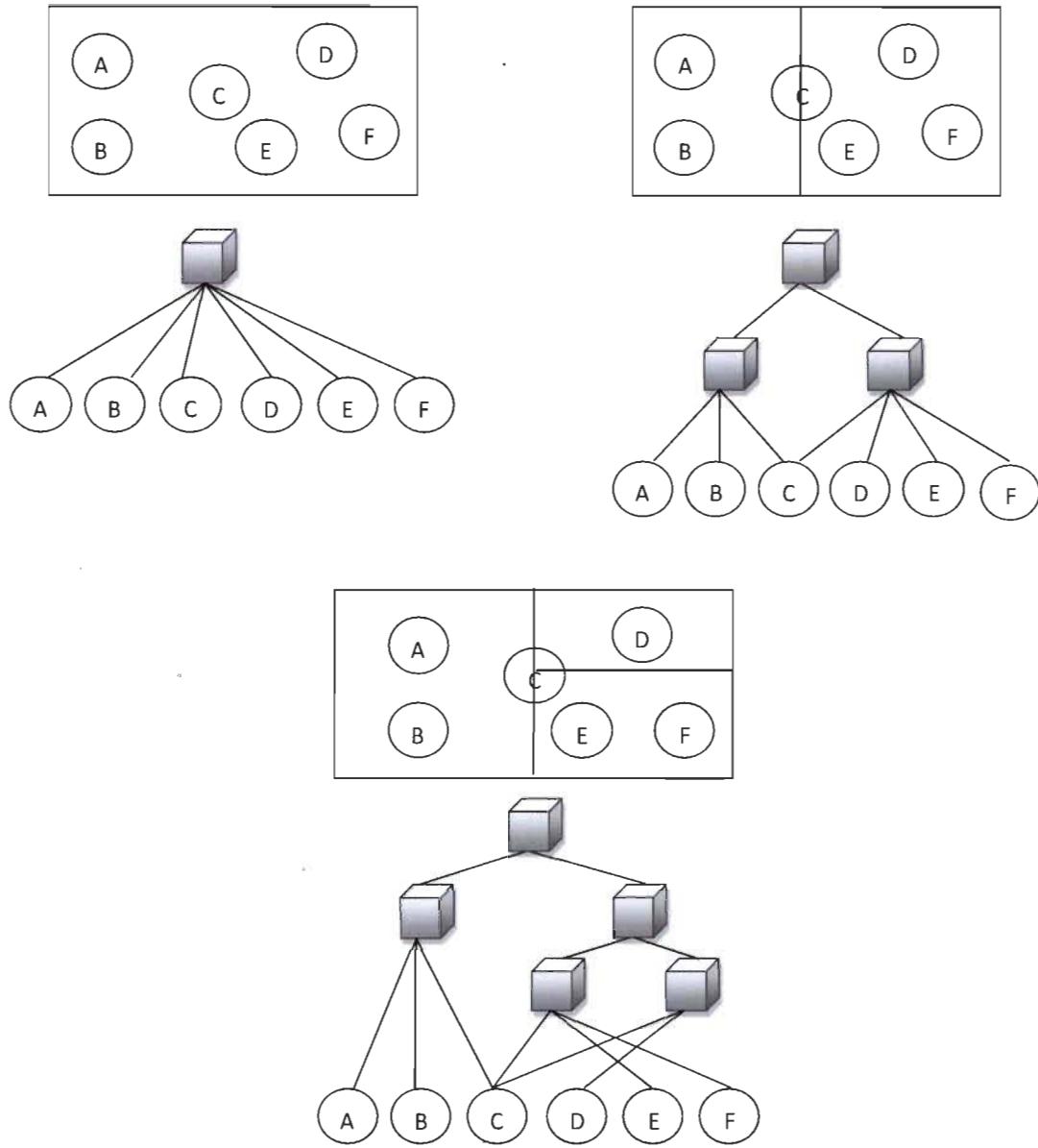


Figure 10 : Exemple d'un k-d tree

Chaque nœud central (nœud central est celui par qui passe les plans séparateurs) divise l'espace en deux. Ce nœud devient un point de référence. Chaque point qui est inférieur au point de référence est placé à gauche dans l'arbre par contre chaque point qui est supérieur au point de référence est placé à droite dans la structure de l'arbre. Le kd-Tree

joue deux rôles : dans le premier, il permet d'accélérer le traitement des données, dans le deuxième il permet de sauvegarder les données sous la forme d'un arbre binaire.

3.2.1.2 *Le R-Tree*

R-trees est une famille d'indexation particulièrement privilégiée dans le cadre des bases de données spatiales. Elle est aussi une généralisation de la famille B-tree (Binary tree en anglais). Elle contient plusieurs variantes comme R*tree [24] et DR-tree [25]. L'indexation des objets multidimensionnels par des rectangles englobant multidimensionnels (appelé aussi des hyper rectangles) est l'idée principale de R-tree. En effet, cette technique a été construite sur un échelonnement de rectangles inclus les uns à l'intérieur des autres. Pour trouver un objet dans R-tree, il faut parcourir tout l'arbre à commencer par la racine et allant jusqu'aux feuilles, une fois arrivé on teste les données. Pour insérer des données, on parcourt l'arbre et on choisit le rectangle qui requiert le plus faible agrandissement pour englober les nouvelles données. La figure 11 montre un exemple d'un arbre R-tree. Si on a le cas de deux rectangles égaux alors on choisit celui qui est le plus petit des deux. Pour une suppression de données, on utilise la même démarche.

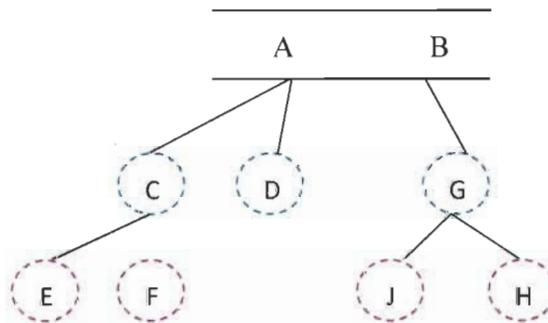
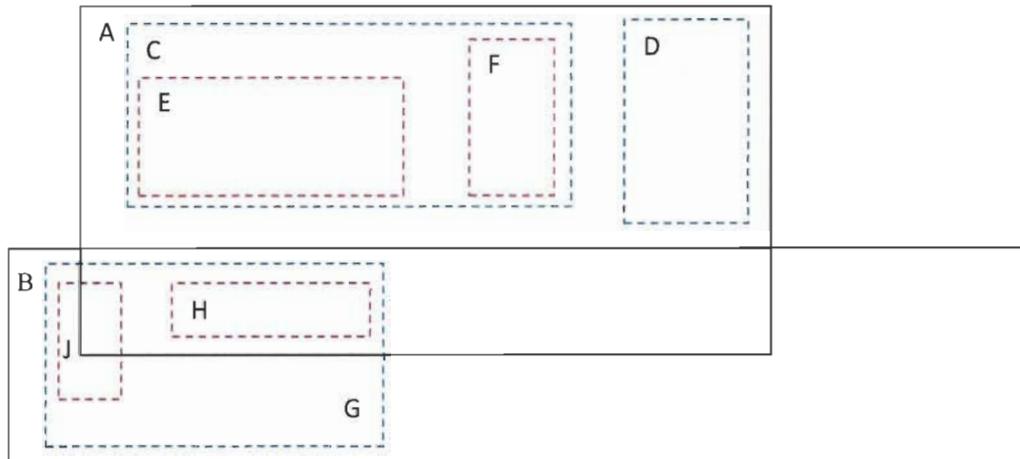


Figure 11 : Exemple d'un R-Tree.

Un arbre R a plusieurs propriétés, parmi elles :

- ✓ Le dernier niveau de l'arbre contient les nœuds feuille, c'est là où sont stockées les données. Chaque nœud feuille comporte au maximum M éléments de données et au minimum m avec $m \leq M/2$ hormis le nœud racine.
- ✓ Tous les nœuds mis à part les nœuds feuilles et le nœud racine ont entre m et M nœuds enfants.

- ✓ Tous les nœuds racines ont au minimum deux fils
- ✓ Tous les nœuds feuilles appartiennent au même niveau

Nous venons de présenter la classification d'images en utilisant la technique des arbres. Pour de plus amples renseignements sur cette technique en général et sur leur application à la classification d'images voir [29] [30] [31] [32] [33]. Dans la section suivante, nous allons voir une autre technique qui est celle des réseaux de neurones.

3.2.2 La technique des réseaux de neurones

D'après Haykin [26], « *un réseau de neurones est un processus distribué de manière massivement parallèle, qui a une propension naturelle à mémoriser des connaissances de façon expérimentale et de les rendre disponibles pour utilisation. Il ressemble au cerveau en deux points:*

- *La connaissance est acquise au travers d'un processus d'apprentissage*
- *Les poids des connexions entre les neurones sont utilisés pour mémoriser la connaissance. »*

À partir de cela on peut dire qu'un réseau de neurones est un ensemble de neurones qui sont interconnectés entre eux dans un seul sens, de l'entrée vers la sortie. La manière de faire la liaison entre un neurone et un autre est de prendre la sortie de ce premier (l'axone), d'affecter un poids synaptique à sa valeur et de rejoindre une entrée d'un autre neurone (les dendrites). Les principales caractéristiques des réseaux sont leurs architectures, leurs algorithmes d'apprentissage et leurs fonctions de leurs éléments.

Dans cette section du chapitre nous allons commencer par définir les deux types de neurones en l'occurrence le neurone biologique et le neurone artificiel, après nous décrirons quelques méthodes de réseaux de neurones qui ont été utilisées pour la classification d'images.

3.2.2.1 *Le neurone biologique*

Dans le domaine de la biologie médicale, un neurone est une cellule nerveuse qui se trouve dans le cerveau et qui est composée de quatre parties principales qui sont : le corps cellulaire qui contient les molécules essentielles à la vie de la cellule ainsi que le noyau du neurone, les dendrites qui reçoivent les signaux en provenance des autres neurones, l'axone qui transmet les signaux sortant vers les autres neurones et finalement la synapse qui permet la connexion entre deux neurones plus particulièrement entre l'axone et les dendrites (la figure 12 schématisé un neurone biologique).

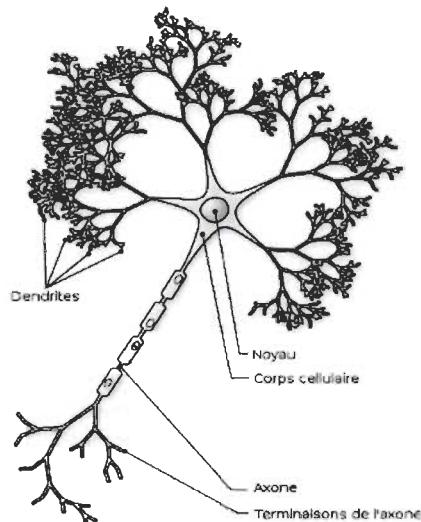


Figure 12 : Schéma d'un neurone biologique [34].

La fonction principale d'un neurone A est de transmettre un signal électrique. En effet, le neurone A est connecté aux neurones qui l'entourent à travers les dendrites puisque celles-ci reçoivent les signaux électriques des neurones en amont. Ces signaux ont été propagés par les axones des neurones environnants. Une fois le neurone A est connecté, les charges vont s'accumuler dans celui-ci jusqu'à ce qu'il dépasse un seuil bien précis. À partir de là, le neurone A transmet un signal électrique à travers son axone vers d'autres neurones.

3.2.2.2 Le neurone formel

Les réseaux de neurones formels (qu'on appelle aussi réseaux de neurones artificiels) ont vu le jour en 1943 et on le doit à Mac Culloch et Pitts [27]. En effet, le neurone formel est une modélisation mathématique et informatique qui reprend les principes du fonctionnement du neurone biologique. En général un neurone formel à plusieurs entrées et une sortie qui correspondent aux dendrites et à l'axone d'un neurone biologique. La figure 13 présente un schéma descriptif du fonctionnement d'un neurone formel.

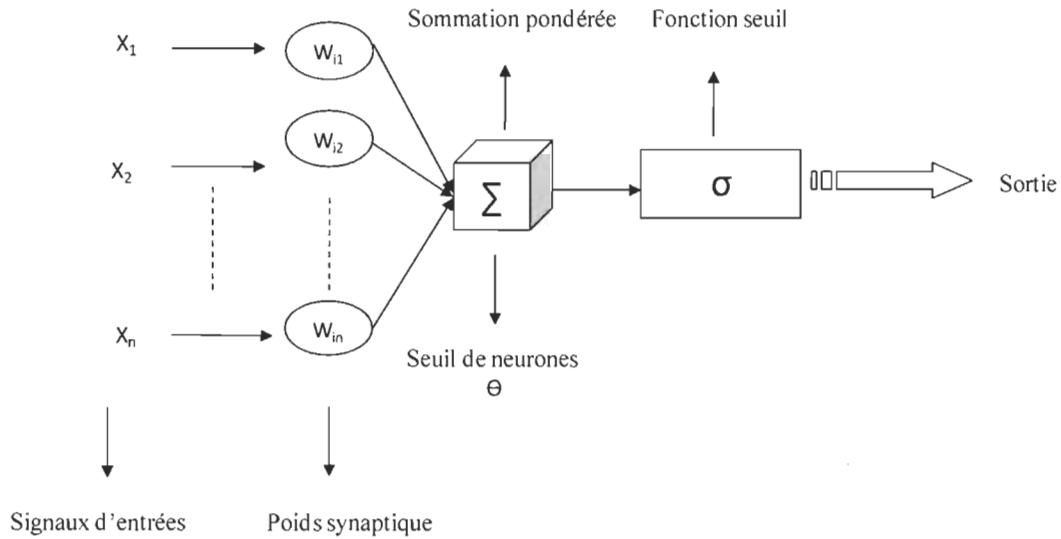


Figure 13 : Schéma d'un neurone formel

Ce neurone permet de calculer la somme pondérée de ses entrées. Cette sommation est définie par :

$$P_i = \sum_{j=1}^n W_{ij} X_j \quad (2.1)$$

Pour calculer la sortie en utilisant la fonction seuil, on compare P_i au seuil Θ . Si la valeur de P_i est plus grande que Θ , le neurone délivre un résultat de sortie qui est égal à 1, sinon il délivre un résultat de sortie nul (égal à 0).

$$Y_i = \sigma(P_i) = \begin{cases} 1, & P_i > \Theta_i \\ 0, & P_i \leq \Theta_i \end{cases} \quad (2.2)$$

3.2.2.3 Les différentes méthodes de réseaux neurones utilisées pour la classification d'images

Actuellement, il existe plusieurs méthodes de réseaux de neurones formels qui sont utilisés dans divers domaines. Parmi ces réseaux on peut citer le perceptron [36], les perceptrons multicouches [36], les réseaux de Hopfield [38], les réseaux de Kohonen, les réseaux de Neurones à Poids Partagés [35], les réseaux de neurones à Fonction radiale de base [36], etc. Tous ces types de réseaux sont appliqués dans différents domaines comme l'intelligence artificielle, le traitement d'images, le traitement de signal, l'évaluation des risques, la prévision et modélisation du marché (cours de monnaies, etc.), etc. Et puisque notre sujet de recherche traite la classification d'images, nous allons présenter dans ce qui suit quelques réseaux célèbres qui ont été utilisés dans l'indexation d'images, parmi eux les réseaux de Hopfield et les perceptrons multicouches.

3.2.2.3.1 Les perceptrons multicouches

Un perceptron multicouche (PMC) est un neurone (vue précédemment) qui est organisé en couche et il en comporte une ou plusieurs couches intermédiaires qu'on appelle couches cachées. En général un PMC possède un nombre d'entrées égales à X et un nombre quelconque de couches. Chaque neurone de chaque couche est relié aux neurones de la couche suivante. L'information se déplace de la couche d'entrée vers la couche de sortie. Un perceptron multicouche ressemble à la figure 14 :

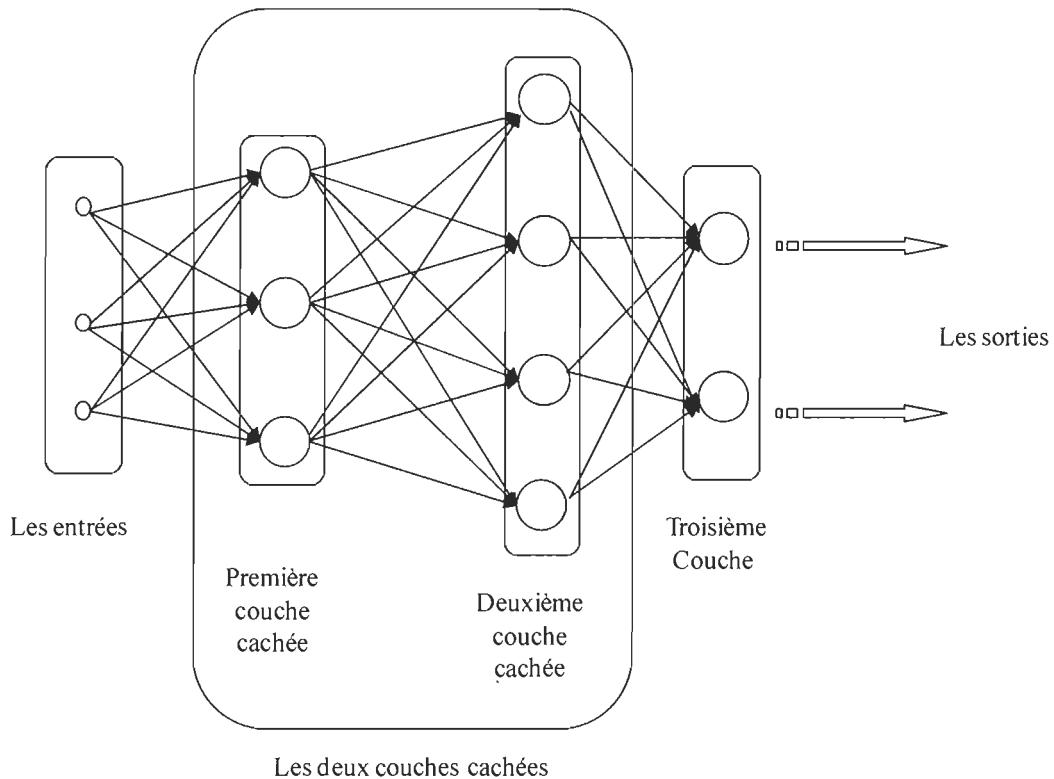


Figure 14 : Schéma d'un perceptron multicouche

D'après la figure 14, on a une couche d'entrée des caractéristiques des images qui est constamment une couche virtuelle, elle est intégrée aux entrées et ne contient pas de neurones. Nous avons des couches cachées qui viennent juste après et qui contiennent par contre des neurones. Et la dernière couche est celle de la sortie et qui contient aussi des neurones. Dans notre cas, on a un PMC à trois entrées, deux couches cachées la première contient trois neurones et la deuxième en contient quatre et une couche de sortie qui renferme deux neurones et qui correspond à la sortie du système. Un perceptron multicouche à X entrées calcule la fonction suivante :

$$Y_i = \sigma(T_i) \text{ avec } T_i = \sum_{i=1}^n W_i X_i + \theta_i \quad (2.3)$$

où :

W_i : est les poids de synapse, Θ_i : est un terme additif, T_i : est la fonction de transfert.

