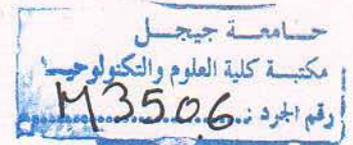


République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur

et de la recherche scientifique

Université Mohammed Seddik Ben Yahia -Jijel



Faculté des sciences et de la technologie

Département d'électronique

Mémoire de fin d'études pour l'obtention du diplôme

de Master en électronique

Option : Electronique et analyse des systèmes



Thème :

***Classification des images par les motifs
binaires locaux (LBP)***

Encadré par :

Mme. Sabrina BOUATMANE

Réalisé par :

BENHABILES Mohammed

BOUDJAOUI Chemseddine

Promotion 2017

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur

2017/01

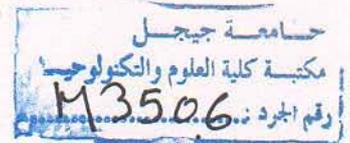
et de la recherche scientifique

01/02

Université Mohammed Seddik Ben Yahia -Jijel

M/ELE.AS

2017/01



Faculté des sciences et de la technologie

Département d'électronique

Mémoire de fin d'études pour l'obtention du diplôme

de Master en électronique

Option : Electronique et analyse des systèmes



Thème :

***Classification des images par les motifs
binaires locaux (LBP)***

Encadré par :

Mme. Sabrina BOUATMANE

Réalisé par :

BENHABILES Mohammed

BOUDJAOUI Chemseddine

Promotion 2017

Remerciement

Nous devons remercier tout d'abord, ALLAH, le tout puissant, qui nous a donné la puissance, la volonté et la patience pour élaborer ce travail.

Nous adressons tout d'abord nos remerciements les plus sincères, à Mme BOUATMANE Sabrina, qui a très volontairement accepté d'être le promoteur de ce mémoire. Sa grande connaissance dans le domaine, ainsi que son expérience, ses contributions, ses orientations précieuses, sa disponibilité et sa compréhension ont joué un rôle important dans la conception de ce travail.

Plus près de nous, nous souhaitons remercier les professeurs DIB Samira et MARDJANA Hasina et MERABET Souad pour leur soutien et leur encouragement. Vous avez toujours cru en nos capacités. Nous vous en remercions infiniment SEDIRA Samira, vous nous avez beaucoup aidé. Vous étiez là à chaque moment important et nous vous en remercions. Votre hospitalité n'a d'égale que votre gentillesse.

Nous exprimons tous nos sincères remerciements pour les membres de jury qui ont accepté d'examiner et d'évaluer ce travail, nous remercions ainsi tous nos enseignants de département électronique.

Nous adressons nos vifs remerciements à tous ceux qui nous ont aidé et supporté de prêt ou de loin durant l'élaboration de ce travail.

Enfin, c'est avec une certaine émotion que nous nous adressons à nos parents ainsi qu'à mes grands-parents, sans lesquels nous ne serons pas arrivés jusque-là. Votre persévérance et vos sacrifices durant toutes ces années se concrétisent aujourd'hui par ce nouvel accomplissement. Nous vous aimons de tous nos cœurs, ainsi que nos frères et nos sœurs que nous

Dédicace

1. *À mon père.*
2. *À ma mère !*
3. *À ma petit sœur **Ryma** qui a toujours était à mes côtés.*
4. *À tous mes chers frères et sœurs, ainsi que leurs enfants **Youssef, Amir, Ashraf, Iyad.***
5. *À mes deux grandes familles, maternelle et paternelle.*
6. *À ma grand-mère !*
7. *À mon ami **Mohammed** avant qu'il soit binôme et collègue, il est un frère.*
8. *À ma chère cousine **Hadjer** et à ma tante **Nora.***
9. *À notre promoteur Mme **BOUATMANE Sabrina.***
10. *À mes amis qui n'ont cessé de m'aider et m'encourager en particulier **Ibtissem, Ishac** et à mes collègues.*
11. *À mes profs.*
12. *À tous ceux qui, par un mot, m'ont donné la force de continue*

Chemseddine

Dédicace

1. *Je dédie ce modeste travail à ma **chère maman** et à la mémoire de **mon papa**.*
2. *À mes chères sœurs, mes chers cousins et cousines.*
3. *À mes deux grandes familles, maternelle et paternelle.*
4. *À tous mes proches, mes amis, mes collègues de classe.*
5. *À **BOUDJAOUI Chemseddine**, mon binôme mon collègue et plus que sa un cher ami.*
6. *À notre promoteur Mme **BOUATMANE Sabrina**.*
7. *À tous nos profs de la spécialité.*
8. *Enfin à tous ceux qui travaillent pour une Algérie meilleure.*

Mohammed

Sommaire

Introduction générale..... 1

Chapitre I : Extraction des caractéristiques de texture

I.1. Introduction..... 4

I.2. Définitions de la texture 4

I.2.1. Définition 1 4

I.2.2. Définition 2..... 5

I.2.3. Familles de textures 5

I.2.3.1. Macro textures..... 6

I.2.3.2. Micro textures..... 7

I.2.4. Propriétés de perception des textures..... 8

I.3. Méthodes d'extraction de caractéristiques de textures 10

I.3.1. Attributs géométriques 11

I.3.2. Attributs spatio-fréquentiels 11

I.3.2.1. Dans le domaine spatial..... 11

I.3.2.2. Dans le domaine fréquentiel..... 11

I.3.2.3. Dans le domaine spatio-fréquentiel..... 12

I.3.3. Attributs statistiques 12

I.3.3.1. Attributs statistiques de premier ordre..... 12

I.3.3.2. Attributs statistiques de second ordre 13

I.3.4. Matrices de co-occurrences..... 13

I.3.4.1. Attributs d'Haralick extraits des matrices de co-occurrences..... 15

I.3.5. Attributs basés sur la modélisation spatiale 16

I.3.5.1. Les modèles de MARKOV	16
I.3.5.2. Méthode des motifs binaires (LBP)	16
I.4. Les motifs binaires locaux (LBP)	17
I.4.1. Définition	17
I.4.2. Principe général	17
I.4.3. LBP uniforme et invariant par rotation	19
I.5. Conclusion	21

Chapitre II : Classification des données

II.1. Introduction	22
II.2. Définition de la classification	22
II.3. Les types de la classification.....	23
II.3.1. Classification non supervisée.....	24
II.3.2. Classification supervisée.....	24
II.4. Méthodes de classification.....	25
II.4.1. Séparateurs à vaste marge (SVM).....	26
II.4.2. K-plus proche voisins (k-NN).....	27
II.3.5. Types d'informations fournies par les classificateurs	29
II.5. Evaluation de la classification.....	30
II.5.1. Base d'apprentissage et base de test	30
II.5.2. La validation croisée	30
II.5.3. Matrice de confusion.....	31
II.5.4. Courbe ROC (Receiver Operating Characteristics)	33
II.6. Conclusion.....	34

Chapitre III Classification des images texturées

III.1. Introduction.....	35
III.2. Présentation de la base de données.....	35
III.3. Implémentation et résultats.....	35
III.4. Résultats et discussions	40
III.4.1. La validation simple	40
III.4.2. La validation croisée.....	41
III.4.3. Discussion	51
III.5. Conclusion	52
Conclusion générale	53
Bibliographie.....	54