Le PMC utilise l'algorithme d'apprentissage par rétro-propagation [34] (c'est un algorithme qui a été généralisé à partir de la règle de Widrow-Hoff) qui lui permet de trouver le poids des connexions minimisant l'erreur quadratique moyenne.

3.2.2.3.2 Les réseaux de Hopfield

Les réseaux de Hopfield ont été créés en 1982 par le chercheur John Hopfield. Ces réseaux sont des réseaux de neurones récurrents, qui sont interconnectés entre eux et qui forment un cycle dans la structure. En effet, chaque neurone de ce réseau est connecté à tous les autres neurones et c'est là la différence avec les autres réseaux puisque celui-ci n'a pas de neurones d'entrées et de sortie, mais il commence à partir d'un état initial. La figure 15 est une représentation du réseau Hopfield.

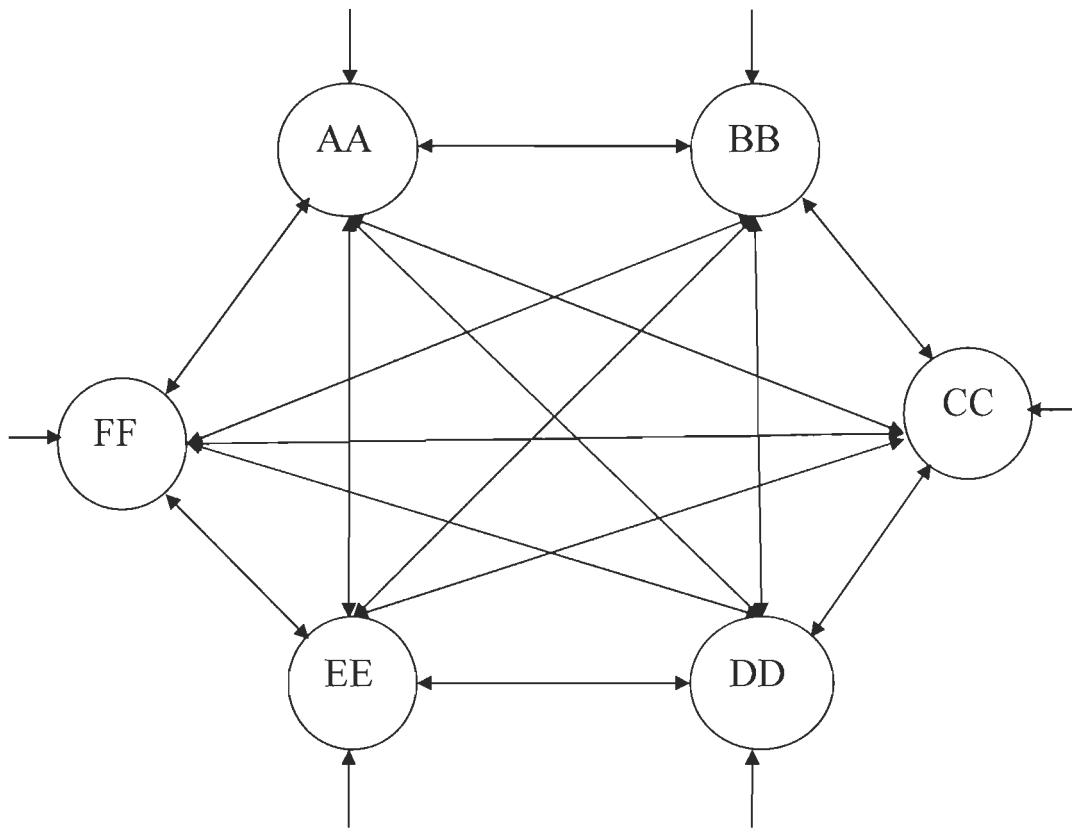


Figure 15 : Réseau de Hopfield.

Un modèle de neurones de Hopfield est composé en général de N neurones (dans notre exemple de 6 neurones) et d'une matrice symétrique de transfert W . L'entrée totale d'un neurone est de la forme :

$$I_i = \sum_j W_{ij} V_j \quad (2.4)$$

Où :

W_{ij} : est le poids de la connexion du neurone i à j , V_j : est l'état du neurone j .

Ce type de réseau répond à une fonction seuil et pour une facilité d'utilisation on considère une fonction seuil plus simple qui est :

$$f(x) = \begin{cases} 1, & \text{si } x > 0 \\ -1, & \text{si } x \leq 0 \end{cases} \quad (2.5)$$

La valeur d'entrée d'un neurone du réseau de Hopfield est égale à -1 ou 1, mais on peut les changer avec des valeurs binaires comme 0 et 1. Ce changement se fait grâce à la modification suivante :

$$A(-1, 1) = 2A(0, 1) - 1 \quad (2.6)$$

Dans le réseau de Hopfield, le processus d'archivage des données ressemble beaucoup à celui d'une mémoire associative et non linéaire c'est-à-dire il permet de trouver une donnée entière stockée à partir de données partielles ou bruitées. En effet, Dans un ordinateur classique, les mémoires sont dites à accès aléatoire, ce qui signifie que l'adresse à laquelle est stockée une information est sans rapport avec celle-ci. Pour avoir accès à l'information complète il faut connaître son adresse exacte, et toute information incomplète est sans intérêt. Dans le cas d'une mémoire associative, une information partielle permet de restituer l'information complète. Pour la récupération des données, on estime alors le poids de la connexion des neurones de façon à minimiser la fonction d'énergie pour arriver à un état d'équilibre. Pour estimer le poids, on utilise la loi Hebb [37].

- La loi de Hebb [17]:

Si deux unités connectées sont actives simultanément, le poids de leur connexion est augmenté ou diminué. R est une constante positive qui représente la force d'apprentissage (learning rate).

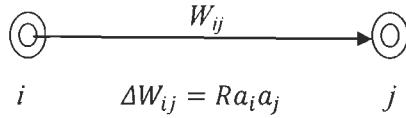


Figure 16 : Loi de Hebb.

où a_i : est le neurone en amant, a_j : est le neurone en aval et $W_{i,j}$: est le poids de la connexion.

Nous venons de voir une brève représentation de la deuxième méthode d’indexation d’images avec ses deux approches les plus connues, maintenant nous allons étudier dans le prochain point la technique statistique.

3.2.3 La technique statistique

On a vu précédemment deux techniques qui nous permettaient de classifier des images dans une base de données. Dans cette partie nous allons parler d’une troisième technique en l’occurrence la technique statistique et plus précisément la méthode de classification basée sur le mélange de mixture de gaussienne vue que notre recherche porte sur celle-ci. Dans ce qui suit, nous allons faire une brève description de cette approche.

Le modèle de mixture de Gaussienne (Gaussian Mixture Model en anglais) est un modèle statistique utilisé dans la classification automatique. Dans le domaine de l’indexation et la recherche d’images par le contenu, le modèle de mixture de Gaussienne est fréquemment utilisé. En effet, on considère un échantillon de données (dans notre cas les données sont les signatures de l’image) qui suit une loi dont la fonction de densité est une densité mélange, mais la loi la plus couramment utilisée est la loi normale dont la fonction de densité est une gaussienne. On parle alors de mélange

gaussien. Son utilité est d'estimer la distribution de variables aléatoires en les modélisant comme une somme de plusieurs gaussiennes.

Considérons un échantillon de données $y_j = (j = 1, 2, \dots; y_j \in \mathbb{R}^p)$ (dans notre cas l'échantillon sont les caractéristiques d'images) qui est distribué selon un mélange discret d'un nombre connu k de composantes multivariées normales. Plus exactement :

$$(y_j | \mu, \Sigma, \theta) \sim \mathcal{N}(\mu_i, \Sigma_i) \quad (2.7)$$

Avec :

$\mu = (\mu_1, \mu_2, \dots, \mu_k)$: Les vecteurs de moyennes

$\Sigma = (\Sigma_1, \Sigma_2, \dots, \Sigma_k)$: Les matrices de variances-covariances

$\theta = (\theta_1, \theta_2, \dots, \theta_k)$: Les probabilités de classification

Maintenant nous allons incorporer des variables de classification x_j telles que $x_j = i$ qui implique que y_j est classé dans le groupe i . Ce qui nous donne :

$$(y_j | \mu, \Sigma, \theta, x_j = i) \sim \mathcal{N}(\mu_i, \Sigma_i) \quad (2.8)$$

En supposant que x_j et θ sont indépendants et ont pour densité $P(x_j = i | \theta) = \theta_i$. Maintenant il s'agit de déterminer le meilleur paramètre de θ , en utilisant le même échantillon de données $y_j = (j = 1, 2, \dots; y_j \in \mathbb{R}^p)$ suis une loi $f(y_j, \theta)$ paramétrée par θ . Pour cela, on utilise l'algorithme du Maximum de vraisemblance EM donnée dans ce cas, par :

$$L(y, \theta) = \sum_{i=1}^n \log(f(y_i, \theta)) \quad (2.9)$$

Une fois les paramètres estimés, il s'agit maintenant d'attribuer chaque image à la classe appropriée. On utilise à cet effet la règle de Bayes. À partir de là, on affecte chaque image à sa propre classe dont sa probabilité a posteriori est la plus grande.

Nous venons de voir ci-haut les différentes techniques utilisées pour indexer une collection d'images, dans la section suivante nous allons faire une description de quelques moteurs de recherche qui ont utilisé les techniques que nous venons de citer.

3.3 Quelques moteurs de recherche existants utilisant la méthode d'indexation

Depuis le début des années 90, l'indexation et la recherche d'images par le contenu est devenu un domaine de recherche très actif. Ce qui a engendré l'émergence de plusieurs systèmes académiques et commerciaux. Malgré les imperfections des systèmes de recherche d'images par le contenu, plusieurs ont vu le jour et quelques-uns sont disponibles sur internet. La majorité de ces systèmes utilisent des méthodes de recherche essentiellement effectuée sur des exemples fournis par l'utilisateur ou par une navigation dans la banque de données d'images. Ces exemples peuvent être soit une image, soit un croquis, soit par une image et du texte. Dans la suite de ce dernier point nous allons tout d'abord présenter les principaux systèmes mis au point, ensuite nous allons illustrer quelques domaines d'application concrets de l'utilisation de l'indexation et de la recherche d'images par le contenu.

3.3.1 QBIC

Le système QBIC (Query By Image Content) [45] a été créé au centre de recherche Almaden d'IBM. Il permet à un utilisateur de faire une recherche d'images sur

l'ensemble des caractéristiques, dont la couleur, la forme et la texture aussi. Il donne la main à l'utilisateur de faire une recherche textuelle à l'aide de mots clés. Dans QBIC la recherche d'images est basée sur une image exemple parmi une palette. En plus de cela, QBIC permet à l'utilisateur de formuler une requête directement. En d'autres termes, il peut faire un dessin à l'aide des outils que propose QBIC et choisir la couleur et la texture parmi une palette. C'est l'un des rares systèmes qui tiennent compte de l'indexation dimensionnelle [46]. En effet, QBIC utilise la méthode des R-*tree pour indexer sa base de données d'images ce qui va lui permettre d'améliorer et de rendre plus efficace sa recherche [47]. Une fois l'image exemple choisie, le système extrait ses caractéristiques et la compare à celle de la base de données. La dernière version a vu l'intégration d'une nouvelle technique d'indexation plus efficace et le moyen de faire une recherche d'images en niveaux de gris ainsi que l'amélioration de l'interface graphique [48]. La version démo de QBIC en ligne est à l'adresse suivante : <http://wwwqbic.almaden.ibm.com/>.

3.3.2 Virage

Virage a été créée par Jain Ramesh est le directeur du laboratoire Visualcom à l'Université de Californie, San Diego en avril 1994. Tout comme QBIC, Virage permet de faire une recherche d'images à l'aide des caractéristiques de la couleur, de la texture, de la forme... Mais il va encore plus loin que QBIC puisque Virage donne la possibilité de rechercher des images en autorisant la combinaison de plusieurs types de requêtes ce qui permet à l'utilisateur d'affiner ses choix de recherche.

Virage est disposé en une série de modules indépendants, ce qui permet aux développeurs de construire des systèmes pour leurs propres programmes. Cela rend plus facile l'extension du système en construisant de nouveaux types de requête, ou d'autres modules pour le traitement d'images spécialisés [49]. Le site web d'Altavista est un

exemple de la technologie Virage. Altavista permet aux internautes de faire une recherche d'image par similarité du contenu. La version démo de Virage est disponible en ligne est à l'adresse suivante : www.virage.com.

3.3.3 VisualSeek

VisualSeek [50] est un système qui est orienté pour le web et qui a été développé à l'Université Columbia de New York. Il a été programmé en langage *JAVA* pour permettre l'indépendance des plates-formes client et son accessibilité sur Internet. VisualSeek utilise les caractéristiques de base de l'image à savoir la couleur, la texture et la forme. Mais sa particularité est qu'il ajoute aux caractéristiques de couleur et de texture une contrainte spatiale sur les régions. Pour émettre une requête, l'utilisateur dessine un certain nombre de régions, les positions et les dimensions de la grille et choisit une couleur pour chaque région. L'utilisateur peut même marquer soit les limites de l'emplacement et la taille, soit les relations spatiales entre les régions. Ensuite, il affiche les résultats obtenus. Pour accélérer et faciliter la recherche d'images, ils ont également programmé l'algorithme de l'arbre binaire pour indexer la base de données d'images. Le système permet aussi de faire une recherche d'images en saisissant une requête qui comprend à la fois un dessin et des mots-clés.

3.3.4 PhotoBook

Le système Photobook [51] a été créé à l'université du Massachusetts et il est fonctionnel sous le système UNIX/Linux. Photobook permet de saisir une requête qui est formulée à partir d'une image exemple. Après l'utilisateur doit choisir parmi les trois caractéristiques (texture, forme et visage) pour effectuer sa recherche. Certaines applications ont utilisé avec succès le système Photobook, et ce pour l'extraction de textures d'images, des formes et des visages humains. Les versions récentes de

Photobook permet aux utilisateurs pour sélectionner la caractéristique la plus adéquate pour la recherche d'images à partir d'un large éventail de solutions de rechange [52].

Nous venons de décrire dans cette section quelques moteurs de recherche qui utilisent ces différentes méthodes. Celles-ci ne permettent pas d'obtenir de bons résultats. Ce qui nous ramène à la prochaine section dans laquelle nous allons critiquer les méthodes d'indexation qu'on a vue dans ce deuxième chapitre.

3.4 Critique générale des méthodes d'indexation

Maintenant que nous avons vu les différentes techniques les plus célèbres qui permettent d'indexer une base de données d'images par le contenu ainsi que certains moteurs de recherche, nous allons discuter dans cette ultime partie du chapitre des inconvénients rencontrés par ces dernières et qui ne permettent pas de bien modéliser les données.

Pour indexer une base de données d'images, certains systèmes comme ImageRover [40], ImageScape [39] emploient la technique d'arbre, mais elle présente certains inconvénients comme :

- ✓ Elle a été conçue non pas pour des questions de similitude utilisées dans la recherche d'images, mais sur des questions traditionnelles de base de données [21].

- ✓ Elle utilise comme mesure de similarité, les distances euclidiennes ce qui n'est pas tout le temps convenable pour la recherche d'images par le contenu.
- ✓ Elle se dégrade lorsque la dimension des descripteurs augmente « *ce qu'on appelle le phénomène de la malédiction de la dimension* ».
- ✓ Elle ne s'adapte pas avec le modèle probabiliste dans le domaine de l'indexation et de la recherche d'images.

Sauf qu'il y a d'autres systèmes qui utilisent d'autres méthodes de classification comme [42] qui emploie la technique des réseaux de neurones et cette dernière comporte aussi un certain nombre d'inconvénients par exemple :

- ✓ Elle ne permet pas d'interpréter les modèles construits vu leur complexité. En cas d'erreur, il est impossible d'identifier la cause.
- ✓ Elle n'est pas robuste, elle est très dangereuse [43].

Il y a aussi [44] qui utilise la troisième approche en l'occurrence la mixture de Gaussiennes. Cette approche présente aussi des désavantages tels que :

- ✓ Elle présente des problèmes qui apparaissent souvent dans l'apprentissage de modèles de mélange, pour des données en « grande » dimension qu'on appelle « *le phénomène de la malédiction de la dimension* ».
- ✓ Le nombre de paramètres augmente très vite avec le nombre de classes.

- ✓ Elle utilise beaucoup de mémoire.
- ✓ Les calculs sont plus longs et plus complexes.
- ✓ On présume que les caractéristiques des images obéissent à des lois normales et on se limite à estimer les paramètres de ces lois en utilisant des modèles de mixtures de gaussiens. Or, une exploration préliminaire nous a permis de voir que dans la plupart des cas, cette hypothèse de normalité n'est pas vérifiée, ce qui limite énormément la validité des résultats obtenus avec les GMM (nous allons détailler ces résultats dans le chapitre 3).

Dans la partie suivante, nous allons présenter nos conclusions après un survol sur les différentes méthodes d'indexation aussi sur certains moteurs de recherche d'images qui les emploie ainsi que leurs désavantages.

3.5 Conclusion

Il existe une multitude de techniques d'indexation d'images par le contenu. Nous avons présenté dans ce chapitre certaines de ces méthodes les plus couramment utilisées dans la recherche d'images, quelques moteurs qui utilisent ces approches ainsi que leurs inconvénients. Nous nous sommes plus concentrés sur la technique statistique vue que notre recherche porte sur cette dernière. Cela va nous permettre de présenter dans le chapitre suivant notre approche que nous allons utiliser pour modéliser nos données en l'occurrence le modèle de copules archimédiennes.

3- Chapitre1

4 NOTRE TRAVAIL

5.1 Introduction

Dans le deuxième chapitre, nous avons parlé des différentes méthodes qui ont été utilisées pour effectuer l'indexation en l'occurrence la méthode basée sur la technique des arbres, celle qui est basée sur la technique des réseaux de neurones ainsi que la méthode basée sur les techniques statistiques. Cette dernière est l'objet principal sur lequel notre recherche s'est basée. Par conséquent, nous allons consacrer une partie de ce chapitre à parler de l'approche statistique, plus précisément de la mixture de Gaussienne que nous avons décrie brièvement dans le chapitre précédent.

Dans la plupart des travaux existants, on présume que les caractéristiques des images obéissent à des lois normales et on se limite à estimer les paramètres de ces lois en utilisant des modèles de mixtures de Gaussiennes (GMM). Or, une exploration préliminaire nous a permis de voir que dans la plupart des cas, cette hypothèse de normalité n'est pas vérifiée, ce qui limite énormément la validité des résultats obtenus avec les GMM. Nous nous sommes donc attaqués à cette problématique en réalisant un nouvel outil d'indexation d'images qui permet de mieux indexer les images d'une base de données en utilisant les modèles de copules archimédiennes qui vont nous permettre de trouver les lois de probabilités qui épousent mieux nos données.

Dans ce chapitre, nous allons détailler dans la section 3.2 la méthode de mixture de Gaussienne puis dans la section 3.3 nous allons critiquer cette dernière, ensuite dans la section 3.4 nous proposons une nouvelle approche basée sur les copules. Finalement nous présentons une conclusion.

5.2 La méthode de mixture de Gaussienne

Pour comprendre cette méthode de classification, nous allons tout d'abord commencer par certains rappels statistiques, par la suite nous allons définir le principe de la méthode et enfin nous donnerons l'algorithme qui permettra de classifier ces images.

5.2.1 Rappel statistique et principe de la méthode de classification basée sur le mélange gaussien

On a vu précédemment deux techniques qui nous permettaient de classifier des images dans une banque de données. Dans cette partie nous allons parler d'une troisième technique en l'occurrence la technique statistique et plus précisément la méthode de classification basée sur le mélange de mixture de gaussiennes. Pour comprendre cette méthode de classification, nous allons tout d'abord commencer par certains rappels statistiques, par la suite nous allons définir le principe de la méthode et enfin nous donnerons l'algorithme qui permettra de classifier les images.

5.2.1.1 *Distribution gaussienne multivariée*

La loi normale est le modèle le plus utilisé en statistique. Elle possède l'avantage d'approcher des distributions de somme de variables indépendantes. Elle fut découverte par Karl Friedrich Gauss (1777-1855).

On dit qu'un vecteur aléatoire $X = (X_1, \dots, X_n)$ suit une loi normale multivariée si sa fonction de densité est de la forme :

$$f(x) = \frac{1}{\left[(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}} \right]} e^{-\left(\frac{1}{2}\right)(x-\mu)\Sigma^{-1}(x-\mu)} \quad (3.1)$$

où

- ✓ $X = (X_1, \dots, X_n)$ est un vecteur aléatoire
- ✓ $\mu = (\mu_1, \dots, \mu_n)$ désigne le vecteur des espérances
- ✓ Σ est la matrice de variance covariance qui fait intervenir toutes covariances entre les composantes X_i du vecteur aléatoire $X = X_1, \dots, X_n$ c'est-à-dire que :

$$\Sigma(X) = [\text{cov}(X_i, X_j)]_{i,j=1,\dots,n} \quad (3.2)$$

Le vecteur aléatoire X est alors de moyenne μ et de matrice de covariance Σ , et on écrit $X \sim \mathcal{N}_n(\mu, \Sigma)$. Notons que μ et Σ sont les paramètres qui gouvernent la distribution gaussienne multivariée.

5.2.1.2 Distribution de mélange gaussien

Soient f_1, \dots, f_k des densités de lois gaussiennes multivariées. Notons par (μ_i, Σ_i) les paramètres associés à f_i . Soient $P = (p_1, \dots, p_k)$ des probabilités telles que la somme de $(p_1 + \dots + p_k) = 1$. La densité de la distribution de mélange de gaussiennes est donnée par :

$$f(x) = p_1 f_1(x) + p_2 f_2(x) + \dots + p_k f_k(x) \quad (3.3)$$

où $x = (x_1, \dots, x_n)$

5.2.1.3 Formule de Bayes

La formule de Bayes permet de calculer des probabilités a posteriori. Une probabilité a posteriori est une probabilité conditionnelle étant donnée une information obtenue à partir de l'observation d'un événement. Dans la loi de Bayes, il y a deux hypothèses :

- On dispose des événements A_1, \dots, A_k formant une partition de S telle que les probabilités a priori $P(A_1), \dots, P(A_k)$ sont connues.
- On observe un événement B pour lequel les probabilités conditionnelles $P(B|A_1), \dots, P(B|A_k)$ sont connues.

On s'intéresse alors à évaluer les probabilités a posteriori de $P(A_1|B), \dots, P(A_k|B)$ après la nouvelle information donnée par B . La formule de Bayes est de la forme :

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \quad (3.4)$$

avec $i = 1, \dots, n$.

Mais elle peut s'écrire d'une autre façon qui est :

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^n P(B|A_j)P(A_j)} \quad (3.5)$$

avec $i = 1, \dots, n$.

5.2.1.4 La méthode de Maximum de Vraisemblance

On considère une population caractérisée par une variable aléatoire X de densité $f_\theta(x)$. On prélève un échantillon x_1, \dots, x_n . La vraisemblance représente la probabilité d'observer x_1, \dots, x_n . Elle est définie par:

$$L(x_1, \dots, x_n, \theta) = \prod_{i=1}^n f_\theta(x_i) \quad (3.6)$$

L'objectif de la méthode du maximum de vraisemblance est d'utiliser un échantillon observé x_1, \dots, x_n afin d'estimer le paramètre inconnu θ . L'idée consiste donc à estimer le paramètre θ par la valeur qui maximise la vraisemblance (probabilité d'observer l'échantillon). C'est-à-dire, on cherche la valeur $\hat{\theta}$ solution du problème d'optimisation.

$$\max L = (x_1, \dots, x_n, \hat{\theta}) \quad (3.7)$$

La valeur $\hat{\theta}$ est appelée estimation du maximum de vraisemblance.

5.2.1.5 Le principe de la méthode de classification basée sur le mélange gaussien

Dans la plupart des travaux existants, on présume que les caractéristiques des images obéissent à des lois normales et on se limite à estimer les paramètres de ces lois en utilisant des modèles de mixtures de Gaussiennes (GMM). Nous supposons qu'on a une collection d'images qui provient d'un nombre de groupes inconnu qu'il faut retrouver. Chacune de ces images est représentée par un point dans un espace multidimensionnel comme présenter dans la figure 17.

VC : Vecteur Caractéristique de l'image

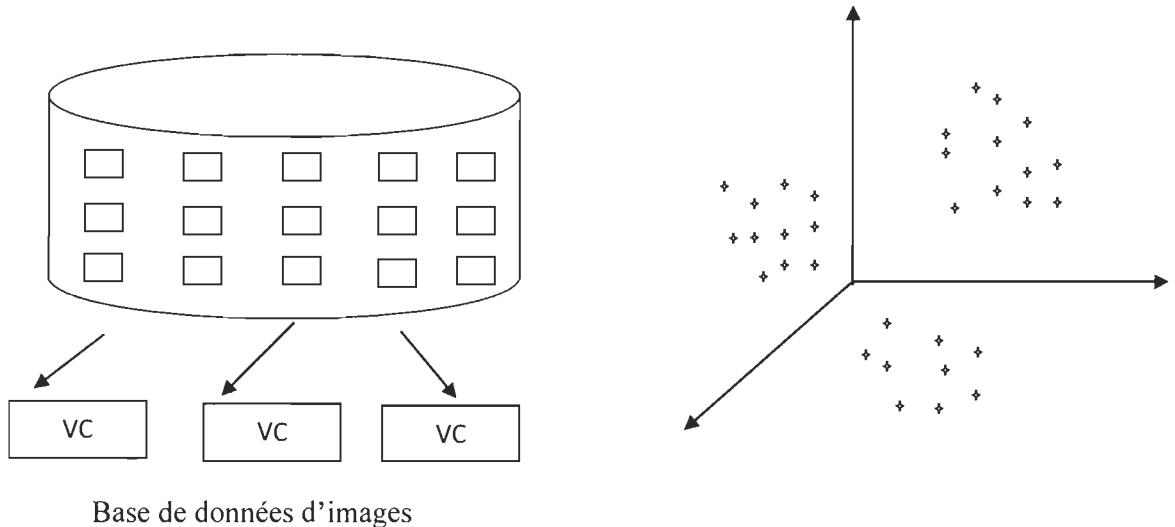


Figure 17 : Représentation d'une image dans un espace bidimensionnel

On présume que l'on connaît le nombre de classes des images. Alors soit r une image quelconque, d'après la formule de probabilité totale, on a :

$$P(r) = \sum_{k=1}^K P(k)P(r|k). \quad (3.8)$$

où k est le nombre de classe

Ceci nous ramène à déterminer la vraisemblance pour toute la collection d'images. Cette vraisemblance est déduite à partir de l'équation (3.8). On pourra écrire alors :

$$L = \prod_{r=1}^N P(r) = \prod_{r=1}^N \left(\sum_{k=1}^K P(k)P(r|k) \right). \quad (3.9)$$

Dans cette approche de mixture de gaussienne, on cherche à déterminer les paramètres de cette dernière qui maximisent la vraisemblance L donc ils maximisent $\log L$. À partir de l'équation (3.9) on a :

$$\log L = \sum_{r=1}^N \log \left(\sum_{k=1}^K P(k)P(r|k) \right) \quad (3.10)$$

Pour chaque classe k , on va donc estimer le vecteur Θ_k de paramètres P_k , μ_k et Σ_k . Afin de déterminer Θ_k on utilise l'algorithme EM (*en anglais Expectation Maximization algorithm*) qui a été proposé en 1977 par Dempster et al. [8]. L'objectif de cet algorithme est de permettre de trouver par maximum de vraisemblance les paramètres du modèle et de distribuer les données dans des classes. On pose :

$$\phi = \log L = \sum_{r=1}^N \log \left(\sum_{k=1}^K P(k)P(r|k) \right). \quad (3.11)$$

Nous devons alors tirer la probabilité du log de la contrainte de l'équation (3.11) par rapport à chacun de nos paramètres, puis poser la dérivée égale à zéro pour obtenir la valeur optimale de chaque paramètre. Nous remarquons que la première partie de la dérivation est commune à tous les autres paramètres P_k, μ_k et Σ_k . Nous allons alors commencer par donner la partie commune de la dérivation, puis les informations sur la partie restante pour chaque paramètre séparément.

La partie commune de la dérivation :

Pour la classe numéro l on a :

$$\frac{\partial}{\partial \theta_l} \phi = \log \sum_{r=1}^N \frac{\partial}{\partial \theta_l} \log \sum_{k=1}^K P(k)P(r|k) \quad (3.12)$$

C'est-à-dire

$$\frac{\partial}{\partial \theta_l} \phi = \frac{1}{\sum_{k=1}^K P(k)P(r|k)} \frac{\partial}{\partial \theta_l} \sum_{k=1}^K P(k)P(r|k). \quad (3.13)$$

Et sachant que

$$P(r) = \sum_{k=1}^K P(k)P(r|k). \quad (3.14)$$

On peut en déduire en appliquant la formule de Bayes vue précédemment que

$$P(r) = \frac{P(r|\theta_l)P(l)}{P(\theta_l|r)} \quad (3.15)$$

Alors l'équation (3.12) s'écrit

$$\frac{\partial}{\partial \theta_l} \Phi = \sum_{r=1}^N \frac{P(\theta_l|r)}{P(r|\theta_l)P(l)} \frac{\partial}{\partial \theta_l} \sum_{k=1}^K P(k)P(r|\theta_k) \quad (3.16)$$

avec $P(r|k) = P(r|\theta_k)$

En plus, en connaissant que $\sum_{k=1}^K P(k)P(r|k)$ dépend de Θ_l uniquement quand

$k = l$, On pourra écrire

$$\frac{\partial}{\partial \theta_l} \Phi = \sum_{r=1}^N \frac{P(\theta_l|r)}{P(r|\theta_l)P(l)} \frac{\partial}{\partial \theta_l} [P(l)P(r|\theta_l)]. \quad (3.17)$$

➤ Si $\theta_l \in \{\mu_l, \Sigma_l\}$ l'équation (3.17) devient :

$$\frac{\partial}{\partial \theta_l} \Phi = \sum_{r=1}^N \frac{P(\theta_l|r)}{P(r|\theta_l)P(l)} P(l) \frac{\partial}{\partial \theta_l} P(r|\theta_l) \quad (3.18)$$

$$= \sum_{r=1}^N \frac{P(\theta_l|r)}{P(r|\theta_l)} \frac{\partial}{\partial \theta_l} P(r|\theta_l). \quad (3.19)$$

➤ Si $\theta_l = P_l$ l'équation (3.17) devient :

$$\frac{\partial}{\partial \theta_l} \Phi = \sum_{r=1}^N \left(\frac{P(\theta_l|r)}{P(r|\theta_l)P(l)} P(r|\theta_l) \right) \quad (3.20)$$

$$= \frac{1}{P(l)} \sum_{r=1}^N P(\theta_l|r). \quad (3.21)$$

Maintenant, nous avons besoin de calculer la moyenne, la covariance et la probabilité a priori séparément.

Calcul de μ_l :

D'après l'équation (3.19) :

$$\frac{\partial}{\partial \mu_l} \Phi = \sum_{r=1}^N \left(\frac{P(\theta_l|r)}{P(r|\theta_l)} \frac{\partial}{\partial \mu_l} P(r|\theta_l) \right). \quad (3.22)$$

mais dans le cas d'une gaussienne, on a :

$$\frac{\partial}{\partial \mu_l} P(r|\theta_l) = P(r|\theta_l) \Sigma_l^{-1} (r - \mu_l). \quad (3.23)$$

On remplace dans l'équation (3.22) pour obtenir :

$$\frac{\partial}{\partial \mu_l} \Phi = \sum_{r=1}^N \left(\frac{P(\theta_l|r)}{P(r|\theta_l)} P(r|\theta_l) \Sigma_l^{-1} (r - \mu_l) \right) \quad (3.24)$$

$$= \sum_{r=1}^N (P(\theta_l|r) \Sigma_l^{-1} (r - \mu_l)) \quad (3.25)$$

maintenant, μ_l optimale vérifie que $\frac{\partial}{\partial \mu_l} \Phi = 0$ donc

$$\frac{\partial}{\partial \mu_l} \Phi = 0 \Leftrightarrow \sum_{r=1}^N \left(P(\theta_l|r) \Sigma_l^{-1} (r - \mu_l) \right) = 0 \quad (3.26)$$

$$\Leftrightarrow \Sigma_l^{-1} \sum_{r=1}^N (P(\theta_l|r) (r - \mu_l)) = 0 \quad (3.27)$$

$$\Leftrightarrow \sum_{r=1}^N P(\theta_l|r) (r - \mu_l) = 0 \quad (3.28)$$

$$\Leftrightarrow [\sum_{r=1}^N P(\theta_l|r)] r = \mu_l \sum_{r=1}^N P(\theta_l|r) \quad (3.29)$$

$$\Leftrightarrow \mu_l = \frac{[\sum_{r=1}^N P(\theta_l|r)] r}{\sum_{r=1}^N P(\theta_l|r)}. \quad (3.30)$$

Calcul de Σ_l :

D'après l'équation (3.19)

$$\frac{\partial}{\partial \Sigma_l} \Phi = \sum_{r=1}^N \left(\frac{P(\Theta_l|r)}{P(r|\Theta_l)} \frac{\partial}{\partial \Sigma_l} P(r|\Theta_l) \right). \quad (3.31)$$

mais dans le cas d'une gaussienne, on a :

$$\frac{\partial}{\partial \Sigma_l} P(r|\Theta_l) = \sum_{r=1}^N \left(\frac{P(r|\Theta_l)}{2\Sigma_l^2} [(r - \mu_l)(r - \mu_l)^T - \Sigma_l] \right) \quad (3.32)$$

On remplace dans l'équation (3.31) pour obtenir

$$\frac{\partial}{\partial \Sigma_l} \Phi = \sum_{r=1}^N \left(\frac{P(\Theta_l|r)}{P(r|\Theta_l)} \frac{P(r|\Theta_l)}{2\Sigma_l^2} [(r - \mu_l)(r - \mu_l)^T - \Sigma_l] \right) \quad (3.33)$$

$$= \sum_{r=1}^N \left(\frac{P(\Theta_l|r)}{2\Sigma_l^2} [(r - \mu_l)(r - \mu_l)^T - \Sigma_l] \right) \quad (3.34)$$

$$= \frac{1}{2\Sigma_l^2} \sum_{r=1}^N (P(\Theta_l|r) [(r - \mu_l)(r - \mu_l)^T - \Sigma_l]) \quad (3.35)$$

Maintenant, Σ_l optimal vérifie que $\frac{\partial}{\partial \Sigma_l} \Phi = 0$ donc

$$\sum_{r=1}^N (P(\Theta_l|r) [(r - \mu_l)(r - \mu_l)^T - \Sigma_l]) = 0 \quad (3.36)$$

$$\Leftrightarrow \sum_{r=1}^N P(\Theta_l|r) [(r - \mu_l)(r - \mu_l)^T] = \Sigma_l \sum_{r=1}^N (P(\Theta_l|r)) \quad (3.37)$$

$$\Leftrightarrow \Sigma_l = \frac{\sum_{r=1}^N (P(\Theta_l|r) [(r - \mu_l)(r - \mu_l)^T])}{\sum_{r=1}^N (P(\Theta_l|r))}. \quad (3.38)$$

Calcul de P_l :

D'après l'équation (3.21) on a :

$$\frac{\partial}{\partial P_l} = \frac{1}{P_l} \sum_{r=1}^N P(\Theta_l|r) \quad (3.39)$$

On rajoute la contrainte suivante

$$\sum_{k=1}^K P(k) = 1 \Leftrightarrow 1 - \sum_{k=1}^K P(k) = 0. \quad (3.40)$$

En appliquant le log de la vraisemblance avec contrainte, on a :

$$\Phi_c = \sum_{r=1}^N [\log \sum_{k=1}^K P(k) P(r|\Theta_k)] + \lambda (1 - \sum_{k=1}^K P(k)). \quad (3.41)$$

Pour la classe numéro l on a :

$$\frac{\partial}{\partial P_l} \Phi_c = \frac{1}{P_l} \sum_{r=1}^N P(\Theta_l|r) + (-\lambda). \quad (3.42)$$

Maintenant, la solution optimale pour P_l vérifie :

$$\frac{\partial}{\partial P_l} \Phi_c = 0 \Leftrightarrow \frac{1}{P_l} \sum_{r=1}^N P(\Theta_l|r) = \lambda \quad (3.43)$$

$$\Leftrightarrow \frac{\sum_{r=1}^N P(\Theta_l|r)}{P_l} = \lambda \Leftrightarrow P_l = \frac{\sum_{r=1}^N P(\Theta_l|r)}{\lambda} \quad (3.44)$$

$$\Leftrightarrow P_l = \frac{1}{\lambda} \sum_{r=1}^N P(\Theta_l|r). \quad (3.45)$$

On va maintenant essayer de se débarrasser du paramètre λ :

On sait déjà que $\sum_{k=1}^K P_k = 1$ on remplace $P(k)$ par sa valeur de l'équation (3.12) pour obtenir

$$\sum_{k=1}^K \left(\frac{1}{\lambda} \sum_{r=1}^N P(\Theta_k|r) \right) = 1 \Leftrightarrow \frac{1}{\lambda} \sum_{k=1}^K \sum_{r=1}^N P(\Theta_k|r) = 1. \quad (3.46)$$

On va intervertir les deux sommes

$$\frac{1}{\lambda} \sum_{r=1}^N \sum_{k=1}^K P(\Theta_k|r) = 1. \quad (3.47)$$

Or on sait que $\sum_{k=1}^K P(\Theta_k|r) = 1$ ainsi on aura :

$$\frac{1}{\lambda} \sum_{r=1}^N 1 = 1 \Leftrightarrow \frac{N}{\lambda} = 1 \Leftrightarrow \lambda = N. \quad (3.48)$$

Finalement on remplace λ dans l'équation (3.12) par sa valeur pour obtenir

$$P_l = \frac{1}{N} \sum_{r=1}^N P(\Theta_l|r). \quad (3.49)$$

5.2.1.6 L'algorithme de la méthode de classification basée sur le mélange gaussien

L'algorithme de la méthode de classification d'images basée sur le mélange gaussien se déroule comme suit :

- i. Initialiser P_k, μ_k et Σ_k pour chaque K (en utilisant le K-means par exemple)
- ii. Calculer $P_k, P(r|\Theta_k)$ et $P(r)$ pour toute image r et pour toute classe k
- iii. En déduire $P(\Theta_k|r)$ pour chaque image r et chaque classe k
- iv. Utiliser les équations suivantes :

$$\mu_l = \frac{[\sum_{r=1}^N P(\Theta_l|r)]r}{\sum_{r=1}^N P(\Theta_l|r)} \quad (3.50)$$

$$\Sigma_l = \frac{\sum_{r=1}^N P(\Theta_l|r) [(r - \mu_l)(r - \mu_l)^T]}{\sum_{r=1}^N P(\Theta_l|r)} \quad (3.51)$$

$$P_l = \frac{1}{N} \sum_{r=1}^N P(\Theta_l|r) \quad (3.52)$$

Pour calculer P_k, μ_k et Σ_k pour chaque classe k.

v. Revenir à l'étape ii. Tant qu'on n'a pas convergé.