Liste des figures

Figure I.1. Exemples de textures répétitives prises de la base de texture VISTEX [4].	5
Figure I.2. Exemple de familles de textures : [a-d] textures déterministes, [e-h] textures observables, [i-l] textures stochastiques.....	6
Figure I.3. Exemples de macro textures.	7
Figure I.4. Exemples de micro textures.	7
Figure I.5. Exemples de sables à 2 résolutions différentes.	8
Figure I.6. Information sur une texture selon plusieurs régionalités.....	9
Figure I.7. Représentation d'une texture selon plusieurs résolutions 10	10
Figure I.8. Les différentes classes d'attributs de texture..... 11	11
Figure I.9. Huit pixels plus proches voisins utilisés pour calculer la matrice de cooccurrence. Les plus proches voisins de 'x' selon la direction $\theta = 0^\circ$ sont les pixels 1 et 5, pour $\theta = 45^\circ$ les pixels 4 et 8, pour $\theta = 90^\circ$ les pixels 3 et 7 et enfin pour $\theta = 135^\circ$ les pixels 2 et 6. 14	14
Figure I.10. Construction d'un motif binaire et calcul du code LBP..... 18	18
Figure I.11. Représentation du pixel central g_c et de son voisinage g_p 18	18
Figure I.12. Exemples de Construction et uniformité d'un motif LBP,..... 19	19
Figure I.13. Liste des 36 uniques motifs binaires locaux invariant en rotation possibles pour $LBP_{8,R}^n$, Les points noirs et blancs correspondent respectivement aux valeurs de bit 0 et 1 dans la sortie sur 8 bits de l'opérateur $ROR(.)$, source [29]. 20	20
Figure II.1. Schéma de principe d'un système de classification..... 23	23
Figure II.2. Classification des types de la classification [31]. 23	23
Figure II.3. La classification supervisée..... 25	25
Figure II.4. Illustration de la marge maximale et les vecteurs supports qui sont lui associés. 26	26
Figure II.5. Illustration de la méthode des k plus proches voisins (k -NN) avec $k = 5$	29

Figure II.6. Principe de la validation croisée pour $m = 10$ 31

Figure II.7. Décision d'un classifieur binaire en fonction du seuil de discrimination s : la courbe bleue représente la probabilité a posteriori de la classe « négative » et la courbe rouge représente la probabilité a posteriori de la classe « positive ». 32

Figure II.8. Courbes ROC : à gauche, les courbes A, B, C et D sont respectivement associées à des classifieurs de performances croissantes. A droite, les courbes A et B ont la même aire (AUC) et la même valeur EER et sont pourtant associées à des classifieurs qui n'ont pas les mêmes qualités. 33

Figure III.1. Echantillons d'image de texture de la base UIUCTex[49]. 36

Figure III.2. Exemple d'application du code LBP sur une image de la base de données. 37

Figure III.3. Application du LBP sur une image de texture (a) : image originale, (b) : 38

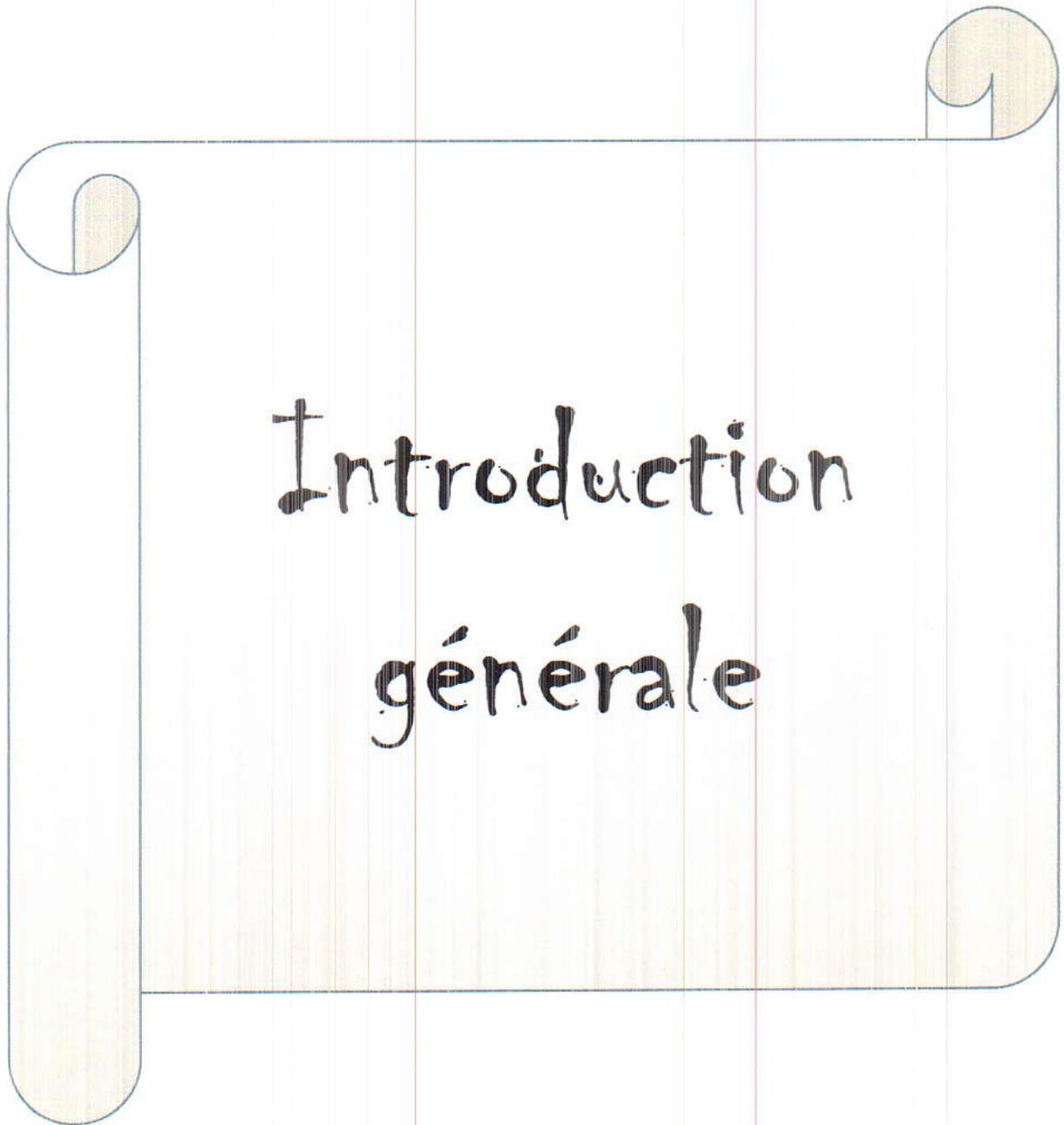
Figure III.4. Schéma général du travail effectué. 39

Figure III.5. Représentation des images de classe : Bark2, Bark3, Granite, Fur, Gravel. 51

Listes des tableaux

Tableau II.1. Matrice de confusion.....	32
Tableau III.1. Taux d'erreur (%) pour la validation simple.....	40
Tableau III.2. La sensibilité et la spécificité (%) pour la validation simple.....	41
Tableau III.3. Taux d'erreur (%) pour la validation croisée.....	42
Tableau III.4. La sensibilité et la spécificité (%) pour la validation croisée.....	42
Tableau III.5. Taux d'erreur (%) des différentes classes pour la validation croisée...	44
Tableau III.6. Taux d'erreur (%) des différentes classes pour la validation simple. ...	46
Tableau III.7. Matrice de confusion pour LBP simple.....	47
Tableau III.8. Matrice de confusion pour LBP uniforme.....	48
Tableau III.9. Matrice de confusion pour LBP invariant par rotation.....	49
Tableau III.10. Matrice de confusion pour LBP uniforme et invariant par rotation	50





Introduction générale

Le problème que cherche à résoudre la reconnaissance des formes est d'associer une étiquette à une donnée qui peut se présenter sous forme d'une image ou d'un signal. La reconnaissance de forme (RF) est un thème particulièrement vaste et pluridisciplinaire qui fait objet de nombreux travaux de recherche. La notion de modélisation d'un comportement intelligent est souvent sous-entendue dans un problème RF et donne une idée de la difficulté de la tâche.