Cet algorithme nous permet d'obtenir deux types de résultat :

- Le premier c'est la création de l'index Hard : il permet d'affecter chaque image r à la classe l tant que $P(r|\theta_l)$ est maximal par rapport aux classes c'est-à-dire $P(r|\theta_l) > P(r|\theta_k)$ pour tout k différent de l .
- Le deuxième c'est la création de l'index Soft : il permet de classer une image dans plusieurs classes, mais avec des probabilités différentes.

Nous venons de voir une vue d'ensemble sur la méthode de mixture de gaussiennes, maintenant nous allons tout d'abord extraire les signatures des images, ensuite nous allons prendre leurs composantes une à une et enfin nous allons essayer de trouver la loi de probabilité qui modélise le mieux chaque composante.

5.3 Critique de la GMM

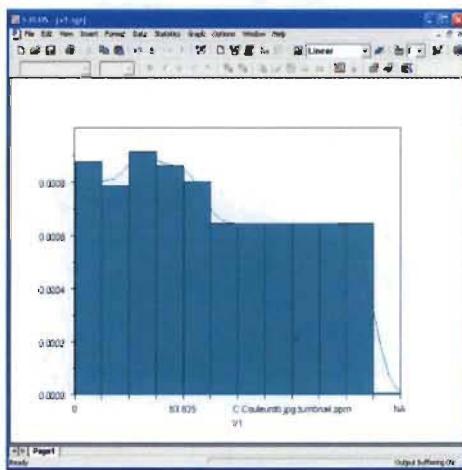
Pour voir si les signatures des images suivent des lois normales, nous avons commencé par extraire les caractéristiques d'une petite banque de données d'images. Les caractéristiques qu'on a extraites de chaque image sont les suivants :

- Les caractéristiques de la couleur qui sont représentées par les histogrammes de couleur RGB (Red, Green, Blue) et les histogrammes de couleur HSV (Hue, Saturation, Value).
- Les caractéristiques de la texture qui sont représentées par la matrice de cooccurrence.

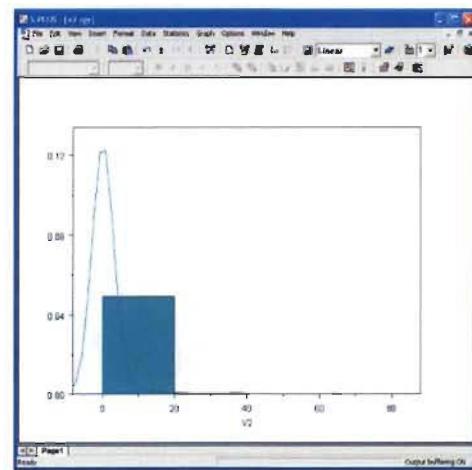
Une fois ces signatures extraites, nous avons pris leurs composantes une à une et nous avons essayé de trouver s'ils suivent une loi normale. On a obtenu les résultats suivants (on a seulement pris les 6 premières composantes de l'histogramme de couleur RGB et les deux premières composantes de la matrice de cooccurrence) :

✓ Les composantes de la couleur RGB

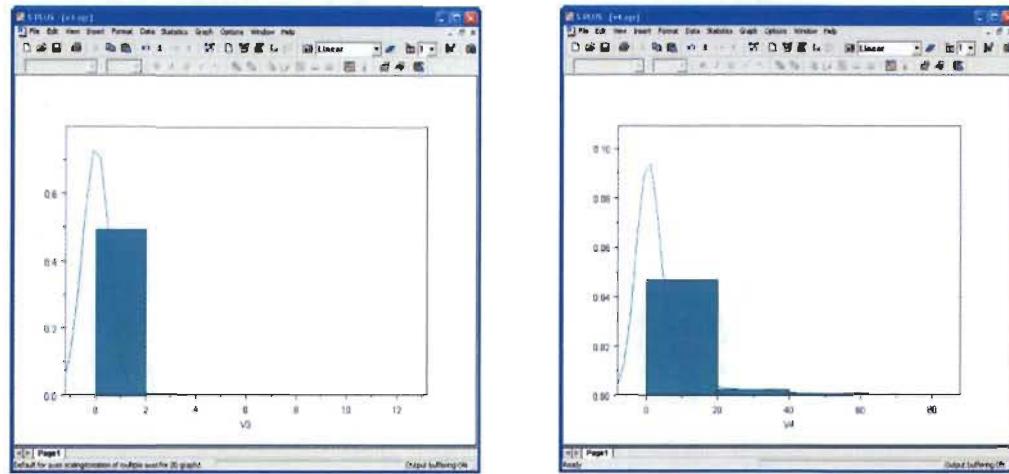
Les caractéristiques de la couleur RGB sont composées de 27 marginales (composantes), on a présenté alors les 6 premières marginales ci-dessous.



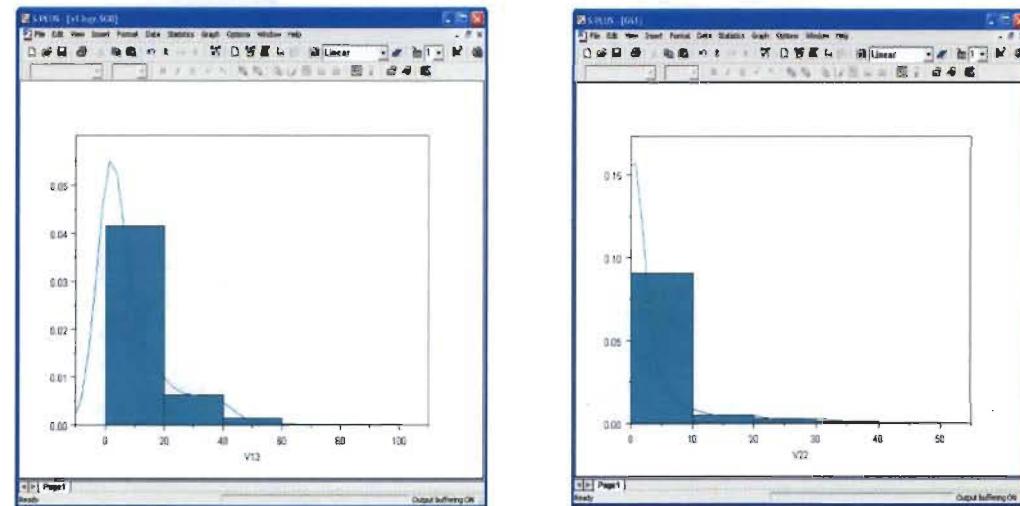
La première marginale de la couleur RGB



La deuxième marginale de la couleur RGB



La troisième marginale de la couleur RGB La quatrième composante de la couleur RGB

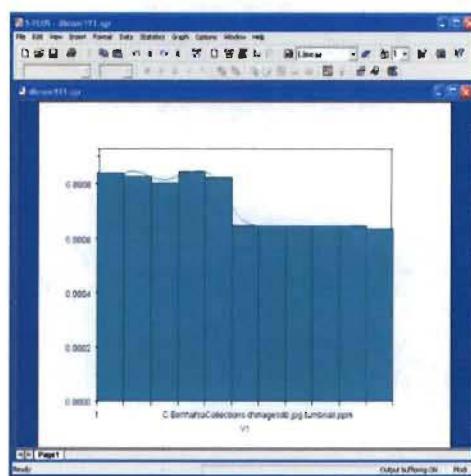


La cinquième marginale de la couleur RGB

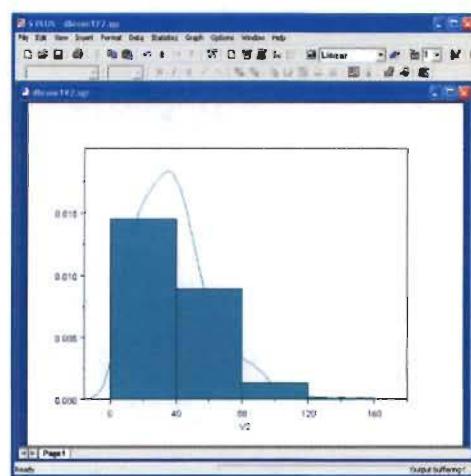
La sixième marginale de la couleur RGB

✓ Les composantes de la matrice de cooccurrence

Les caractéristiques de la texture et plus précisément la matrice de cooccurrence sont composées de 4 marginales (composante), on a présenté alors les 2 premières marginales ci-dessous.



La première marginale de la texture



La deuxième marginale de la texture

À partir de ces résultats que nous avons trouvés, nous avons constaté d'après les courbes de distribution de valeurs que les marginales ne suivent presque jamais des lois normales, mais suivent plutôt des lois de probabilité uniforme et des lois de probabilité gamma.

Au vu de ces résultats obtenus, nous avons alors décidé d'explorer la piste des modèles de copules archimédienne afin de trouver les lois de probabilités qui épousent mieux nos données ce qui devrait nous permettre de modéliser séparément les lois marginales et la dépendance. Dans ce qui suit, nous allons définir les copules avec ses différentes propriétés.

5.4 Copules et quelques propriétés

Ces dernières années il y a eu la naissance d'un nouvel outil de modélisation qui est en pleine expansion nommé « Les copules ». En effet, en faisant une recherche, il était vraiment difficile de trouver le mot « copule » dans la littérature mathématique et statistique. La toute première fois où on a trouvé le mot « copule » c'est était en 1981 dans le volume 7 du livre « *the paper is* » de Schweizer et Wolff (1981), mais avant cette date, le mot « copule » n'a jamais été mentionné. C'est à partir de 1993 que le nombre de références qui se rapporte sur les copules a commencé à augmenter.

Toutefois, cette théorie de copule est en pleine expansion vue qu'on lui porte un très grand intérêt et ça a été justifié par quatre conférences internationales consacrées aux copules et à leurs applications :

- Le "Symposium sur « les distributions aux marginaux Given (Classes Fréchet) » à Rome en 1990.
- La conférence sur « Distributions with Fixed Marginals, Doubly Stochastic Measures, and Markov Operators » à Seattle en 1993
- La conférence sur la « Distributions with Given Marginals and Moment Problems » à Prague en 1996.
- La conférence sur « les modèles de copules : théorie et application » à l'université de Laval en 2008.

En effet le mot « copule » remonte à un désormais célèbre théorème dû à Sklar (1959) il a été utilisé à des fins mathématiques et statistiques. Ce théorème consiste à séparer pour chaque distribution bidimensionnelle, la partie marge de distribution de la partie dépendance. Soit (X, Y) un vecteur aléatoire de fonction de répartition H :

$$H(x, y) = P(X \leq x, Y \leq y). \quad (3.52)$$

Bien évidemment, on peut extraire les lois de F et G par l'équation suivante :

$$F(x) = H(x, \infty) \quad \text{et} \quad G(y) = H(\infty, y) \quad (3.53)$$

Le théorème de Sklar permet d'écrire $H(x, y)$ en fonction des lois marginales et de la fonction de dépendance C , appelée copule. Ainsi on a :

$$H(x, y) = C\{F(x), G(y)\} \quad (3.54)$$

Dans le cas où les marges sont continues, la fonction C est unique. On peut alors extraire la copule d'une loi bivariée continue par la formule ci-après :

$$C(u, v) = H\{F^{-1}(u), G^{-1}(v)\} \quad (3.55)$$

Cette formule de Sklar met en évidence deux parties importantes qui sont :

- Les lois marginales de X et Y
- La structure de dépendance C entre les variables aléatoires X et Y

Les copules sont des fonctions qui joignent les fonctions de répartition conjointe à leur distribution marginale à une dimension. Aussi, elles sont des outils statistiques et leur principale utilité est de modéliser plusieurs variables aléatoires en même temps. En effet, les copules contiennent toutes les informations nécessaires au sujet de la structure de dépendance d'un vecteur de variables aléatoires. Elles sont utilisées dans divers domaines comme celui de la finance puisqu'elles permettent de représenter la mesure de dépendance statistique comparée à la mesure traditionnelle de corrélation. Aussi elles

permettent la construction de fonction de répartition multivariée indépendamment des lois marginales qui peuvent avoir chacune des lois de probabilités différentes.

5.4.1 Définition

5.4.1.1 Définition d'une copule

On appelle copule toute fonction $C : [0, 1]^2 \rightarrow [0, 1]$ qui satisfait les conditions suivantes:

➤ $C(u, 0) = C(0, u) = 0$ pour tout $u \in [0, 1]$

➤ $C(u, 1) = C(1, u) = u$ pour tout $u \in [0, 1]$

➤ Pour tout $u_1 \leq u_2$ et pour tout $v_1 \leq v_2$

$$C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0 \quad (3.56)$$

On peut aussi étendre la définition de copules à une fonction multivariée à n dimensions de $[0, 1]^n$ de sorte que chaque distribution marginale soit uniforme sur l'intervalle $[0, 1]$. Plus spécifiquement, $C : [0, 1]^n \rightarrow [0, 1]$ est une copule à n dimensions si :

➤ Pour tout $0 \leq u \leq 1$, $C(u) = 0$ si au moins un des u_i est égale à 0.

➤ Pour tout $0 \leq u \leq 1$, $C(u) = u_i$ si toutes les composantes sont égales à 1 sauf pour là $i^{\text{ème}}$ composante.

Il existe plusieurs familles de copules qui diffèrent chacune dans les détails de la dépendance qu'elle représente. Elles ont chacune des paramètres reliés à leur force et à leur forme de dépendance. Dans ce qui suit, nous allons présenter quelques-unes de ces familles de copules.

5.4.1.2 Quelques exemples de copules

5.4.1.2.1 Les bornes de Fréchet :

Elles permettent de déterminer un intervalle dans lequel toutes les lois bivariées sont logées. En effet, pour toute loi H on a :

$$\max \{F(x) + G(y) - 1, 0\} \leq H(x, y) \leq \min \{F(x), G(y)\}. \quad (3.57)$$

En sachant que ces bornes sont aussi des fonctions de répartition, c'est-à-dire qu'elles ne peuvent pas être améliorées. En appliquant l'équation (3.55) on obtient que tout copule C est tel que :

$$W(u, v) \leq C(u, v) \leq M(u, v) \quad (3.58)$$

Avec :

$W(u, v)$ est la copule minimum, appelée aussi la borne inférieure de Fréchet Hoeffding.

Dans le cas bivarié, elle est donnée par :

$$W(u, v) = \max (0, u + v - 1) \quad (3.59)$$

Dans le cas de n variables, elle est donnée par :

$$W(u_1, \dots, u_n) = \max \{1 - n + (\sum_1^n u_i), 0\} \quad (3.60)$$

$W(u, v)$: est la copule maximum, appelée aussi la borne supérieure de Fréchet Hoeffding. Dans le cas bivarié, elle est donnée par :

$$M(u, v) = \min (u, v) \quad (3.61)$$

Dans le cas de n variables, elle est donnée par :

$$\min(u_1, \dots, u_n) = M(u_1, \dots, u_n). \quad (3.62)$$

5.4.1.2.2 Copule d'indépendance

L'une des copules usuelles est la copule indépendante. Cette dernière est caractérisée par la factorisation des marges d'une loi H c'est-à-dire que X et Y sont indépendantes si et seulement si leur loi conjointe s'écrit de la manière suivante :

$$H(x, y) = F(x)G(y) \quad (3.63)$$

En appliquant l'équation numéro (3.55) on peut monter que la copule associée est :

$$\Pi(u, v) = uv \quad (3.64)$$

Cette copule caractérise donc l'indépendance bivariée. Notons qu'il existe plusieurs familles de modèles de copules comme les copules gaussiennes, les copules de Student, les copules archimédiennes, les copules de valeurs extrêmes, les copules Archimax, etc. Dans ce travail nous allons nous focaliser sur la famille des copules archimédiennes. Ci-après nous allons introduire la notion de copules archimédiennes.

5.4.2 Copules Archimédiennes

5.4.2.1 Définition

Cette théorie de copules archimédiennes a été introduite et développée par Christian Genest (1986). Elle permet de représenter un bon nombre de lois bidimensionnelles connues. Les copules archimédiennes sont définies de la manière suivante :

$$C(u_1, u_2) = \begin{cases} \varphi^{-1}(\varphi(u_1) + \varphi(u_2)) & \text{si } \varphi(u_1) + \varphi(u_2) \leq \varphi(0) \\ 0 & \text{sinon} \end{cases} \quad (3.65)$$

Où la fonction $\varphi : [0, 1] \rightarrow R^+$ est appelée une fonction génératrice, est convexe, est décroissante telle que $\varphi(1) = 0$, $\varphi'(u) < 0$ et $\varphi''(u) > 0$ pour tout $0 \leq u \leq 1$.

Nous allons énumérer quelques propriétés de ces copules :

- i. La distribution est symétrique ce qui veut dire que $C_\varphi(u, v) = C_\varphi(v, u)$ pour tout $(u, v) \in (0, 1)^2$.
- ii. Si $r > 0$ est une constante alors $\varphi^*(t) = r\varphi(t)$ engendre la même copule que φ ce qui veut dire que $C_\varphi(u, v) = C_{\varphi^*}(u, v)$ pour tout $(u, v) \in (0, 1)^2$.
- iii. C_φ est la copule d'indépendance si et seulement si $\varphi(t) = \alpha - \log(t)$.

Dans ce qui suit, nous allons donner quelques exemples de copules archimédiennes.