Des méthodes générales ont été développées en reconnaissance des formes pour extraire automatiquement des informations des données sensibles afin de caractériser les classes de forme (apprentissage) et d'assigner automatiquement les données à ses classes (reconnaissance ou classification).

Un système de reconnaissance de forme peut être résumé en deux phases principales : l'extraction de caractéristiques et la classification.

L'extraction de caractéristiques consiste en des transformations mathématiques calculées sur les pixels d'une image numérique. Les caractéristiques visuelles permettent généralement de mieux rendre compte de certaines propriétés visuelles de l'image, utilisées pour des traitements ultérieurs entrant dans le cadre d'applications telles que la détection d'objets ou la recherche d'images par le contenu.

La classification, se réfère au classement ou groupement d'éléments de données dans des ensembles similaires. L'étape d'extraction de caractéristiques (attributs) réduit donc les données en mesurant certaines propriétés ou caractéristiques des images étiquetées. Ces attributs sont ensuite transmis à un classifieur qui évalue ces caractéristiques et prend une décision relative à la classe de chaque image.

On s'intéresse dans notre travail à la conception et l'évaluation d'un système de classification automatique des images texturées.

L'objectif visé dans notre cas par la classification est de pouvoir distinguer entre les classes d'images correspondantes aux différentes textures.

L'analyse des textures est un sujet ancien de l'analyse d'image et de la vision par ordinateur. Les techniques existantes sont utilisées pour identifier une texture quelconque ou

diviser une image texturée en plusieurs régions comportant des caractéristiques identiques. Nous les retrouvons dans plusieurs domaines d'application comme l'analyse de scènes et de vidéos, l'imagerie médicale, l'industrie etc.

L'analyse d'images texturées a pour but d'extraire des caractéristiques ou attributs de texture afin de classer les images. La classification des images est utilisée dans de différents domaines tels que la reconnaissance des empreintes, des visages, indexation, reconnaissance de tumeurs etc....

Dans notre cas, nous avons choisi l'opérateur du motif binaire local (Local Binary Pattern) pour l'extraction des caractéristiques. L'idée de cet opérateur est d'assigner à chaque pixel un code dépendant des niveaux de gris de son voisinage. Le niveau de gris du pixel central d'un bloc 3×3 est comparé à celui de ses voisins suivant certaine équation. Les pixels de ce bloc sont seuillés par sa valeur de pixel central, multiplié par des puissances de deux et ensuite additionné pour obtenir un code pour le pixel central.

Ces codes ou leurs statistiques, le plus souvent l'histogramme, sont ensuite utilisées comme vecteur de caractéristiques de l'image. La deuxième phase qui est la classification est effectuée par la méthode du k plus proche voisin KNN (k Nearest Neighbor). Pour la classification d'une nouvelle observation ou image x , l'idée fondatrice est de faire voter les plus proches voisins de cette observation. La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x .

Pour évaluer les performances de notre système, nous avons utilisé une base de données qui contient 1000 images réelles réparties sur 25 classes de textures différentes (marbre, bois, eau, verre, ...). Les images sont prises à des orientations et des résolutions différentes. Nous avons divisé la base de données en groupe d'apprentissage et groupe de test selon deux procédures : une validation simple et une validation croisée. Ces procédures structurent les itérations nécessaires à l'évaluation des performances du classifieur. L'évaluation de la classification est réalisée en termes de taux d'erreur global, le taux d'erreur par classe, la sensibilité, la spécificité et la matrice de confusion.

L'objectif de notre travail se définit dans les chapitres suivants :

Dans le premier chapitre, nous allons décrire la notion de texture et les différentes méthodes utilisées pour l'analyse de texture. Nous présentons la méthode LBP qui est basée sur l'extraction des attributs spatiaux, où chaque pixel reçoit un code binaire qui dépend des valeurs des pixels voisins. Cette méthode LBP sert à classer les images texturées.

Dans le deuxième chapitre, nous allons présenter les différentes approches de la classification à savoir la méthode KNN ainsi que l'étape d'évaluation du taux de reconnaissance qui s'est effectué par la validation simple et croisée afin d'étudier les performances des méthodes utilisées pour classer les images texturées et de les comparer.

Dans le dernier chapitre, nous allons tester la méthode de KNN sur les attributs de textures calculés à partir des images de la base de données UIUCTex en utilisant la méthode d'extraction LBP et 3 autres variantes de celle-ci à savoir la méthode LBP uniforme, LBP invariante par rotation et LBP uniforme et invariante par rotation. Nous présenterons aussi les taux d'erreur des différentes méthodes utilisées.

Nous terminerons ce mémoire par une conclusion générale et des perspectives.

CHAPITRE I

Analyse et
extraction des
caractéristiques de
texture

I.1. Introduction

L'analyse des textures est un sujet ancien de l'analyse d'image et de la vision par ordinateur. Les techniques existantes sont utilisées pour identifier une texture quelconque ou diviser une image texturée en plusieurs régions comportant des caractéristiques identiques. Nous les retrouvons dans plusieurs domaines d'application comme l'analyse de scènes et de vidéos, l'imagerie médicale, l'industrie etc. Les recherches en analyse de textures se divisent en trois grandes catégories : la classification, permettant de reconnaître une texture donnée, la segmentation ayant pour objectif d'identifier les différentes classes de textures dans une même image et la synthèse de textures qui peut être également utilisée pour la compression d'images. Différents types d'images comprenant les images binaires, en niveaux de gris et en couleurs sont concernés par ces outils d'analyse. Selon les données à traiter, les techniques d'analyse de textures peuvent varier.

I.2. Définitions de la texture

Plusieurs définitions de la notion de texture ont été présentées au cours des vingt dernières années. Ces définitions sont rarement génériques et portent plutôt sur un des aspects ou une des applications de la notion de texture.

I.2.1. Définition 1

Une texture est une région de l'image, pour laquelle on peut définir une fenêtre de dimension minimale, telle qu'une observation au travers de celle-ci se traduit par une perception identique pour toutes les translations possibles de cette fenêtre à l'intérieur de la région considérée. Pour une texture, il existe toujours une telle fenêtre qui correspond à la résolution de la texture.

A la fois structurée et aléatoire, une texture peut être décrite spatialement ou statistiquement, et, d'une manière générale, on qualifiera une texture de fine ou grossière, plus ou moins homogène, rugueuse, plus ou moins régulière, dense, linéaire, isotrope ou directionnelle, tachetée, marbrée, etc.

I.2.2. Définition 2

La texture est un facteur fondamental dans la perception de l'environnement et la reconnaissance de ses objets. Contrairement à la couleur, la texture reste difficile à définir de manière précise et générique. Le nombre de définitions possibles données dans la littérature en témoignent [1, 2,3]. La définition littéraire de la texture est la "répétition spatiale d'un même motif dans différentes directions de l'espace". Cependant, cette explication est parfois considérée insuffisante étant donné qu'elle est indépendante du comportement de l'observateur humain. À titre d'exemple, la texture d'un tissu ou d'un mur de briques (figure I.1) peut satisfaire cette définition, ce qui n'est pas toujours le cas pour d'autres textures. La texture possède plusieurs aspects visuels discriminants à savoir l'aspect grossier, fin, granuleux, régulier ou irrégulier. Par conséquent, l'homogénéité de la surface est le critère à considérer pour différencier deux objets de textures différentes.

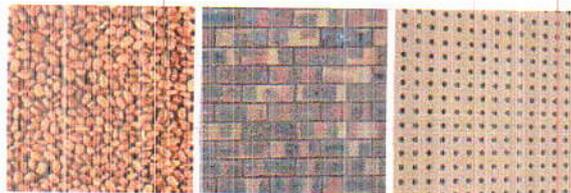


Figure I.1. Exemples de textures répétitives prises de la base de texture VISTEX [4].