5.4.2.2 Exemples de copules Archimédiennes

- La copule de Clayton :

La copule Clayton est une copule archimédienne asymétrique, présentant une plus grande dépendance à l'égard de la queue négative que de la positive. Cette copule est donnée par:

$$C(u_1, u_2, \theta) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta} \quad (3.66)$$

avec $\theta \geq 0$ dont le générateur est :

$$\varphi_\theta(t) = \frac{t^{-\theta} - 1}{\theta} \quad (3.67)$$

- La copule de Frank

La copule de Frank est une copule archimédienne symétrique donnée par:

$$C_\alpha(u, v) = -\frac{1}{\alpha} \ln \left\{ 1 - \frac{(e^{-\alpha u} - 1)(e^{-\alpha v} - 1)}{e^{-\alpha} - 1} \right\} \quad (3.68)$$

dont le générateur est :

$$\varphi_\alpha(t) = -\ln \left(\frac{e^{-\alpha t} - 1}{e^{-\alpha} - 1} \right), \text{ avec } \alpha \neq 0 \quad (3.69)$$

- La copule de Gumbel

La copule de Gumbel est un autre modèle populaire des copules archimédiennes, dont le générateur est :

$$\varphi_\alpha(t) = (-\ln t)^\alpha, \text{ avec } \alpha \geq 1 \quad (3.70)$$

L'expression de la copule engendrée est donc :

$$C_\alpha(u, v) = \exp \left\{ [(-\ln u^\alpha) + (-\ln v^\alpha)]^{1/\alpha} \right\} \quad (3.71)$$

5.4.3 Les tests d'adéquation

5.4.3.1 Description de l'algorithme de simulation d'une copule

Pour simuler un couple aléatoire (u, v) dont la loi jointe est une copule C , la méthode basée sur la fonction de répartition conditionnelle est souvent employée. À cette fin, définissons d'abord

$$F_v(u) = P(U \leq u \mid V = v) = C_{01}(u, v) = \frac{\partial C(u, v)}{\partial v} \quad (3.72)$$

Ainsi, si U et V sont des variables aléatoires indépendantes de loi uniforme sur $[0, 1]$, alors la loi jointe de $U = F_V^{-1}(t)$ et V est C . D'où l'algorithme

Algorithme :

- 1) Générer de façons indépendantes deux valeurs aléatoires u et v de loi uniforme sur $[0, 1]$
- 2) Calculer $F_v^{-1}(t)$
- 3) Stocker les couples $(F_v^{-1}(t), v)$

5.4.3.2 Description de l'algorithme de simulation d'une copule Archimédienne

Soit $C(u, v)$ une copule archimédienne de générateur φ , c'est-à-dire $C(u, v)$ s'écrit sous forme :

$$C(u, v) = \varphi^{-1}(\varphi(u) + \varphi(v)) \quad (3.73)$$

On a

$$F_v(u) = P(U \leq u \mid V = v) = C_{01}(u, v) = \frac{\partial C(u, v)}{\partial v} = \frac{\varphi'(v)}{\varphi' \varphi^{-1}(\varphi(u) + \varphi(v))} \quad (3.74)$$

On a aussi

$$F_v^{-1}(t) = \varphi^{-1} \left(\varphi \left((\varphi')^{-1} \left(\frac{\varphi'_v}{t} \right) \right) - \varphi(v) \right) \quad (3.75)$$

D'où l'algorithme.

Algorithme :

- 1) Générer de façons indépendantes deux valeurs aléatoires t et v de loi uniforme sur $[0, 1]$.
- 2) Stoker les couples $\left(\varphi^{-1} \left(\varphi \left((\varphi')^{-1} \left(\frac{\varphi'_v}{t} \right) \right) - \varphi(v) \right), v \right)$.

Nous venons de voir durant les points précédant de ce chapitre la méthode de mixture de Gaussienne ainsi que ses inconvénients et la définition d'une copule et quelques unes de ses propriétés, cela va nous permettre d'aborder le point suivant qui concerne le travail que nous allons réaliser.

5.5 Notre proposition

Notre objectif dans ce mémoire est d'indexer une base de données d'images en utilisant la méthode des copules archimédiennes. Pour cela, nous allons essayer de modéliser adéquatement nos vecteurs aléatoires (nos caractéristiques d'images). Afin d'y parvenir,

nous proposons une approche en deux étapes, qui consiste à modéliser séparément les lois marginales et la dépendance sous-jacente. Plus spécifiquement :

- Lors de la première étape, nous allons essayer d'identifier les lois marginales des vecteurs aléatoires impliqués, en utilisant des tests d'adéquation univariés.
- Lors de la deuxième étape, nous essayons de modéliser la dépendance entre les composantes de nos vecteurs aléatoires en utilisant les modèles de copules archimédiennes. Des tests d'adéquation sur ces dernières seront exécutés afin d'identifier la copule archimédienne qui ajuste la dépendance entre les composantes des vecteurs aléatoires d'intérêt.

Finalement, les densités multivariées retenues par cette démarche seront utilisées pour effectuer une indexation automatique des collections d'images.

Dans ce qui suit, nous allons détailler chacune des deux étapes de notre travail.

5.5.1 Première étape

Comme nous l'avons mentionné ci-haut, la première étape de notre travail va tenter d'identifier la loi marginale de chaque composante de chaque caractéristique, et ce en utilisant des tests d'adéquation. Pour ce faire, nous allons procéder comme suit :

1. Dans un premier temps, nous allons extraire les caractéristiques des images de notre base de données.
2. Dans un second temps, nous appliquerons l'algorithme K-MEANS pour faire une classification initiale de ces images.

3. Après cela, nous appliquerons le test d'adéquation de Kolmogorov-Smirnov pour déterminer la loi de chaque marginale de chaque classe.

4. Finalement, nous calculerons les fonctions de répartition et les fonctions de densité pour chaque loi trouvée. Voici maintenant les détails par rapport à chacune de ces sous-étapes.

5.5.1.1 Extraction des caractéristiques de ces images :

Pour la réalisation de notre système de classification d'images par le contenu, nous avons besoin d'extraire les caractéristiques des images. Pour cela, nous allons extraire celles de la couleur (RGB et HSV) et celles de la texture (la matrice de cooccurrence). Nous allons les détailler dans le chapitre suivant à l'aide d'un système déjà existant comme le montre la figure 18.

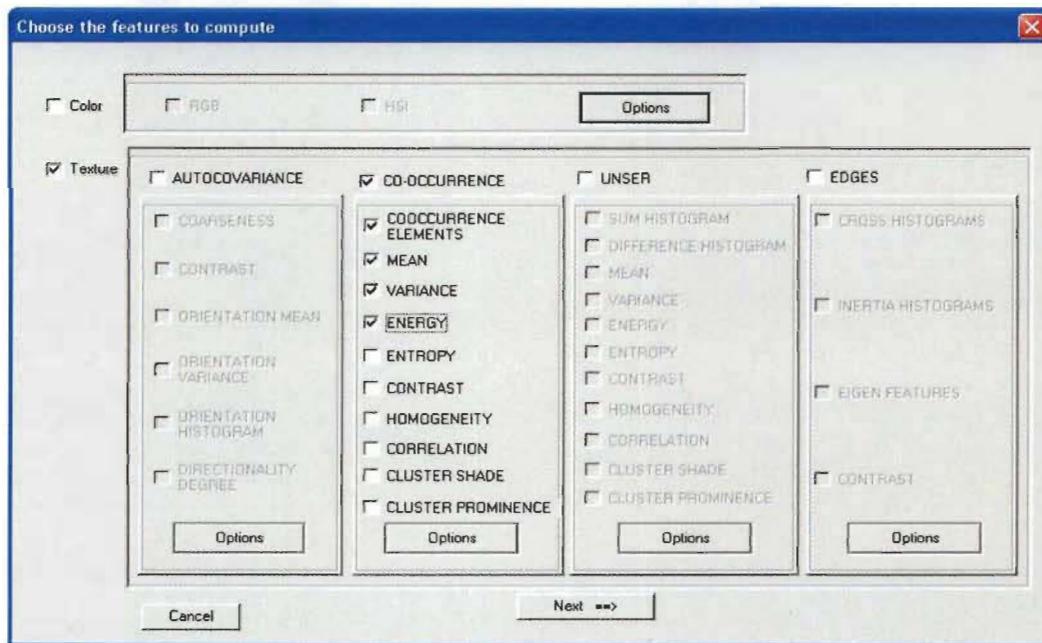


Figure 18 : Exemple d'extraction des caractéristiques (dans notre cas c'est la texture)

L'interface nous donne le choix entre deux caractéristiques. Dans notre cas nous avons choisi la caractéristique de la texture (pour la caractéristique de la couleur, il faut cocher la case couleur ensuite choisir l'une des deux signatures). Après cela, nous avons choisi la matrice de cooccurrence comme caractéristique à extraire avec les quatre indices qui correspondent à des caractères descriptifs de la texture en l'occurrence la moyenne, la variance, l'énergie et l'entropie.

5.5.1.2 Application du K-MEANS pour faire une classification initiale :

Une fois l'extraction des caractéristiques faite, nous allons appliquer l'algorithme K-MEANS pour faire une première classification de notre collection d'images.

Rappel sur l'algorithme K-MEANS :

L'algorithme K-MEANS est une technique dont le but est de partitionner les données en N groupes (classes) où chaque donnée appartient à la partition avec la moyenne la plus proche. Ce résultat est obtenu en positionnant N centroïdes dans les zones où il y a le plus de données. Chaque donnée est alors affectée au centroïde le plus proche en appliquant la règle de la distance minimale. Chaque groupe contient donc les données qui sont plus proches d'un certain centroïde que tout autre centroïde. Les données sont positionnées par une procédure itérative qui va les amener graduellement dans leur position finale comme le montre la figure 19.

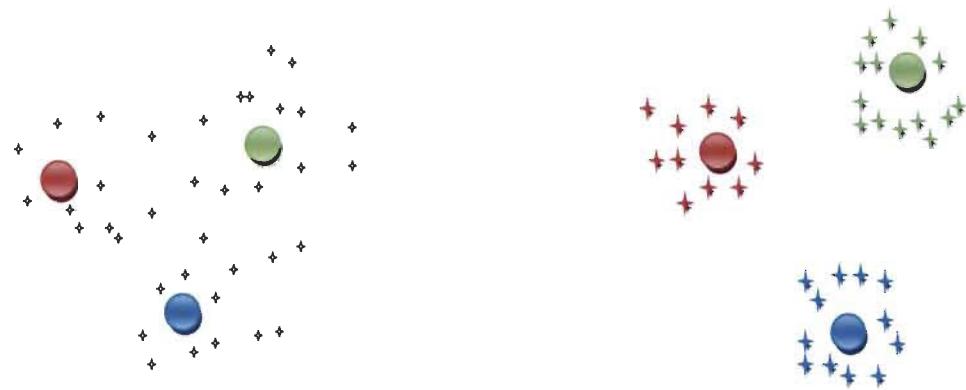


Figure 19 : Schéma d'une classification en utilisant l'algorithme K-MEANS

Voici par ailleurs une illustration de notre base de données classifiée en utilisant l'algorithme K-MEANS comme le montre la figure 20:

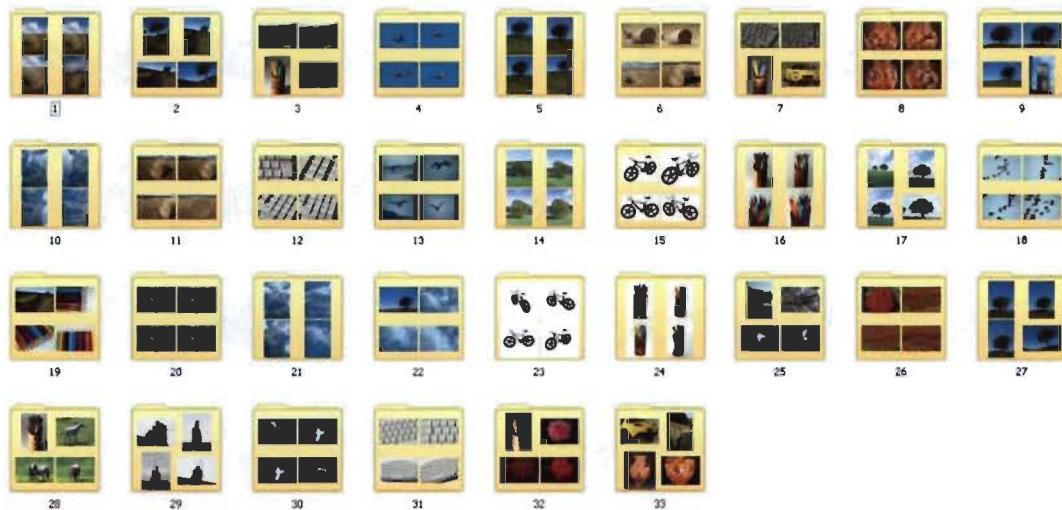


Figure 20 : Indexation de notre base de données en utilisant l'algorithme K-MEANS

5.5.1.3 Test d'adéquation de Kolmogorov-Smirnov :

Une fois la classification faite, on va déterminer pour chaque marginale de chaque classe les lois marginales qui épousent mieux nos données vu que dans la section 3.3 de ce chapitre, nous avons montré que les marginales ne suivent pas des lois normales, mais elles suivent d'autres lois. Pour déterminer ces lois, nous avons procédé à des tests d'adéquation en appliquant le test de Kolmogorov-Smirnov pour chaque marginale de chaque classe après application de l'algorithme K-MEANS.

Rappel sur le test de Kolmogorov-Smirnov :

Le test de Kolmogorov-Smirnov est un test d'ajustement qui permet d'identifier si un échantillon donné suit une loi donnée connue par sa fonction de répartition continue où bien si deux échantillons suivent une même loi. En effet, ce test permet de comparer la fonction de répartition continue avec la fonction de répartition de l'échantillon empirique. Plus spécifiquement, est de calculer la distance maximale entre les fonctions théoriques et empiriques, mais si la distance dépasse un seuil, la loi testée pour l'échantillon n'est pas bonne. Nous avons donc implémenté l'algorithme de Kolmogorov-Smirnov en utilisant cinq lois différentes en l'occurrence :

- 1 : La loi normale
- 2 : La loi exponentielle
- 3 : La loi de la valeur extrême généralisée
- 4 : La loi gamma
- 5 : La loi uniforme

5.5.1.4 Détermination des lois marginales et calcul des fonctions de répartition et les fonctions de densité :

Après que nous ayons trouvé les lois, nous allons calculer pour chacune d'elle et pour chaque classe les fonctions de répartitions et les fonctions de densité.

Les figures 21 présentent quelques résultats que nous avons obtenus après la classification faite par K-MEANS (détermination des lois marginales pour chaque marginale de chaque classe) ainsi que le calcul des fonctions de répartition et des fonctions de densité.

Figure 21 : Détermination des lois marginales.

Nous remarquons dans les résultats obtenus à l'aide du test de Kolmogorov-Smirnov dont nous avons parlé précédemment que les lois marginales de la première classe suivent trois lois : normale (1), gamma (4), uniforme (5) et exponentiel (2) ce qui nous amène à la deuxième approche.

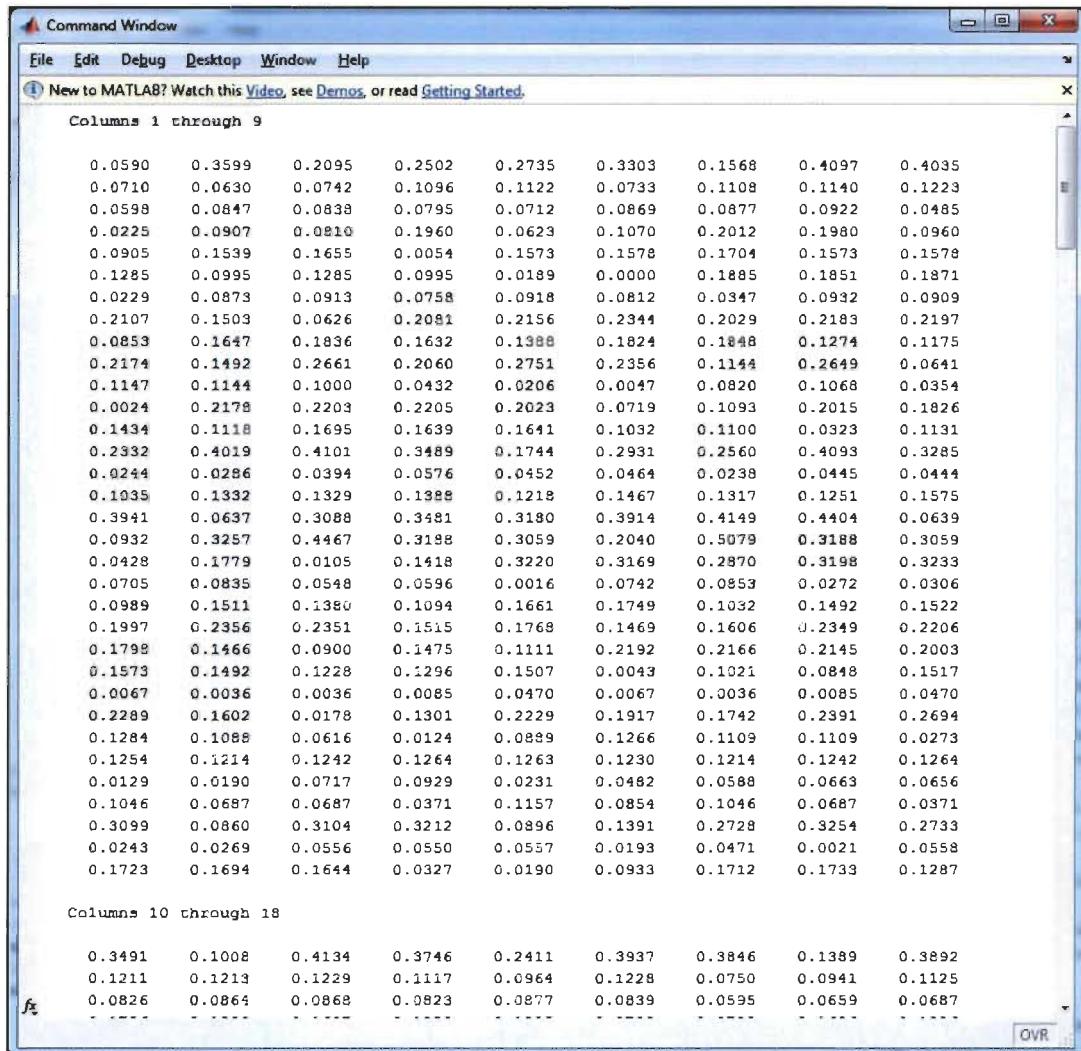


Figure 22 : Calcul des fonctions de répartition et des fonctions de densité

5.5.2 Deuxième étape

Une fois la première étape réalisée, nous allons maintenant attaquer la deuxième étape que nous avons mentionnée ci-haut. Cette deuxième partie va nous permettre d'identifier la copule archimédienne qui ajuste la dépendance entre les composantes des vecteurs aléatoires d'intérêt via des tests d'adéquation. Nous allons alors effectuer la démarche suivante.

1. Dans un premier lieu, nous allons modéliser la dépendance entre les marginales de nos vecteurs aléatoires deux à deux en utilisant les modèles de copules et en appliquant le test d'adéquation de Ro-Speerman pour déterminer la copule qui épouse le mieux nos données.
2. Dans un second lieu, nous allons calculer les probabilités d'appartenance de chaque image par apport à sa classe d'appartenance et aussi par apport aux autres classes. Une fois ces probabilités calculées, nous allons affecter chaque image à la classe concernée de telle sorte que sa probabilité d'appartenance soit maximale.

Voici maintenant les détails par rapport à chacune de ces sous-étapes.

5.5.2.1 Détermination de la copule

Maintenant que nous avons déterminé les lois pour chacune de nos marginales et calculé les fonctions de répartition et les fonctions de densité, nous allons modéliser la dépendance entre ces marginales deux à deux en utilisant les modèles de copules archimédiennes vue qu'il n'y a pas dans la littérature une procédure qui nous permet de tester l'adéquation d'une copule multivariée, c'est pour cela qu'on a procédé de cette manière. Nous allons alors appliquer des tests d'adéquations en utilisant la fonction de dépendance de Spearman.

Rappel sur la fonction de Dépendance de Spearman :

➤ Rho de Spearman :

Le Rho de Spearman est une mesure non paramétrique qui permet d'évaluer la force de la dépendance entre deux variables aléatoires. Au lieu de calculer la corrélation entre A et B , ce coefficient mesure la corrélation entre les variables uniformisées $X = F(A)$ et $Y = G(B)$. Ainsi la définition du Rho de Spearman est :

$$\rho_s = \text{cor} \{F(A), G(B)\} = \frac{E(XY) - E(X)E(Y)}{\{\text{var}(X)\}^{\frac{1}{2}}\{\text{var}(Y)\}^{\frac{1}{2}}} \quad (3.76)$$

Comme les variables $F(A)$ et $G(B)$ sont uniformes sur $(0,1)$, on calcule plus facilement $\text{var}(X) = \text{var}(Y) = 1/12$ ainsi, $\rho_s = 12(E(XY) - E(X)E(Y))$

➤ Définition de la fonction de dépendance de Spearman :

ρ_s est basé sur la moyenne de la variable aléatoire $\xi = XY$, où $X = F(A)$ et $Y = G(B)$, à savoir $E(\xi)$ selon la version théorique. Au lieu de baser une notion de dépendance sur la moyenne ξ , on pourrait se baser sur la dépendance par la fonction :

$$K(z) = P(XY \leq z). \quad (3.77)$$

Cette forme de dépendance généralise donc le Rho de Spearman vue que

$$E(XY) = \int_0^1 z dK(z) = \int_0^1 \{1 - K(z)\} dz. \quad (3.78)$$

Et puisque $\rho_s = 12E(XY) - 3$ alors,

$$\rho_s = 9 - 12 \int_0^1 K(z) dz. \quad (3.79)$$

Si C est la copule de (A, B) et que C est absolument continu

$$L(t) = t + \int_t^1 C_{01} \left(\frac{t}{y}, y \right) dy. \quad (3.80)$$

Où $C_{01} = \frac{\partial C}{\partial y}$ est la loi conditionnelle de X étant donnée $Y = y$.

En modélisant nos vecteurs, nous avons remarqué que la copule de FRANK (définis dans la section 3.4.2.2) est celle qui revient le plus souvent.

5.5.2.2 Calcul des probabilités

Après que nous ayons déterminé la copule la plus adaptée à nos données ainsi que le calcul des fonctions de répartition et les fonctions de densité, nous allons maintenant procéder au calcul des probabilités d'appartenance de chaque image à chaque classe ensuite, nous allons affecter chaque image à la classe concernée.

Plus spécifiquement, nous allons déterminer les lois de probabilité de chaque marginale ainsi que leur fonction de répartition F_i et leur fonction de densité f_i . Nous avons identifié la copule de FRANK (en estimant nos vecteurs deux à deux) ainsi que sa fonction de répartition définit par :

$$C(u_1, \dots, u_n) = -\frac{1}{\theta} \ln \left\{ 1 + \frac{1}{(e^{-\theta} - 1)^{n-1}} \prod_{i=1}^n (e^{-\theta u_i} - 1) \right\} \quad (3.81)$$

Et sa fonction de densité définie par :

$$c(u_1, \dots, u_n) = \frac{\partial^n C(u_1, \dots, u_n)}{\partial u_1 \dots \partial u_n} \quad (3.82)$$

Donc les probabilités à calculer sont :

$$P(x_1, \dots, x_n) = c(F_1(x_1), \dots, F_n(x_n)) \prod_{i=1}^n f_i(x_i) \quad (3.83)$$

Sauf que le terme $c(F_1(x_1), \dots, F_n(x_n))$ n'est pas explicite, on doit donc l'approcher.

Pour ce faire on a posé :

$$c(F_1(x_1), \dots, F_n(x_n)) = \frac{C[F_1(x_1), \dots, F_i(x_i) + h, \dots, F_n(x_n)] - C[F_1(x_1), \dots, F_n(x_n)]}{h^n}$$

Le terme $c(F_1(x_1), \dots, F_n(x_n))$ sera approché par

$$c(F_1(x_1), \dots, F_n(x_n)) = \frac{\Delta_{n,h} \circ \dots \circ \Delta_{1,h} C[F_1(x_1), \dots, F_n(x_n)]}{h^n} \quad (3.84)$$

D'où la probabilité d'affectation sera :

$$P(x_1, \dots, x_n) \approx \frac{\Delta_{n,h} \circ \dots \circ \Delta_{1,h} C[F_1(x_1), \dots, F_n(x_n)]}{h^n} \prod_{i=1}^n f_i(x_i) \quad (3.85)$$

Pour simplifier, nous posons $u_1 = F_1(x_1), \dots, u_n = F_n(x_n)$. Notre objectif est d'évaluer

$$\frac{\Delta_{n,h} \circ \dots \circ \Delta_{1,h} C[u_1, \dots, u_n]}{h^n}$$

Voici une expression alternative de $\Delta_{n,h} \circ \dots \circ \Delta_{1,h} C[u_1, \dots, u_n]$

$$\sum_{i=1}^n (-1)^{n-i} s_i \quad (3.86)$$

où encore

$$(-1)^n s_0 + (-1)^{n-1} s_1 + (-1)^{n-2} s_2 + (-1)^{n-2} s_3 + s_n$$

avec

$$s_0 = C[u_1, \dots, u_n]$$

$$s_1 = \sum_{i \in \{1, \dots, n\}} C[u_1, \dots, u_i + h, \dots, u_n] : \text{il y a } \binom{n}{1} = n \text{ termes}$$

$$s_2 = \sum_{i < j \in \{1, \dots, n\}} C[u_1, \dots, u_i + h, \dots, u_j + h, \dots, u_n] : \text{il y a } \binom{n}{2} \text{ termes}$$

$$s_3 = \sum_{i < j < k \in \{1, \dots, n\}} C[u_1, \dots, u_i + h, \dots, u_j + h, \dots, u_k + h, \dots, u_n] : \text{il y a } \binom{n}{3} \text{ termes}$$

ainsi de suite

$$s_n = C[u_1 + h, \dots, u_n + h] \quad (3.87)$$

Description de l'algorithme :

L'algorithme que nous avons conçu pour classifier une collection d'images en utilisant la méthode des copules archimédienne est comme suit :

- Pour classifier une image j , nous allons déterminer les marges F_1^j, \dots, F_n^j et la copule de FRANK C_j de paramètre estimé Θ_j .
- Pour une image donnée $x = (x_1, \dots, x_n)$, nous calculons $F_1^j(x_1), \dots, F_n^j(x_n)$ et $f_1^j(x_1), \dots, f_n^j(x_n)$ associé à la classe j .
- Nous calculons par la suite les probabilités d'affectation de l'image i à la classe j , $P(i|j)$ par :

$$P(i|j) \approx \frac{\Delta_{n,h} \circ \dots \circ \Delta_{1,h} C[F_1(x_1), \dots, F_n(x_n)]}{h^n} \prod_{i=1}^n f_i(x_i) \quad (3.88)$$

- Nous affectons chaque image i à la classe k tel que :

$$P(i|k) = \max P(i|j) \quad (3.89)$$

Nous venons de voir ci-dessus la méthode que nous avons proposée pour classifier une base de données d'images et qui est basée sur le modèle de copule archimédienne.

5.6 Conclusion

Durant ce chapitre, nous avons en premier lieu expliqué la classification à l'aide de la GMM. En second lieu, nous avons critiqué cette méthode et en troisième lieu, nous avons expliqué notre proposition. En effet, nous avons développé un système d'indexation d'images qui utilise les caractéristiques visuelles des images. Pour ce faire, nous avons extrait ses caractéristiques qui représentent bien le contenu des images de notre banque de données. Ensuite nous avons utilisé ses caractéristiques pour une première classification à l'aide de l'algorithme K-MEANS. Une fois la classification faite, nous avons dans un premier temps déterminé les lois marginales et dans un deuxième temps, nous avons calculé les fonctions de répartition et les fonctions de densité de chacune des lois. Puis, nous avons déterminé la copule archimédienne qui est la plus adaptée à nos données en appliquant des tests d'adéquation. Enfin, nous avons calculé les probabilités d'affection de chaque image pour les classifier. Mais pour la réalisation de ce système dont nous allons comparer ses résultats avec d'autres systèmes qui ont utilisé d'autres méthodes, nous avons besoin des caractéristiques visuelles des images, cela nous amène au chapitre suivant dans lequel nous allons détailler les caractéristiques d'images que nous avons utilisées pour réaliser notre travail ainsi que l'évaluation de notre système par rapport aux systèmes déjà existants.

4- Chapitre 4

6 EXPÉRIMENTATION

7.1 Introduction

Dans le chapitre précédent, nous avons proposé une nouvelle méthode pour la classification des BD d'images. Notre méthode est basée sur le modèle de copules archimédiennes. Dans le présent chapitre, nous allons valider la méthode proposée, et ce en effectuant un certain nombre d'expériences.

Rappelons que notre méthode utilise les caractéristiques des images afin de classifier ces dernières. Ceci nous amène donc à expliquer, dans la section 4.2 les caractéristiques utilisées et comment elles ont été extraites. Dans la section 4.3 nous allons décrire les mesures avec lesquelles on va faire notre évaluation. Dans la section 4.4 nous allons faire une brève présentation des méthodes en l'occurrence la méthode du K-MEANS, la méthode GMM avec lesquelles on va comparer notre système d'indexation. Dans la section 4.5 nous effectuerons nos expérimentations, enfin, dans la section 4.6 nous allons finir notre chapitre par une conclusion.

7.2 Les caractéristiques utilisées

Dans le chapitre 1, nous avons parlé de façon générale des différentes caractéristiques utilisées dans la recherche et l'indexation d'images par le contenu. Maintenant, nous allons donner plus de détails sur les caractéristiques que nous avons utilisées dans notre travail. Ces caractéristiques sont en l'occurrence la caractéristique de la couleur dont

l'histogramme de couleur et les espaces de couleur RGB et HSV, mais aussi la caractéristique de la texture et plus précisément la matrice de cooccurrence.

7.2.1 La couleur

La couleur est une propriété de la lumière qui permet à l'être humain en utilisant son système de vision de distinguer les différentes répartitions spectrales de la lumière. En effet, la lumière est une onde électromagnétique, elle peut avoir des fréquences différentes. Par exemple quand les longueurs d'onde sont courtes vers 380 nm, elles sont perçues comme du bleu-violet par un œil humain. Par contre si les longueurs d'onde sont plus longues vers 700 nm, elles sont perçues comme rouge. La figure 23 montre le spectre des ondes électromagnétiques.

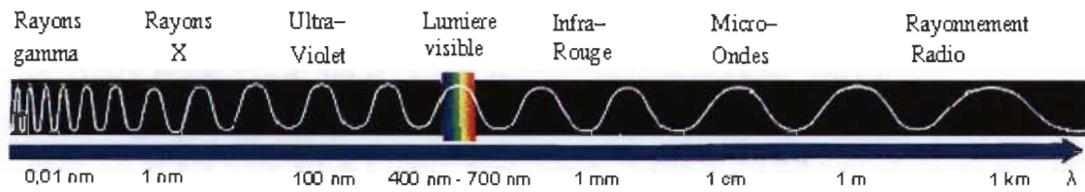


Figure 23 : Spectre des ondes électromagnétiques

L'ensemble des ondes visibles par le système de vision de l'être humain s'appelle le spectre de la lumière visible. La figure 24 montre le spectre visible pour l'œil humain.

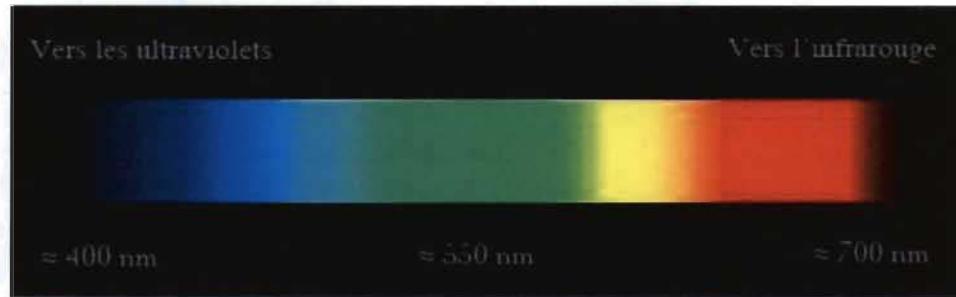


Figure 24 : Spectre visible

Actuellement nous remarquons avec les avancées technologiques que toutes les banques de données d'images sont en couleur puisqu'elles sont plus représentatives et convenables pour l'être humain. L'obligation de créer, stocker et envoyer des données en couleur (images) a engendré l'émergence d'un nouveau système qui permet de représenter correctement ces couleurs. Ce nouveau système est l'espace des couleurs. Il existe plusieurs espaces de couleurs qui ont été utilisés pour la représentation de la couleur basée sur les concepts perceptuels comme l'espace de couleur RGB, HSV, CMY, etc. Dans ce qui suit, nous allons présenter avec détail les deux espaces de couleurs RGB et HSV.

7.2.1.1 *L'espace de couleur RGB*

L'espace de couleur RGB (Red, Green, Blue) appelé aussi RVB (Rouge, Vert, Bleu) reste de loin le plus utilisé et le plus simple à manipuler. C'est un espace de couleur additif c'est-à-dire qu'une couleur est représentée par le mélange des trois couleurs de base de diverses façons qui composent cet espace. Il est basé sur un système de coordonnées cartésiennes et sa représentation numérique a des valeurs allant de 0 à 255. Ainsi, chaque composante RGB possède une intensité de luminosité allant de 0 (noir) à 255 (blanc). Si les trois couleurs ont une intensité égale à 0 la couleur obtenue sera le

noir et si les trois composantes ont une intensité égale à 255 alors la couleur obtenu est le blanc. On représente généralement cet espace de couleur RGB par un cube appelé « cube de Maxwell » comme illustré dans la figure 25 dont chacun des axes correspond à une couleur primaire :

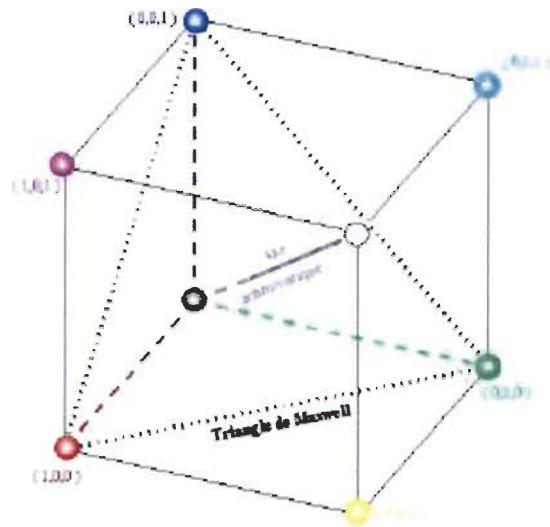


Figure 25 : L'espace de couleur RGB

L'avantage de l'espace de couleur RGB est qu'il est simple d'utilisation. En effet, il a été utilisé dans plusieurs appareils électriques et électroniques comme la télévision, les caméras vidéo, le numérisateur d'images, les appareils photo numériques et les ordinateurs. Par contre, l'inconvénient est qu'il n'est pas indépendant du matériel utilisé, et qu'il n'est pas très intuitif pour les utilisateurs non initiés.

7.2.1.2 L'espace de couleur HSV

L'espace de couleur HSV (Hue Saturation Value) utilisé aussi sous le nom TSV (Teinte Saturation Valeur) est un espace qui a été dérivé de celui de RGB (Red Green Blue) vu au point précédent. Cet espace est particulièrement utilisé dans des applications informatiques de graphisme. En 1978, le chercheur Alvy Ray Smith [55] a clairement décrit cet espace. Cet espace est plus intuitif à utiliser car il correspond à la façon dont nous percevons les couleurs. En effet, les couleurs, plutôt que d'être décomposées en primitives, comme dans RGB, sont représentées selon des notions évidentes: la luminosité, la teinte et la saturation.

La teinte -» Hue :

La teinte décrit la couleur en elle-même. Sa valeur varie de 0 à 360 puisqu'elle est codée suivant l'angle qui lui correspond sur le cercle des couleurs où se trouvent six couleurs primitives comme l'illustre la figure 26 : rouge, jaune, vert, cyan, bleu et magenta et les différentes couleurs intermédiaires.



Figure 26 : Le cercle des couleurs.

La saturation -» Saturation :

La saturation décrit la pureté de la couleur. Elle correspond à la distance au blanc. Au centre du cône, les couleurs sont des tons de gris; plus on s'éloigne du centre, plus les couleurs se démarquent et plus elles sont pures. Les couleurs sont saturées, ou maximalement pures, sur la surface du cône. Sa valeur varie entre 0 et 100 %.

La luminosité -» Value :

La luminosité correspond à la brillance perçue de la couleur. Sa valeur varie entre 0 et 100%. Plus la valeur d'une couleur est faible, plus la couleur est sombre. Une valeur de 0 correspond au noir. L'espace HSV est connu aussi sous le nom de cône hexagonal. Deux modèles peuvent être distingués : le modèle de cône hexagonal simple et double. Ils représentent la couleur sous forme d'un triplet : teinte H (Hue), Saturation S et luminosité V (Value). La figure 27 illustre l'espace HSV.