I.2.3. Familles de textures

Dans la littérature, il est souvent question de plusieurs familles de textures avec notamment les textures déterministes, les textures stochastiques et les textures observables [5] (figure I.2). Les textures déterministes se caractérisent par la répétition d'un motif géométrique aussi appelé "texton". Ce genre de textures peut donc facilement être synthétisé à partir d'un motif si on connaît son orientation et sa taille.

Au contraire, les textures stochastiques sont irrégulières. Elles obéissent à certaines lois statistiques et peuvent être considérées comme une réalisation d'un champ aléatoire bidimensionnel. Une texture observable se décrit comme un mélange entre une texture stochastique et une texture déterministe. En effet, à l'intérieur de ce type de textures, nous pouvons trouver une répétition de motifs très similaires mais pas identiques.

En général, on peut classer la texture en deux grandes familles :

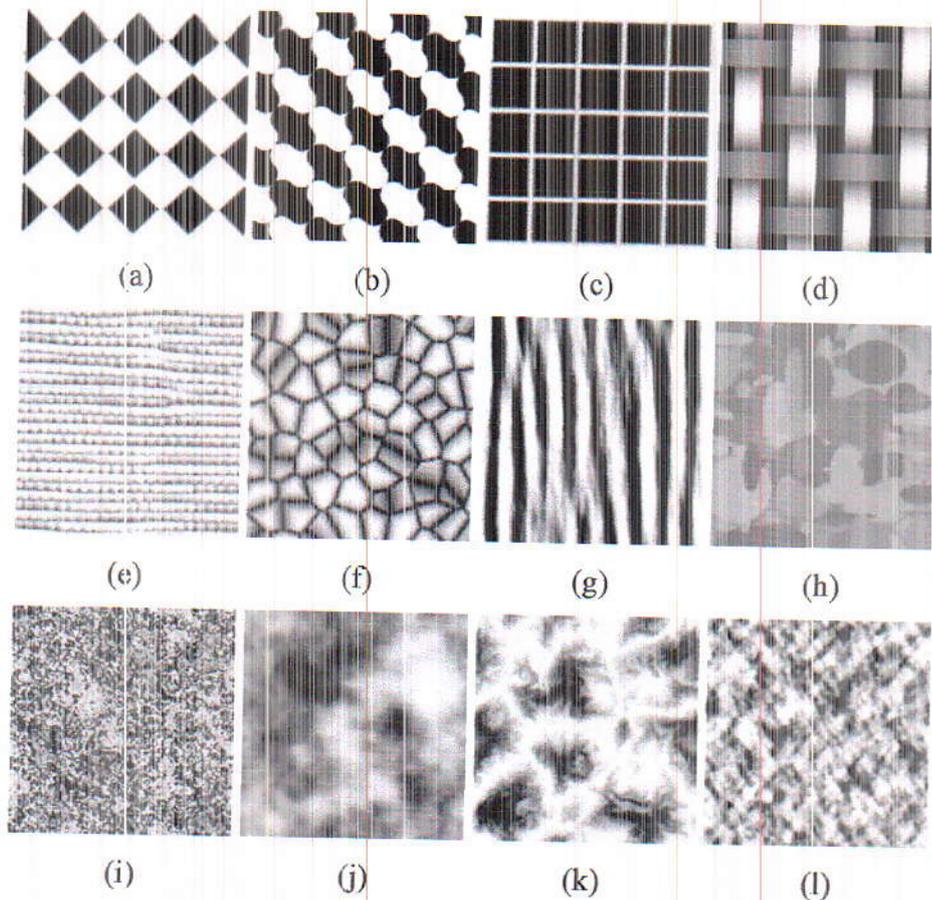
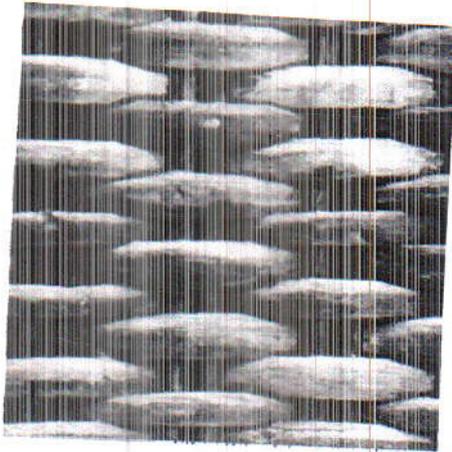


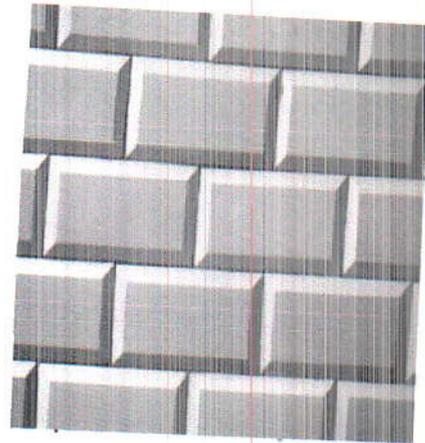
Figure I.2. Exemple de familles de textures : [a-d] textures déterministes, [e-h] textures observables, [i-l] textures stochastiques.

I.2.3.1. Macro textures

(Ou textures structurées) pour lesquelles il est facile d’extraire visuellement le motif de base et les lois d’assemblage des primitives entre elles. Ces textures peuvent même présenter une certaine périodicité ou cyclo stationnarité (processus aléatoire plaqué sur un processus périodique). Certains exemples sont représentatifs de ce type de textures, comme la texture d’un mur de brique, de certains tissus ou d’un grillage [6].



Rotin

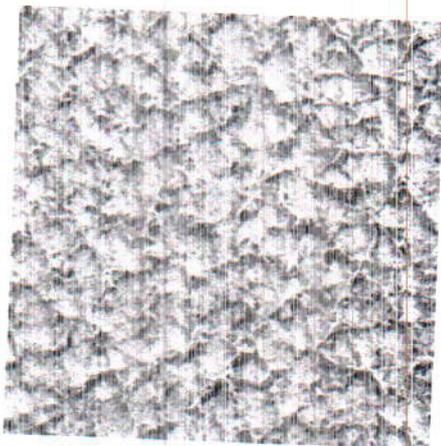


Carrelage

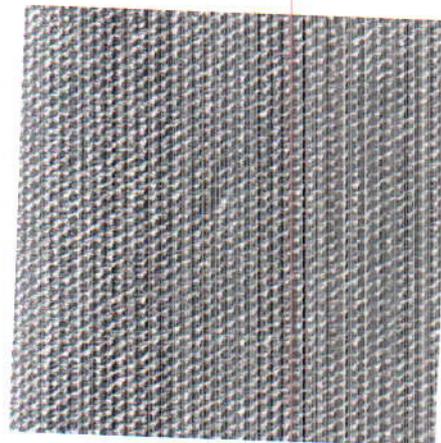
Figure I.3. Exemples de macro textures.

I.2.3.2. Micro textures

(Ou textures aléatoires) qui présentent un aspect plus chaotique. Et plus désorganisé, mais dont l'impression visuelle reste globalement homogène. Les différentes régions d'une image aérienne, les bois, les champs, etc., représentent des textures microscopiques [7].



Peau de cheval



Tresse

Figure I.4. Exemples de micro textures.

Mais, il est parfois difficile de classer une texture dans l'une ou l'autre des catégories (textures mixtes) d'autant que la résolution de l'image pose un problème supplémentaire, comme en témoigne la figure (I.5) ou du sable est photographié à 2 résolutions différentes : la première image laisse à penser que le sable est une micro texture, la seconde, qu'il est plutôt une macro texture.

Ce qui paraît évident et qui est admis par tous, c'est que la notion de texture est intimement liée à la perception visuelle, notamment des régions et des surfaces de l'image, et qu'elle se manifeste par une information visuelle fortement redondante. Ceci est traduit par la propriété essentielle des textures qui est leur invariance par translation. Leur observation laisse la même impression visuelle quelle que soit la partie de la texture observée.

Cette idée a été formalisée par la définition ci-dessous :

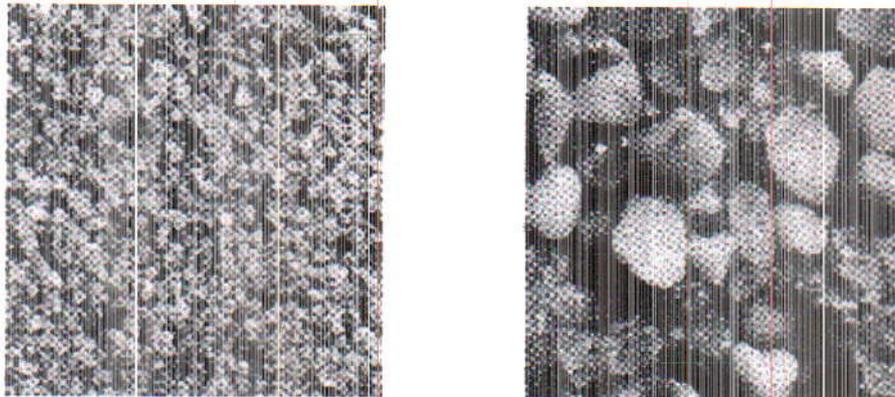


Figure I.5. Exemples de sables à 2 résolutions différentes.

La texture est décrite dans la littérature par deux caractéristiques à savoir les contours (information monodimensionnelle) qui séparent les régions homogènes et la surface (information bidimensionnelle) qui définit ces régions homogènes. Cependant, cette description peut être erronée dès qu'un changement de résolution de l'image survient, ce qui entraîne la modification de l'échelle d'observation.

I.2.4. Propriétés de perception des textures

Dans [8], l'auteur décrit deux propriétés qui sont essentielles dans la façon de percevoir une texture : la régionalité et la résolution.

La régionalité correspond au voisinage autour d'un pixel/voxel. La figure I.6 présente plusieurs régionalités d'une même texture avec tout d'abord en (a) une régionalité de 1. Dans cet exemple, nous ne pouvons pas encore parler de textures puisque la région est décrite par un unique pixel. En effet, la notion de textures apparaît dès lors où une région est composée d'un pixel/voxel et d'un voisinage non nul qui lui est associée. La figure (I.6) (b) présente un exemple

de régionalité égale à 3^2 . Si nous la comparons avec la figure I.6 (c) (régionalité égal à 9^2), celle-ci ne contient qu'une partie des motifs présents dans la texture. L'information que l'on peut obtenir sur une texture dépend donc très fortement de la taille du voisinage que l'on considère. Il est donc difficile de caractériser correctement une texture si la régionalité n'est pas choisie de façon adéquate.

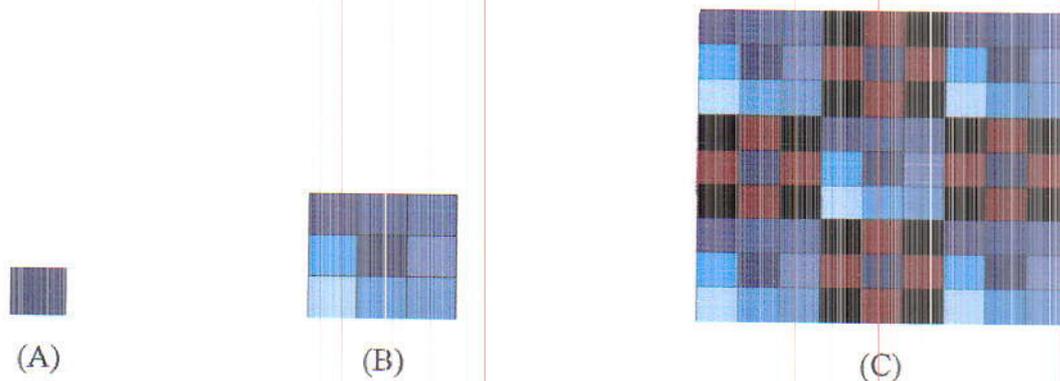


Figure I.6. Information sur une texture selon plusieurs régionalités

La résolution est également primordiale en analyse de textures dans le sens où une texture peut apparaître comme différente selon le niveau de précision auquel celle-ci est observée. A forte résolution, nous caractérisons ce qui constitue les primitives de bases d'une texture (perle, granulé de sable, ...). A faible résolution, la texture peut apparaître comme une juxtaposition des motifs de base plus grossiers. Selon l'échelle d'observation, on aura des caractéristiques différentes mais complémentaires.

Ce principe est notamment utilisé par les ondelettes ou les fractales multi résolutions. La figure (I.7) illustre bien ce phénomène. En nous éloignant, l'information concernant la texture aux résolutions hautes est perdue au profit des informations textures basses résolutions.

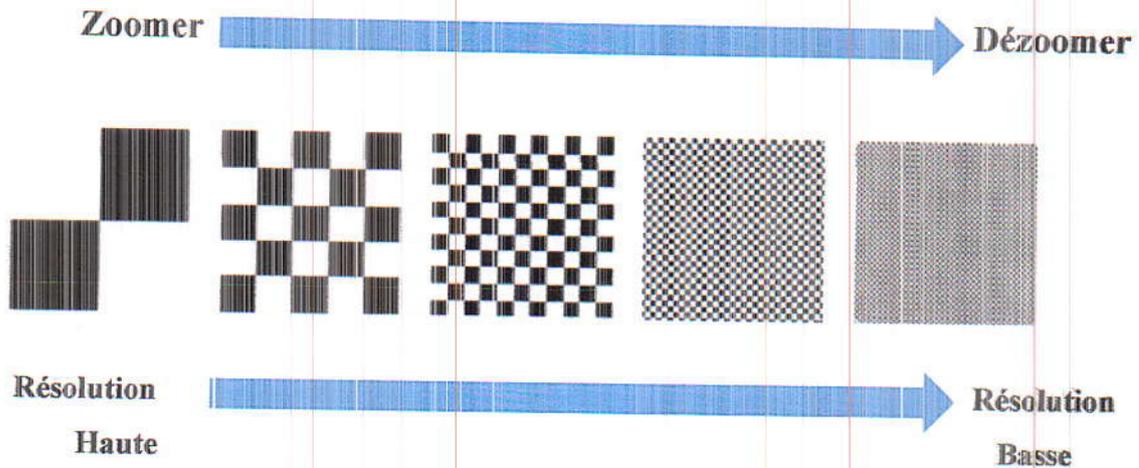


Figure I.7. Représentation d'une texture selon plusieurs résolutions

I.3. Méthodes d'extraction de caractéristiques de textures

Le but de l'analyse des textures est de décrire ces textures de façon formelle, en utilisant une formation mathématique. Une multitude de méthodes, de variantes et de combinaisons de méthodes ont déjà été proposées dans la littérature et éprouvées en pratique.

Les paramètres calculés permettent donc de caractériser et d'analyser les images en question afin de les segmenter. Il existe quatre classes connues d'attributs de texture [1, 9]. Notons qu'un attribut peut satisfaire plusieurs propriétés (figure I.8). Nous détaillons dans la suite de ce chapitre, chaque catégorie d'attributs.