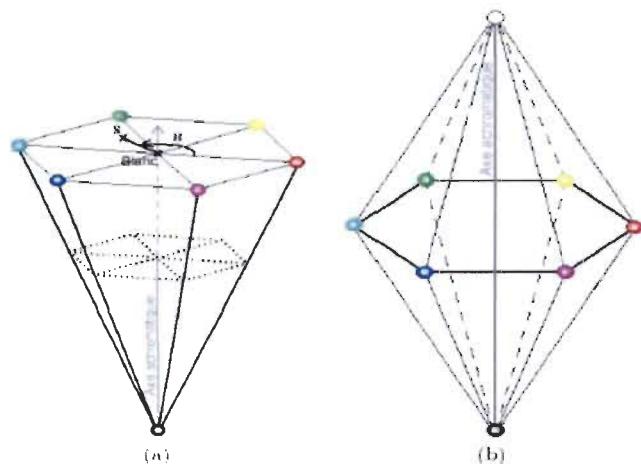


Figure 27 : L'espace de couleur HSV.

Comme on a vu précédemment, l'espace HSV est déduit à partir de RGB d'où les valeurs des composantes HSV sont obtenues à partir d'une transformation non linéaire des couleurs RGB, selon des notions mathématiques qui sont :

$$t = \begin{cases} 60^\circ * \left(\frac{v-b}{\max - \min} + 360^\circ \right) \bmod 360^\circ & \text{si } \max = r \\ 60^\circ * \left(\frac{b-r}{\max - \min} + 120^\circ \right) & \text{si } \max = v \\ 60^\circ * \left(\frac{r-v}{\max - \min} + 240^\circ \right) & \text{si } \max = b \end{cases} \quad (4.1)$$

$$s = \begin{cases} 0 & \text{si } \max = 0 \\ 1 - \frac{\min}{\max} & \text{sinon} \end{cases} \quad (4.2)$$

Une étude faite récemment [56] compare six espaces colorimétriques et montre que l'espace HSV est le plus efficace pour la recherche d'images par le contenu, cependant cet espace n'est pas perceptuellement uniforme.

7.2.2 La texture

On trouve des formes texturées un peu partout surtout dans la nature, mais il n'existe pas de définition pertinente de la texture. Cependant elle a une définition de sens commun. La texture joue un rôle très important dans l'identification et l'extraction des informations contenues dans l'image. En effet, la texture est la répétition d'éléments de base construite à partir de pixels qui respectent un certain ordre telle que l'herbe, l'eau, le sable... mais l'aléatoire joue un rôle particulier dans la texture. On peut alors remarquer deux types de texture : la première qui est la texture régulière dans laquelle la périodicité du motif est évidente comme les murs et les grilles, la deuxième est la texture aléatoire pour laquelle la distribution des intensités n'est l'objet d'aucune régularité

apparente comme les nuages et les foules. La texture se révèle donc par une information visuelle qui permet de la décrire comme lisse, grossière, fine, tachetée, granuleuse. La figure 28 présente quelques types de textures existantes.



Figure 28 : Quelques exemples de texture

Haralick [57] a donné une autre définition plus élargie de la texture en la présentant comme un phénomène à deux dimensions. La première, concerne la description d'éléments primitifs ou de base, par exemple les motifs. La deuxième est liée à la représentation spatiale de l'organisation spatiale de ces primitives. Pour analyser la texture, il existe plusieurs méthodes qui ont été présentées dans la littérature. Parmi ses méthodes, il y a la méthode statistique : elle permet d'étudier les relations entre les pixels et leurs voisins. La méthode la plus citée est la méthode de la matrice de cooccurrence [58]. Dans certains travaux [59] [60] [61], la méthode de la matrice de cooccurrence transmet d'assez bons résultats. Dans ce qui suit, nous allons détailler cette méthode.

7.2.2.1 La méthode de matrice de cooccurrence

La méthode de matrice de cooccurrence est énormément utilisée dans le domaine du traitement d'images [62]. Elle a été suggérée par Haralick et al. [62]. Elle permet de déterminer la fréquence d'apparitions d'un motif formé par deux pixels séparés par une distance déterminée d (d est la distance entre les 2 pixels) dans une direction Θ (Θ est l'angle de la droite reliant ces 2 pixels par rapport à l'horizontale). Plus exactement, dans une matrice de cooccurrence $P(a,b)_{d,\Theta}$ représente le nombre de points de niveau de gris a ayant comme voisin un autre point de niveau de gris b .

$$d = (da, db) = (d \cos \Theta, d \sin \Theta) \quad (4.3)$$

Une matrice de cooccurrence est une matrice de taille $N \times N$ où N est le nombre de niveaux de gris d'une image. Nous prenons alors un exemple d'une image à 6 niveau de gris. Dans ce qui suit la représentation de l'exemple avec ses quatre différentes directions (les valeurs généralement utilisées pour Θ sont : $0^\circ, 45^\circ, 90^\circ, 135^\circ$).

3	5	5
1	3	2
6	5	1
5	3	6

0°	1	2	3	4	5	6
1	0	0	1	0	0	0
2	0	0	0	0	0	0
3	0	1	0	0	1	1
4	0	0	0	0	0	0
5	1	0	1	0	1	0
6	0	0	0	0	1	0

a) Une image à 6 niveau de gris

b) Première direction $\Theta = 0^\circ$

45°	1	2	3	4	5	6
1	0	0	0	0	1	0
2	0	0	0	0	0	0
3	1	0	1	0	0	0
4	0	0	0	0	0	0
5	0	1	0	0	0	1
6	0	0	1	0	0	0

c) Deuxième direction : $\Theta = 45^\circ$

90°	1	2	3	4	5	6
1	0	0	0	0	0	2
2	1	0	0	0	0	0
3	1	0	0	0	1	0
4	0	0	0	0	0	0
5	0	1	2	0	0	0
6	0	0	0	0	1	0

d) Troisième direction : $\Theta = 90^\circ$

135°	1	2	3	4	5	6
1	0	0	1	0	0	0
2	0	0	0	0	1	0
3	0	0	0	0	0	1
4	0	0	0	0	0	0
5	1	0	1	0	1	0
6	0	0	0	0	0	0

e) Quatrième direction : $\Theta = 135^\circ$

Figure 29 : exemple de la méthode de matrice de cooccurrence

La matrice de cooccurrences contient d'énormes quantités d'information très importantes et sont donc difficilement manipulables, c'est pour cette raison qu'on ne l'utilise pas directement, mais à travers des mesures dites indices de texture. De ce fait Haralick et al. [62] ont suggéré quatorze indices qui correspondent à des caractères descriptifs des textures et qui peuvent être calculés à partir de ces matrices. Malgré qu'ils soient reliés entre eux, l'objectif de ces indices est de diminuer l'information contenue dans la matrice de cooccurrence et de permettre une meilleure distinction des

différents types de textures. D'après [63] les caractéristiques de texture les plus souvent utilisées sont :

➤ La moyenne :

$$MEAN = \sum_i \sum_j i \cdot C(i, j) \quad (4.4)$$

Où $C(i, j)$ correspond aux éléments de la matrice de cooccurrences.

➤ La variance :

$$VARIANCE = \sum_i \sum_j (i - \mu)^2 \cdot C(i, j) \quad (4.5)$$

Où μ est la moyenne calculée ci-dessus, et $C(i, j)$ correspond aux éléments de la matrice de cooccurrences. Cette mesure caractérise la distribution des niveaux de gris autour de la valeur moyenne « Mean ».

➤ L'énergie :

$$ENERGY = \sum_i \sum_j C(i, j)^2 \quad (4.6)$$

Où $C(i, j)$ correspond aux éléments de la matrice de cooccurrences. L'énergie caractérise l'homogénéité de l'image. S'il y a beaucoup de transitions de niveaux de gris dans l'image si la valeur de l'énergie est grande.

➤ L'entropie :

$$ENTROPY = \sum_i \sum_j -C(i, j) \cdot \log(C(i, j)) \quad (4.7)$$

Où $C(i, j)$ correspond aux éléments de la matrice de cooccurrences. L'entropie permet de mesurer la complexité de l'image, mais aussi de caractériser le degré de grandeur des granules dans l'image.

➤ L'homogénéité :

$$HOMOGENEITY = \sum_i \sum_j \frac{1}{1 + (i-j)^2} \cdot C(i, j) \quad (4.8)$$

Où $C(i, j)$ correspond aux éléments de la matrice de cooccurrence. L'homogénéité est l'inverse du contraste. Une texture homogène est caractérisée par une forte valeur.

➤ La corrélation :

$$CORRELATION = \sum_i \sum_j (i - \mu) (j - \mu) C(i, j) \quad (4.9)$$

Où μ est respectivement les moyennes des lignes et des colonnes de la matrice de cooccurrences, et $C(i, j)$ correspond aux éléments de la matrice. Cet indice permet de mesurer la corrélation de la distribution des niveaux de gris dans l'image.

➤ Le contraste :

$$CONTRAST = \sum_i \sum_j (i - j)^2 C(i, j) \quad (4.10)$$

Où $C(i, j)$ correspond aux éléments de la matrice de cooccurrences. Le contraste permet de caractériser la netteté de la texture. Lorsque les variations des niveaux de gris sont importantes, le contraste est élevé.

➤ Le cluster prominence :

$$CLUSTER PROMINENCE = \sum_i \sum_j (i + j - 2\mu)^4 C(i, j) \quad (4.11)$$

Où μ est la moyenne calculée par l'équation (4.4), et $C(i, j)$ correspond aux éléments de la matrice de cooccurrences.

➤ Le cluster shade :

$$CLUSTER SHADE = \sum_i \sum_j (i + j - 2\mu)^3 C(i, j) \quad (4.12)$$

Où μ est la moyenne calculée par l'équation (4.4), et $C(i, j)$ correspond aux éléments de la matrice de cooccurrences.

Cette méthode de matrice de cooccurrence présente une grande simplicité de mise en œuvre et donne d'assez de bons résultats sur la plupart des types d'images.

7.3 Les mesures d'évaluation

Afin d'évaluer notre système d'indexation d'images par le contenu nous devons définir les critères selon lesquels nous allons mesurer sa performance. Il existe plusieurs mesures, mais les plus utilisées sont principalement la précision (Pr) «*en anglais Precision*» et le rappel (Re) «*en anglais Recall*».

La précision nous donne la proportion des résultats pertinents par rapport au nombre total de résultats affichés. La formule pour calculer la précision est :

$$Pr = \frac{\sum(\text{Images pertinentes affichées dans la classe})}{\sum(\text{Images affichées dans la classe})} \quad (4.13)$$

Le rappel nous fournit la proportion des résultats pertinents affichés par rapport au nombre total de résultats pertinents dans la BD. La formule pour calculer le rappel est comme suit :

$$Re = \frac{\sum(\text{Images pertinentes affichées dans la classe})}{\sum(\text{Images affichées dans la BD})} \quad (4.14)$$

Nous avons remarqué dans certaines littératures [64][65][66] que le calcul du rappel est difficile vu l'énormité de la base de données et cela est dû au fait que le nombre total de résultats pertinents à une requête donnée est difficile à estimer.

7.4 Les méthodes étudiées (comparées)

Le but de notre recherche est de réaliser un système qui permet d'indexer une base de données d'images par le contenu. Afin d'évaluer notre système, nous avons besoin de le comparer avec d'autres systèmes existants, dans notre cas nous allons faire la comparaison de notre algorithme avec celui de K-MEANS et celui de la Mixture de Gaussienne.

7.4.1 Classification avec l'algorithme de K-MEANS

Au début de notre travail, nous avons commencé par prendre notre collection d'images et nous lui avons appliqué l'algorithme K-MEANS. Nous avons donc supposé que notre collection d'images provient d'un nombre de groupes connu (dans notre cas c'est le

nombre de classes qui sont représentées par les points rouges dans la figure 30 et qu'on appelle les centroides). Chacune de ces images est représentée par un point dans la figure 30.

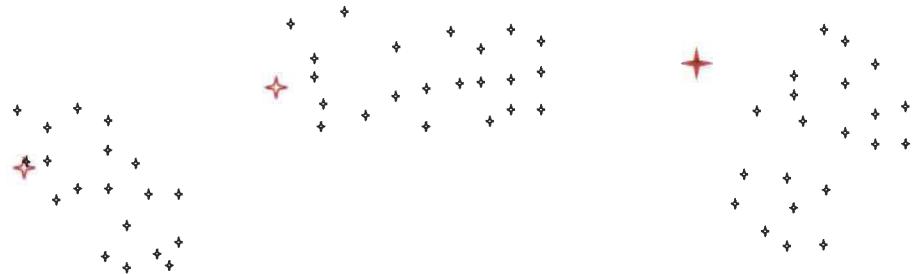
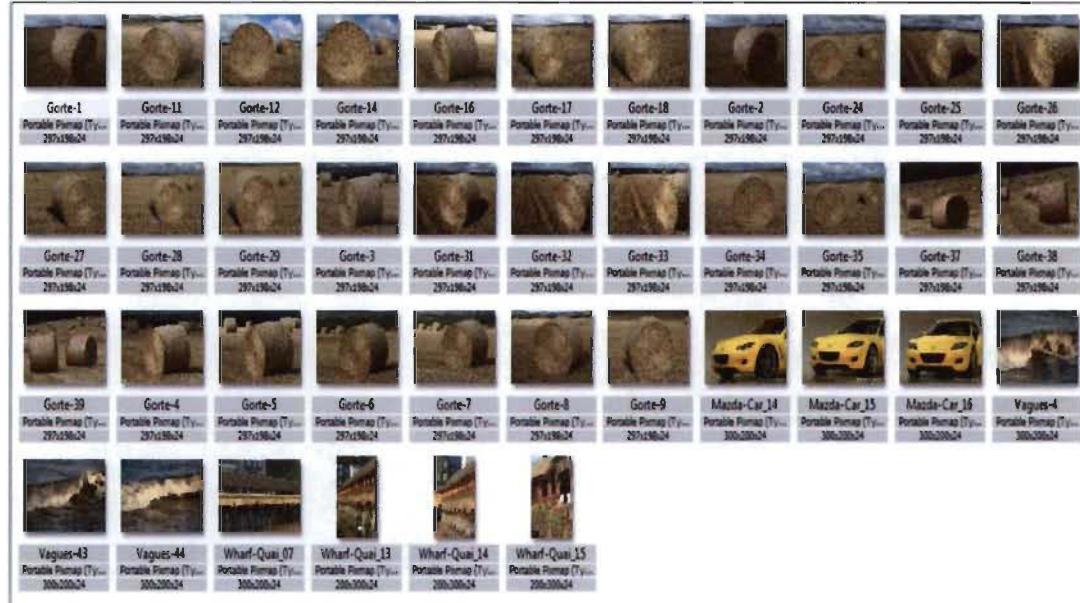


Figure 30 : Représentation des images avant la classification avec K-MEANS.

Alors l'algorithme de K-MEANS est :

- 1- Choisit au hasard K centres (les points qui sont en rouge dans la figure 30 et)
- 2- Affecte chaque donnée (dans notre cas chaque image) à la classe dont le centre est le plus proche
- 3- Recalcule les nouveaux centres
- 4- Et boucle jusqu'à convergence, c'est-à-dire aucune image n'a changé de classe lors de la dernière itération comme présenté dans la figure 30.

La figure 31 montre quelques classes après avoir effectué la classification à l'aide de l'algorithme de K-MEANS en se basant sur les caractéristiques de la couleur (RGB et HSV) et de la texture (la matrice de cooccurrences) qu'on a décrit précédemment et sur la distance euclidienne comme mesure de similarité.



Classe 11 : représente la classe des « Gorte »



Classe 8 : représente la classe des « Drawing »

Figure 291 : Exemple de classes après application de l'algorithme de K-MEANS.

7.4.2 Classification avec l'algorithme de la Mixture de Gaussienne

Vu que nous nous sommes basés sur la classification avec la mixture de Gaussienne pour développer notre système, nous avons appliqué l'algorithme de la GMM sur notre base de données d'images afin de comparer les deux systèmes.

Pour développer cet algorithme, nous avons procédé de la manière suivante :

- 1- Nous avons utilisé l'algorithme K-MEANS (que nous avons présenté à la section point 4.3.1) pour une classification initiale.
- 2- Pour chaque classe K , nous avons initialisé P_k, μ_k et Σ_k

- 3- Pour chaque image $i = (i_1, \dots, i_n)$

Pour chaque classe K

Calculons $P(i|K) = G_{\theta_K}(i)$

Nous affectons l'image i à la classe qui donne la probabilité la plus élevée

- 4- Nous avons calculé les probabilités à postériori $P((\theta_K|i))$

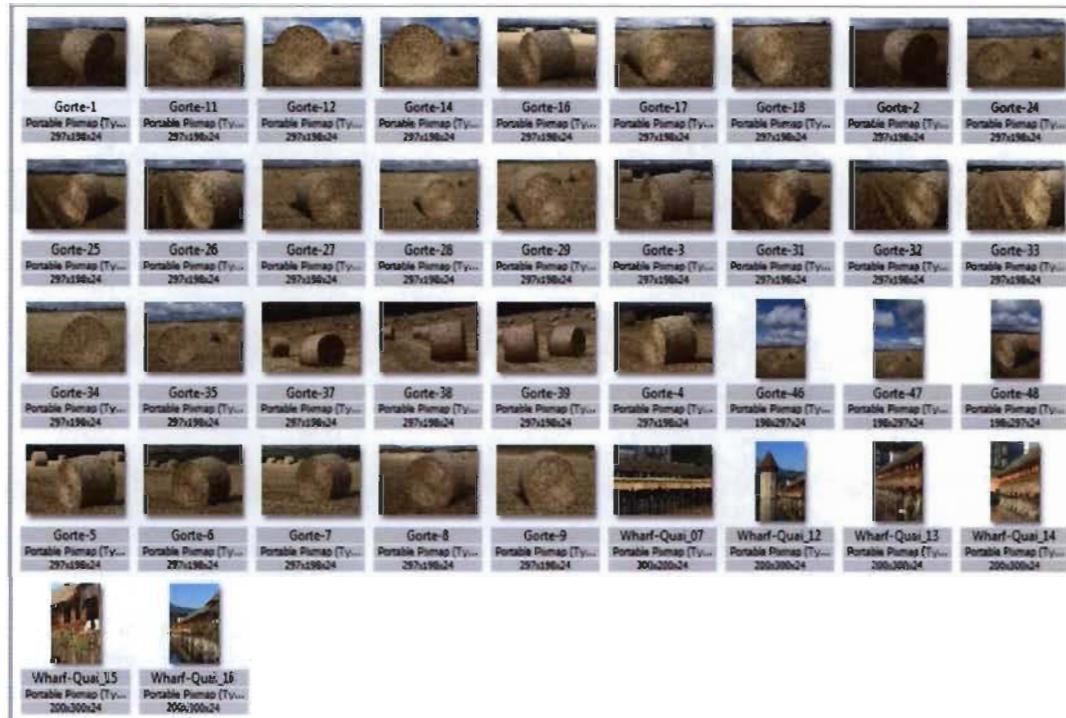
- 5- Nous avons calculé pour chaque classe K , P_k, μ_k et Σ_k

- 6- Nous avons réaffecté les images comme au troisième point

- 7- Nous avons répété les opérations jusqu'à convergence, c'est-à-dire jusqu'à ce qu'aucune image ne change de classe.

Pour plus de détails sur l'algorithme de la GMM, vous trouverez une description détaillée à la section (3.2.1.5) du chapitre 3.