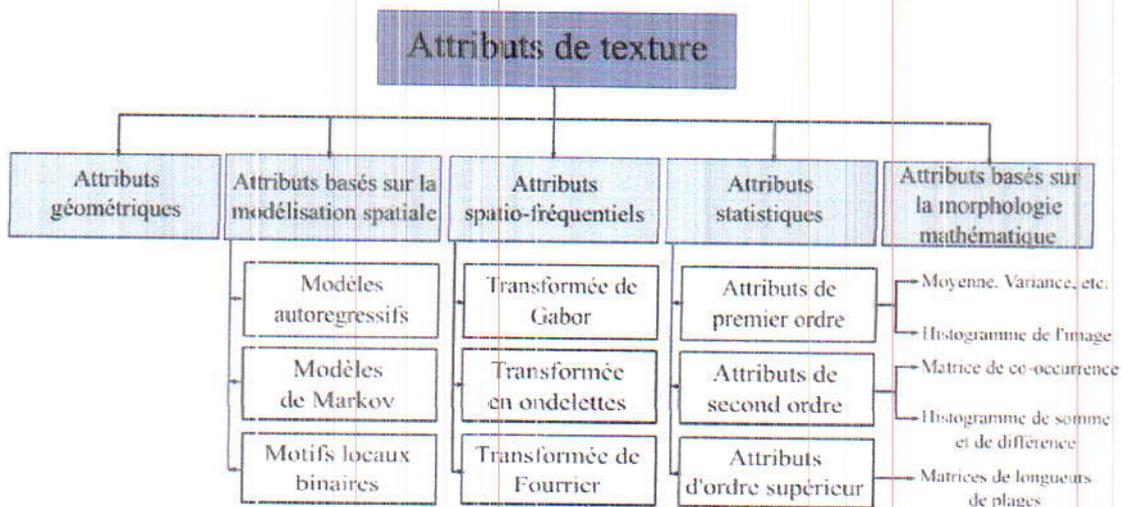


Figure I.8. Les différentes classes d'attributs de texture.

I.3.1. Attributs géométriques

La famille d'attributs géométriques est généralement adaptée à la perception macroscopique de la texture. Le calcul de ce type d'attributs commence par chercher des motifs de base (primitifs) de la texture et identifie ensuite des règles de placement de ces primitifs. En effet, les attributs géométriques exigent la régularité de la texture étudiée. Il est à noter que cette famille d'attributs est très peu utilisée vu qu'elle ne prend pas en compte les irrégularités de la texture fortement présentes dans les images naturelles [10].

En ce qui concerne leur utilisation en analyse de textures, l'approche la plus souvent utilisée consiste à binariser l'image originale de texture, à l'aide de seuils différents et de calculer les attributs géométriques sur les régions ainsi créées.

I.3.2. Attributs spatio-fréquentiels

Les attributs spatio-fréquentiels se divisent en trois catégories : ceux qui se calculent dans le domaine spatial, dans le domaine fréquentiel ou bien dans le domaine spatio fréquentiel. Nous détaillons dans ce qui suit chaque catégorie :

I.3.2.1. Dans le domaine spatial

Le calcul d'attributs de texture dans le domaine spatial consiste à caractériser la texture par le nombre de transitions par unité de surface. Par conséquent, contrairement aux textures fines, les textures grossières tendent à avoir une faible densité de transition [9].

Cependant, ce type d'attributs s'utilise beaucoup plus dans les problématiques de segmentation d'image.

I.3.2.2. Dans le domaine fréquentiel

Le calcul d'attributs de texture dans le domaine fréquentiel consiste à effectuer une transformation du domaine spatial au domaine fréquentiel. Les transformations les plus répandues sont effectuées par une transformée de Fourier et une transformée en cosinus. Celles-ci sont basées sur l'utilisation d'un ensemble de filtres sensibles chacun à une fréquence particulière. À partir de l'image résultante (image filtrée), les attributs de texture sont calculés en utilisant des outils statistiques à savoir l'énergie et la variance. Il est à noter que la transformée de fourrier et la transformée en cosinus sont souvent mieux adaptées pour

des images ayant des zones homogènes (texture grossière). Par conséquent, l'intégralité de l'information est représentée par les basses fréquences tandis que les hautes fréquences sont réservées à des changements locaux.

I.3.2.3. Dans le domaine spatio-fréquentiel

Un troisième type d'attributs de texture associant les deux domaines spatial et fréquentiel existe. Parmi les transformations issues de cette catégorie et communément utilisées, nous pouvons citer la transformée de Gabor [11] et la transformée d'ondelettes [12].

I.3.3. Attributs statistiques

Les attributs statistiques traduisent la relation (en termes de variation de couleur) entre un pixel et ses voisins en tenant compte d'un système de voisinage défini a priori. Les attributs statistiques peuvent être divisés en trois ensembles à savoir les attributs statistiques de premier ordre tel que seulement le pixel en question est pris en compte (les histogrammes), les attributs statistiques de deuxième ordre (les matrices de co-occurrences et les histogrammes de sommes et de différences) et les attributs statistiques d'ordre supérieur (les matrices de longueurs de plages). Notons que les attributs statistiques sont très utilisés grâce à la pertinence de leurs outils pour caractériser même les textures fines qui ne présentent pas de régularités apparentes. Nous rappelons, dans la suite, les attributs statistiques de texture, de chaque ensemble, les plus communément utilisés.

I.3.3.1. Attributs statistiques de premier ordre

Les statistiques du premier ordre mesurent la probabilité d'observer un niveau de gris à un emplacement aléatoire d'une image. Elles sont calculées à partir de l'histogramme des intensités. Parmi les attributs du premier ordre nous retrouvons :

- Les moments d'ordre k .
- Les moments centrés d'ordre k avec : La moyenne des niveaux de gris, La variance, Le biais, L'aplatissement, L'énergie, L'entropie, Le contraste, La dynamique, Le coefficient de variation, L'exposant de Holder [13].

Pour améliorer la qualité des statistiques, il est conseillé de requantifier l'histogramme des intensités. En effet, le fait de considérer 256 niveaux de gris réduit le nombre d'occurrences

moyen par pixel. Les mesures effectuées dans un voisinage restreint, risquent de devenir peu significatives.

I.3.3.2. Attributs statistiques de second ordre

Les deux attributs statistiques de second ordre les plus répandus sont les matrices de co-occurrences et les histogrammes de somme et de différence. Les matrices de co-occurrences, proposées par [14], constituent un outil performant pour la description des interactions spatiales entre les pixels. En effet, cet outil permet d'étudier la distribution du niveau de gris [15], tout en tenant compte de la relation spatiale entre les pixels.

I.3.4. Matrices de co-occurrences

Les matrices de co-occurrence constituent une représentation simple des relations spatiales inter pixels. Elles permettent d'estimer les statistiques d'ordre 2 de l'image. Ces statistiques sont accumulées en un ensemble de matrices 2D dont chacune mesure la dépendance spatiale de deux niveaux de gris, étant donné un vecteur de déplacement. On peut dériver de la matrice de co-occurrence des caractéristiques de la texture, comme l'énergie, l'entropie, le contraste, l'homogénéité et la corrélation. Elles ont été introduites en 1973 [16].

La matrice de co-occurrence permet de déterminer la fréquence d'apparition des paires de valeurs de pixels situés à une certaine distance dans l'image. Elle est basée sur le calcul de la probabilité $P(N_i, N_j, d, \theta)$ qui représente le nombre de fois où un pixel de niveau de gris N_i apparaît à une distance relative (d) d'un pixel de niveau de gris N_j et selon l'orientation de la droite reliant ces deux pixels par rapport à l'horizontal (θ). On prend généralement pour valeur $d = 1$ et $\theta = 0^\circ, 45^\circ, 90^\circ$ et 135° . Les relations de voisinage entre pixels par rapport à l'orientation, nécessaires au calcul des matrices de co-occurrence, sont illustrées en Figure (I.9).

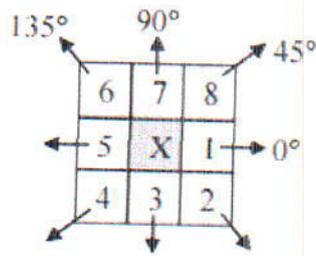


Figure I.9. Huit pixels plus proches voisins utilisés pour calculer la matrice de cooccurrence. Les plus proches voisins de 'x' selon la direction $\theta = 0^\circ$ sont les pixels 1 et 5, pour $\theta = 45^\circ$ les pixels 4 et 8, pour $\theta = 90^\circ$ les pixels 3 et 7 et enfin pour $\theta = 135^\circ$ les pixels 2 et 6.

Les matrices obtenues selon ces quatre directions sont alors calculées à l'aide des équations : (1.1), (1.2), (1.3) et (1.4) où (k, l) sont les coordonnées d'un pixel de niveau de gris $N_i \in [0, 2^P - 1]$, (m, n) celles du pixel de niveau de gris $N_j \in [0, 2^P - 1]$ et n le nombre de bits [17].

$$P(N_i, N_j, d, 0^\circ) = \left| \begin{array}{l} (k, l), (m, n) \in (N, M)^2 \text{ tq } (k - m = 0, |l - n| = d), \\ I_{k,l} = N_i, I_{m,n} = N_j \end{array} \right| \quad (1.1)$$

$$P(N_i, N_j, d, 45^\circ) = \left| \begin{array}{l} (k, l), (m, n) \in (N, M)^2 \text{ tq} \\ (k - m = d, l - n = -d) \wedge (k - m = -d, l - n = d) \\ I_{k,l} = N_i, I_{m,n} = N_j \end{array} \right| \quad (1.2)$$

$$P(N_i, N_j, d, 90^\circ) = \left| \begin{array}{l} (k, l), (m, n) \in (N, M)^2 \text{ tq } (|k - m| = d, l - n = 0), \\ I_{k,l} = N_i, I_{m,n} = N_j \end{array} \right| \quad (1.3)$$

$$P(N_i, N_j, d, 135^\circ) = \left| \begin{array}{l} (k, l), (m, n) \in (N, M)^2 \text{ tq} \\ (k - m = d, l - n = d) \wedge (k - m = -d, l - n = -d), \\ I_{k,l} = N_i, I_{m,n} = N_j \end{array} \right| \quad (1.4)$$

I.3.4.1. Attributs d'Haralick extraits des matrices de co-occurrences

Afin d'extraire des attributs de texture des matrices de co-occurrence, quatorze indices ont été définis par [18]. Ci-dessous les 6 paramètres considérés comme étant les plus utilisés et les plus pertinents [19] :

- **l'énergie : ENE**, mesure la monotonie de la texture. Elle atteint de fortes valeurs lorsque la distribution des niveaux de gris est constante ou de forme périodique. Dans ce cas, le déplacement correspond alors à la période.

$$ENE = \sum_{N_i} \sum_{N_j} (P_{N_i, N_j}(d, \theta))^2 \tag{I.5}$$

- **le contraste : CST**, sa valeur est d'autant plus élevée que la texture présente des Passages d'un pixel très clair à un autre très foncé dans l'image (fort contraste). Ce Paramètre est fortement non corrélé à l'énergie.

$$CST = \sum_{N_i} \sum_{N_j} [(N_i - N_j)^2 \cdot P_{N_i, N_j}(d, \theta)] \tag{I.6}$$

- **l'entropie : ENT**, mesure le désordre dans l'image. Contrairement à l'énergie, l'entropie atteint de fortes valeurs pour les textures complètement aléatoires.

$$ENT = - \sum_{N_i} \sum_{N_j} [P_{N_i, N_j}(d, \theta) \cdot \log_2 (P_{N_i, N_j}(d, \theta))] \tag{I.7}$$

- **la variance : VAR**, la variance mesure l'hétérogénéité de la texture. Sa valeur augmente lorsque les niveaux de gris diffèrent par rapport à leur moyenne. Ce paramètre est indépendant du contraste.

$$VAR = \sum_{N_i} \sum_{N_j} [(N_i - \mu)^2 \cdot P_{N_i, N_j}(d, \theta)] \tag{I.8}$$

- **la corrélation : COR**, ce paramètre mesure la dépendance linéaire relative au déplacement "d" des niveaux de gris dans l'image. Ce paramètre n'est corrélé ni à l'énergie, ni à l'entropie.

$$COR = \sum_{N_i} \sum_{N_j} \left[\frac{(N_i - \mu)(N_j - \mu) \cdot (P_{N_i, N_j}(d, \theta))}{\sigma^2} \right] \quad (I.9)$$

- le moment de différence inverse : *IDM* (pour Inverse Difference Moment), à l'inverse de la variance ce paramètre, mesure l'homogénéité de la texture de l'image. Il est corrélé à une combinaison linéaire des paramètres *ENE* et *CST*.

$$IDM = \sum_{N_i} \sum_{N_j} \left[\frac{(P_{N_i, N_j}(d, \theta))}{1 + (N_i - N_j)^2} \right] \quad (I.10)$$

Tous ces attributs sont définis pour une valeur de déplacement "*d*", donc, le choix de ce paramètre est déterminant pour l'obtention d'un résultat de qualité, cependant, il n'existe pas de règles pour choisir de bonnes valeurs de déplacement. Récemment, de nombreuses études ont été menées afin de caractériser et classifier des textures osseuses à l'aide de ces paramètres [20-23].

I.3.5. Attributs basés sur la modélisation spatiale

Cette catégorie d'attributs dépend essentiellement de la répartition spatiale des couleurs dans l'image. Par conséquent, le calcul d'attributs est effectué en modélisant cette répartition spatiale. Parmi les modèles les plus répandus, nous pouvons citer deux modèles suivants [24] :

I.3.5.1. Les modèles de MARKOV

Les modèles de MARKOV constituent un cas particulier des modèles auto régressifs. L'intérêt de ces modèles (par rapport aux modèles autorégressifs) réside dans leur stratégie de modélisation des interactions spatiales entre un pixel avec ses voisins ce qui nécessite la définition d'un système de voisinage. En effet, pour définir les champs de Markov, l'image est modélisée en un ensemble fini *S* de sites *s_i* correspondant aux pixels. À chaque site, un descripteur est associé, ce qui permet de décrire l'état du site. Le descripteur peut être le niveau de gris ou une étiquette, mais il peut être une information plus complexe [25].

I.3.5.2. Méthode des motifs binaires (LBP)

Le principe de cette méthode est de comparer le niveau de luminance d'un pixel avec les niveaux de ses voisins. Cela rend donc compte d'une information relative à des motifs réguliers

dans l'image, autrement dit une texture. Selon l'échelle du voisinage utilisée, certaines zones d'intérêt tel des coins ou des bords peuvent être détectées par ce descripteur.

L'opérateur de motif binaire local [26] est un moyen efficace de description de texture. Nous détaillons la méthode des LBP dans ce qui suit.

I.4. Les motifs binaires locaux (LBP)

I.4.1. Définition

Les motifs binaires locaux (local binary patterns en anglais) sont des caractéristiques utilisées en vision par ordinateur pour reconnaître des textures ou pour la détection d'objet dans des images numériques.

Ce descripteur a été mentionné pour la première fois en 1993 pour mesurer le contraste local d'une image mais réellement popularisé trois ans plus tard par Ojala en 1996, afin de caractériser les textures présentes dans des images en niveaux de gris. Ils consistent à attribuer à chaque pixel P de l'image $I(i, j)$ à analyser, une valeur caractérisant le motif local autour de ce pixel. Ces valeurs sont calculées en comparant le niveau de gris du pixel central P aux valeurs des niveaux de gris des pixels voisins [27].