La figure 32 montre quelques classes après avoir effectué la classification à l'aide de l'algorithme de la Mixture de Gaussienne.



Classe 11



Classe 8

Figure 302: Exemple de classes après application de l'algorithme GMM.

Après avoir appliqué l'algorithme GMM, nous avons remarqué que le résultat obtenu avec le K-MEANS est moins bon que celui de la GMM. Pour la classe 11 nous avons $\frac{10}{39}$ de bruit (mauvaises images) en appliquant l'algorithme K-MEANS par contre nous avons $\frac{7}{49}$ de bruit avec celui de la GMM. Pour la classe 8, nous avons constaté aussi que l'algorithme de la GMM donne de meilleur résultat vu qu'on a $\frac{1}{28}$ mauvaises images contre $\frac{9}{34}$ mauvaises images pour le K-MEANS.

7.4.3 Classification avec notre algorithme

Nous venons de présenter précédemment les deux algorithmes, le K-MEANS et la GMM, dans ce qui suit nous allons faire une brève description de notre algorithme qui va nous permettre de classifier notre banque de données d'images. En effet, nous allons procéder de la façon suivante.

- 1- Tout comme la classification avec l'algorithme de la mixture de Gaussienne, nous avons initialisé notre classification en utilisant l'algorithme de K-MEANS.
- 2- Nous avons appliqué des tests d'adéquation en utilisant la fonction de dépendance de Spearman pour déterminer la copule la plus adéquate pour nos données (le résultat obtenu est la copule de FRANK).
- 3- Pour chaque classe K
 - Pour chaque image $i = (i_1, \dots, i_n)$
 - Calculons les fonctions de répartition $F_1^j(i_1), \dots, F_n^j(i_n)$
 - Calculons les fonctions de densité $f_1^j(i_1), \dots, f_n^j(i_n)$
- 4- Nous avons calculer les probabilités d'affectations $P(i|K)$

5- Nous avons affecté chaque image i à la classe j tel que

$$P(i|k) = \max P(i|j)$$

Pour plus de détails sur notre algorithme, le lecteur trouvera une description détaillée à la section 3.5 du chapitre 3. Malheureusement, nous n'avons pas pu obtenir les résultats à temps pour pouvoir faire une évaluation de notre algorithme vu la complexité dans le développement de l'algorithme plus précisément la complexité dans le calcul de la densité de la copule de Frank.

7.5 Expérimentation et évaluation

Afin d'évaluer les performances d'un système d'indexation d'images on a besoin de ce qu'on appelle les mesures d'évaluation (décris à la section 4.4) aussi appelée critère de performance « en anglais *performance criteria* » [64] et on a besoin aussi de la vérité terrain « en anglais *ground truth* ».

À propos de la vérité terrain, il existe plusieurs façons de l'établir comme le jugement humain, une base de données classifiée par ordinateur, etc. Dans notre cas et compte tenu de notre travail, nous avons opté pour une classification manuelle de notre base de données d'images.

Malheureusement, nous ne pouvons faire une évaluation de notre système puisque nous n'avons pas obtenu les résultats à temps.

7.6 Conclusion

Dans ce dernier chapitre nous venons de voir une description détaillée des signatures des images que nous voulions utiliser pour notre système. Avec ces signatures que nous avons extraites et qui représentent bien le contenu des images de base de données, nous avons pu développer et réaliser les deux algorithmes celui de K-MEANS et celui de la GMM. Dans les sections 4.4.1 et 4.4.2, et ce pour chacun des deux algorithmes, nous avons donné quelques résultats, cependant, pour notre algorithme, nous n'avons pas pu en donner ce qui nous a empêché de faire une évaluation de notre système.

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons présenté les aspects les plus importants pour le développement d'un système d'indexation d'images par le contenu.

Nous avons remarqué que le développement de système d'indexation d'images est devenu de plus en plus important, ceci est dû à la quantité phénoménale des données multimédias et à la nécessité de bien les gérer.

Dans le premier chapitre, nous avons fait une description détaillée de tous les aspects concernant la recherche et l'indexation d'images. En effet, nous avons présenté la démarche à suivre pour faire une recherche d'images que se soit par le texte où par le contenu et son utilité et aussi nous avons présenté la procédure à réaliser pour indexer une base de données d'images par le texte où par le contenu ainsi que son utilité. Pour chacun de ces aspects, nous avons présenté leurs avantages et leurs inconvénients. Nous avons terminé ce chapitre en décrivant quelques domaines d'application qui utilisent la recherche et l'indexation d'images.

Dans le deuxième chapitre, nous avons vu les différentes méthodes d'indexation d'une base de données d'images en l'occurrence la méthode des arbres, la méthode des réseaux de neurones ainsi que la méthode statistique. Ceci fait, nous avons critiqué ces méthodes en présentant leurs défauts et leurs inconvénients. Nous avons clôturé ce chapitre par une description de quelques systèmes qui utilisent l'indexation d'images.

Dans le troisième chapitre, nous avons tout d'abord fait un rappel détaillé sur l'indexation d'images basé sur la mixture de gaussienne, ensuite, nous avons démontré que les lois marginales ne suivent pas seulement des lois normales mais suivent des lois gamma, uniforme et bien d'autres, à partir de là, nous avons proposé une nouvelle approche qui est basée sur les copules archimédiennes. Nous finissons notre chapitre en faisant en premier lieu un rappel sur les copules archimédiennes et en second lieu en décrivant en détail les étapes de proposition.

D'après notre étude, les caractéristiques d'images sont indispensables pour la réalisation d'un système d'indexation d'image ce qui nous amène au quatrième et dernier chapitre. En effet, nous avons présenté au début de ce chapitre les caractéristiques des images que nous avons utilisées à savoir la couleur (l'espace de couleur RGB et HSV), la forme et la texture (la matrice de cooccurrence). Notre contribution consiste à proposer un nouvel algorithme de classification d'images basé sur les modèles de copules ce qui n'a jamais été réalisé avant. Nous avons aussi présenté des algorithmes de classification classiques à savoir l'algorithme K-MEANS et l'algorithme de mixture de gaussiennes. Des résultats de ces derniers algorithmes ont été présentés. Toutefois, le développement de notre algorithme basé sur les copules archimédiennes était très difficile à réaliser vu la complexité à calculer la densité de la copule de Frank, malgré les approximations que nous avons fait ça nous a pas permis d'évaluer notre système.

Cependant, des améliorations peuvent être faites à notre proposition. Parmi les pistes les plus prometteuses en premier lieu, l'ajout par exemple des caractéristiques d'images de bas niveau comme la forme, en second lieu l'ajout de l'annotation textuelle aux images. Cela permettra d'obtenir une meilleur indexation et facilitera la navigation dans l'index et l'accès aux données de la collection d'images.

RÉFÉRENCE

- [1] J. R. Smith and S. Chang, "Searching for Images and Videos on the World Wide Web," Technical Reports N 459-96-25, Columbia University, Center for Image Technology for New Media, New-York, August 1996.
- [2] L. L. Thi, "Indexation et recherche d'images par le contenu," Mémoire de Master, - *Institut polytechnique de Hanoi*, Vietnam 2005.
- [3] A. Boucher and L.Thi-Lan, "Comment extraire la sémantique d'une image," *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT)*, Tunisia, March 2005.
- [4] D. Filliat, "Robotique Mobile," Thèse de doctorat, *École Nationale Supérieure de Techniques Avancées de paris (ENSTA)*, France 2005.
- [5] M.A Stricker and A.Dimai, "Color Indexing with Weak Spatial Constraints," *Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases*, vol 2670, pp. 29–40. San Diego/La Jolla, CA, 1996.
- [6] Y.Gong, G.Proietti and C.Faloutsos, "Image Indexing and Retrieval Based on Human Perceptual Color Clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.578 - 583, Santa Barbara, CA , USA, 1998.
- [7] M.Tuceryan and A. K Jain, "Texture analysis," *The Handbook of Pattern Recognition and Computer Vision. 2nd Edition*, by C. H. Chen, L. F. Pau, P. S. P. Wang (editors.), pp.207-248, World Scientific Publishing Co. (1998)
- [8] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol.67, no.5, pp.786–804, May 1979.
- [9] F. Liu and R. W. Picard, "Periodicity, Directionality and Randomness: Wold Features for Image Modeling and Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18, no.7, pp.722–733, July, 1996.

- [10] T.Yamawaki, H.Tamura and S.Mori, "Textural features corresponding to visual perception," *IEEE Transaction on Systems, Man and Cybernetics*, vol.8, no.6, pp: 460-473, June, 1978.
- [11] B.S.Manjunath and W.Y.Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Analysis and. Machine Intelligence*, vol.18, no.8, pp. 837-842, California Univ., Santa Barbara, CA,1996.
- [12] M. R.Turner, "Texture discrimination by Gabor functions," *Biological Cybernetics*, vol.55, no.2-3, pp.71-82, 1986.
- [13] M.K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Transactions on Information Theory*, vol.8, no.2, pp.179-187, 1962.
- [14] B.Gunsel and M.Tekalp, "Shape similarity matching for query-by-example," *Pattern Recognition*, vol.31, no.7, pp.931-944, 1998.
- [15] S. Loncaric, "A Survey of Shape Analysis Techniques," *Pattern Recognition*, vol.31, no.8, pp.983-1001, 1998.
- [16] J. Shah, "Gray skeletons and segmentation of shapes". *Computer Vision and Image Understanding*, vol.99, no.1, pp.96-109, 2005.
- [17] J. Eakins and M. Graham, "Content-based Image Retrieval". Technical Report N 39, University of Northumbria at Newcastle, UK, 1999.
- [18] IEEE 1985 "Computer graphics in the detective business" *IEEE Computer Graphics and Applications*, vol.5, no.4, pp.14-17, 1985.
- [19] S. C. Orphanoudakis, C. Chronaki, and S. Kostomanolakis, "I2C: A System for the Indexing, Storage, and Retrieval of Medical Images by Content," *Institute of Computer Science, Foundation for Research and Technology-Hellas and Department of Computer Science, University of Crete*, pp.1-18, January 1994
- [20] Y.Liu et al, "Content-based 3-D neuroradiologic image retrieval: preliminary results" in *IEEE International Workshop on Content-based Access of Image and Video Databases (CAIVD '98)*, Bombay, India, pp.91-100, 1998.

- [21] J. H. Friedman, J. L. Bentley, and R. A. Finkel, “An algorithm for finding best matches in logarithmic expected time,” *ACM Trans. Math. Software*, vol. 3, no. 3, pp. 209-226, 1997.
- [22] A. Guttman, “R-Trees: A Dynamic Index Structure for Spatial Searching,” *in Proc. ACM SIGMOD Conf.*, pp. 47-57.Boston, MA, June 1984.
- [23] C. Fleury “Le kd-Tree : une méthode de subdivision spatiale,” *Présentation de Master 2 Recherche Informatique - Module CTR*, Université de Rennes 1 - INSA de Rennes, France, 19 Décembre 2007.
- [24] N. Beckman, H.P. Kriegel, R. Schneider, B. Seeger, “R*trees: an efficient and robust access method for points and rectangles,” *In. proceedings of ACM SIGMOD*, pp. 322-331, 1990.
- [25] Y.J. Lee, C.W. Chung, “The DR-tree: A main memory data structure for complex multi-dimensional objects,” *Geoinformatica*, 201, vol 5, no. 2, pp. 181-207.
- [26] S. Haykin, *Neural Networks: A Comprehensive Foundation*, College Publishing Company, New York, 1994.
- [27] R. Lippmann, “An Introduction To Computing with Neural Nets” *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4-22, 1987.
- [28] Dempster *et al.*, “Maximum Likelihood from Incomplete Data via the EM Algorithm”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp.1-38. 1977.
- [29] J. T. Robinson, “The K-D-B-tree: A search structure for large multidimensional dynamic indexes,” *in Proc. ACM SIGMOD*, pp. 11-18, Ann Arbor, MI, 1981.
- [30] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R* -tree: An efficient and robust access method for points and rectangles,” *in Proc. ACM SIGMOD*, pp.322-331, Atlantic City, NJ, 1990.

- [31] D. A. White and R. Jain, “Similarity indexing with the SS-tree,” *in Proc. Int. Conf. on Data Eng.*, pp.516-523, New Orleans, LA, 1996.
- [32] N. Katamaya and S. Satoh, “The SR-tree: An index structure for high-dimensional nearest neighbor queries,” *in Proc. ACM SIGMOD*, pp.1-12, Tuscon, AZ, 1997.
- [33] T. Gevers and A. W. M. Smeulders, “The PicToSeek WWW image search system,” *in Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, pp. 246–269, Florence, Italy, 1999.
- [34] R. Lippmann, “An Introduction To Computing with Neural Nets,” *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4-22, Avril 1987.
- [35] Y. LeCun, “Constrained Neural Network for Unconstrained Handwritten Digit Recognition,” *Proc. of International Workshop on Frontiers in Handwriting Recognition*, pp 145-154, Montreal, Canada, April 1990.
- [36] B. GOSELIN, “Application de réseaux de neurones artificiel à la reconnaissance automatique de caractère manuscrits,” Thèse de Doctorat, *Faculté Polytechnique de Mons*, BEL, 1996.
- [37] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York, 1949.
- [38] J. J. Hopfield, “Les réseaux de neurones et les systèmes physiques avec émergents collective capacités de calcul, ” *Actes de l'Académie nationale des sciences des États-Unis*, vol.79, no. 79, pp. 2554-2558, April 1982.
- [39] M. S. Lew, “Next generation web searches for visual content,” *IEEE Comput.*, vol. 33, no. 11, pp. 46-53, 2000.
- [40] S. Sclaroff, L. Taycher, and M. La Cascia, “Image-Rover: A Content Based Image Browser for the World Wide Web,” *in Proc. IEEEWorkshop on Content-Based Access of Image and Video Libraries*, pp. 2-9, San Juan , Puerto Rico, 1997.

- [41] M.L. Kherfi and D. Ziou, “Image Collection Modeling and its Application to Indexing, Browsing and Semantic Retrieval,” *IEEE Transactions on Multimedia*, vol. 9, no. 4, pp. 893-900, June 2007.
- [42] Site web officiel de la compétition RoboCup : <http://www.robocup.org>
- [43] Site web officiel de l’université de Lille3, <http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie005.html>
- [44] <http://www.mixmod.org/index.php>
- [45] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, “Efficient and effective querying by image content,” *Journal of Intelligent Information Systems*, vol. 3, no.3-4, pp.231-262, 1994.
- [46] M.L. Kherfi, D. Ziou and A. Bernardi, “Image Retrieval from the World Wide Web: Issues, Techniques and Systems,” *ACM Computing Surveys*, vol. 36, no. 1, pp. 35-67, March 2004.
- [47] C. Faloutsos *et al.* “Efficient and effective querying by image content” *Journal of Intelligent Information System*, Vol.3, Issue 3-4, pp. 231-262. 1994.
- [48] W. Niblack, *et al.* “Updates to the QBIC system” in Storage and Retrieval for Image and Video Databases VI (I.K Sethi, and R.C Jain,), *Proc SPIE*, vol. 3312, pp. 150-161, 1998.
- [49] J. Eakins and M. Graham, “Content-based Image Retrieval,” Technical Report N 39, University of Northumbria at Newcastle, UK.
- [50] J.R.Smith and S.-F. Chang, “Visulaseek: A fully automated content-based image query system,” *In Proceedings of the ACM International Conference on Multimedia*, pp. 87-98, Boston, MA, 1996.

- [51] A.Pentland, R. Picard, and S. Sclaroff, “Photobook: Content-based manipulation of image databases,” *In Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases*, pp. 34-47, San Jose, CA, February 1994.
- [52] R.W.Picard, “A society of models for video and image libraries,” *IBM Systems Journal*, vol. 35, pp. 292-312, 1996.
- [53] M.L. Kherfi, D. Ziou, and A. Bernardi. *Atlas: An Image Retrieval Engine for Databases and the World Wide Web*. Poster à la Journée de la recherche, Université de Sherbrooke, novembre 2003.
- [54] Lien internet :
- http://www.google.ca/imgres?q=r%C3%A9seau+de+neurones+biologique&um=1&hl=fr&biw=1280&bih=802&tbo=isch&tbnid=oC1VYccj41vQNM:&imgrefurl=http://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_%28biologie%29&docid=iG8sOsHALjhKrM&imgurl=http://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Neuron-figure-fr.svg/220px-Neuron-figure-fr.svg.png&w=220&h=311&ei=_c8DT-qeH4jVgQfZvpXGAw&zoom=1&iact=hc&vpx=174&vpy=147&dur=794&hovh=248&hovw=176&tx=105&ty=118&sig=113302860376134032180&page=6&tbnh=138&tbnw=98&start=122&ndsp=24&ved=1t:429,r:12,s:122
- [55] J. R.Smith, and S.F, Chang. “Automated image retrieval using color and texture,” Technical Report CU/CTR 408-95-14, Columbia University, July, 1995.
- [56] S.Lim, and G. Lu, “Effectiveness and efficiency of six colour spaces for content based image retrieval” In Proceedings of nternational Workshop on Content-Based Multimedia Indexing (CBMI), 2003.
- [57] R.M. Haralick, “Statistical and structural approaches to texture,” Proceedings of the IEEE, Volume: 67, no. 5, pp. 786- 804.
- [58] J.S.Weszka, C.R Dyer and A.Rosenfeld, “A comparative study of texture measures for terrain classification,” *IEEE Trans. Systems, Man Cybernet*, 6 269–285, 1976.

- [59] Bharati, M. H., Jay Liu, J., & MacGregor, J.F, “Image texture analysis: methods and comparisons,” *Chemometrics and intelligent laboratory systems*, vol. 72, no1, pp. 57-71, 2004.
- [60] Sharma. M., Singh.S, “Evaluation of texture methods for image analysis,” *Proceedings of the 7th Australian and New Zealand Intelligent Information Systems Conference*, 2001.
- [61] Zhang, J., & Tan, T., “Brief review of invariant texture analysis methods,” *Pattern Recognition*. vol. 35, no. 3, March, pp. 735-747, 2001.
- [62] R.M.Haralick, K. Shanmugan and I. Distein, “Textural features for image classification,” *IEEE Transactions System, Man and Cybernetics*, vol. SMC-3, pp. 610-621, 1973.
- [63] M. Unser, “Sum and difference histograms for texture classification,” *IEEE transactions on pattern analysis and machine intelligence*, vol. PAMI-8, no.1, Janvier, 1986.
- [64] J. R. SMITH and S. CHANG, “Searching for Images and Videos on the World Wide Web”, Columbia University, Center for Image Technology for New Media, New-York, August 1996.
- [65] B. FURHT, S. W. SMOLIAR, and H. ZHANG, “Video Processing in Multimedia Systems”, Kluwer International Series In Engineering And Computer Science, pp. 377, October 1995.
- [66] M. L. KHERFI AND D. ZIOU, “Image Retrieval from the World Wide Web: Issues, Techniques, and Systems”, dans le journal ACM Computing Surveys, vol.36, no.1, pp.35-67, Mars 2004.