I.4.2. Principe général

Le concept du LBP est simple, il propose d'assigner un code binaire à un pixel en fonction de son voisinage. Ce code décrivant la texture locale d'une région est calculé par seuillage d'un voisinage avec le niveau de gris du pixel central. Afin de générer un motif binaire, tous les voisins prendront alors une valeur "1" si leur valeur est supérieure ou égale au pixel courant et "0" autrement (Figure I.11). Les pixels de ce motif binaire sont alors multipliés par des poids et sommés afin d'obtenir un code LBP du pixel courant. On obtient donc pour toute l'image, des pixels dont l'intensité se situe entre 0 et 255 comme dans une image à 8 bits ordinaire. Plutôt que de décrire l'image par la séquence des motifs LBP, on peut choisir comme descripteur de texture un histogramme de dimension 255.

Un voisinage pour un pixel central est réparti sur un cercle et construit à partir de deux paramètres : le nombre de voisins sur le cercle et un rayon pour définir une distance entre un pixel central et ses voisins. Définissons une texture :

$$T = t(g_c, g_0, \dots, g_{P-1}) \tag{I.11}$$

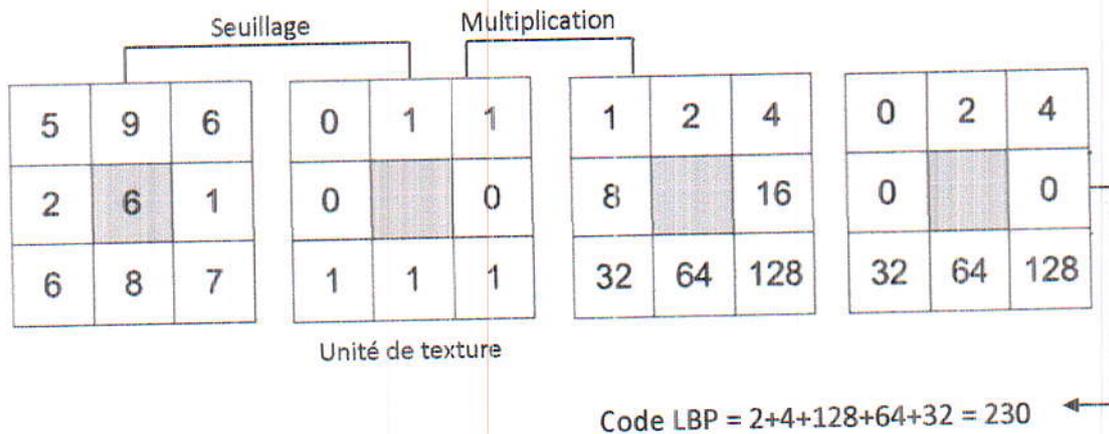


Figure I.10. Construction d'un motif binaire et calcul du code LBP.

g_c correspond à la valeur de niveau de gris du pixel central et g_p , avec $p = 0, P - 1$, correspond au niveau de gris de P pixels espacés régulièrement sur un cercle de rayon R . Si les coordonnées de g_c sont égales à $(0,0)$, alors les coordonnées de g_p sont données par :

$$(-R \sin(2\pi p/P) \quad -R \cos(2\pi p/P)) \tag{I.12}$$

Comme nous pouvons le voir à la figure I.9 les coordonnées d'un voisin ne sont pas forcément situées au centre d'un pixel. Dans ce cas, le niveau de gris est calculé par l'intermédiaire d'une interpolation bilinéaire [23].

Voici une représentation graphique de T faisant varier les paramètres P et R :

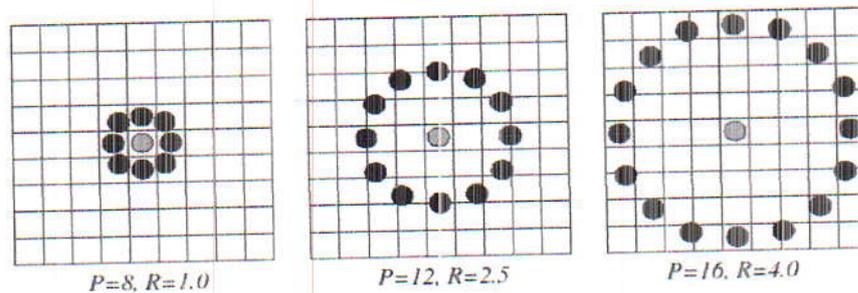


Figure I.11. Représentation du pixel central g_c et de son voisinage g_p

Par la suite, g_c est utilisé comme un seuil de la façon suivante :

$$T = t(s(g_0 - g_c), \dots, s(g_{P-1} - g_c)) \tag{I.13}$$

Avec $s(x)$ égal à 1 si x positif et $s(x)$ égal à 0 si x est strictement négatif.

En multipliant chaque fonction $s(g_p - g_c)$ par un poids de valeur 2^p , on obtient alors un code binaire tel que :

$$LBP_{p,R}(xc, yc) = \sum_{p=0}^{p-1} s(g_p - g_c) 2^p \tag{I.14}$$

I.4.3. LBP uniforme et invariant par rotation

Un motif binaire uniforme est défini, comme étant tout motif possédant exactement 0 ou 2 transitions (01 ou 10) dans un parcours circulaire comme le montre l'exemple de la Figure (I.13) (par exemple 10000001 ou 00011000 est un motif uniforme, mais pas 00101010).

Cette notion d'uniformité est importante dans la méthode LBP pour représenter les informations de primitives structurelles comme les arêtes (coins) et les contours.

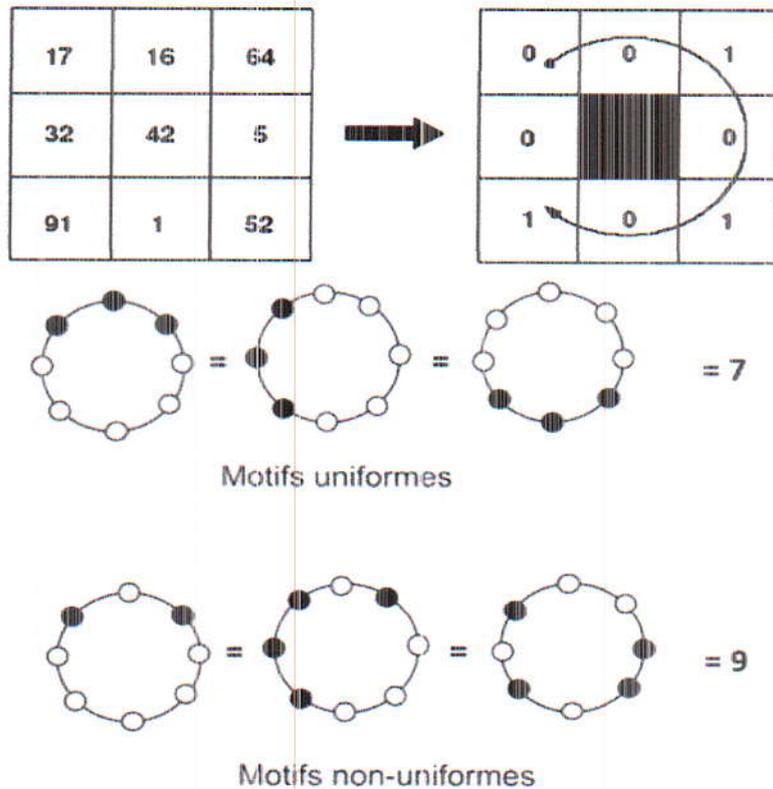


Figure I.12. Exemples de construction et uniformité d'un motif LBP, et motifs uniformes et non-uniformes.

Pour supprimer l'effet de rotation, un code LBP ($LBP_{P,R}^{ri}$) est défini en vue de retourner une valeur unique quelle que soit la rotation appliquée :

$$LBP_{P,R}^{ri} = \min \{ROR(LBP_{P,R}, i)\} \tag{I.15}$$

Avec $i = 0, 1, \dots, P - 1$, et $ROR(x, i)$ permettant de faire une rotation bit à bit de x et i fois. Avec cette méthode, 36 motifs différents sont considérés au lieu de 2^P .

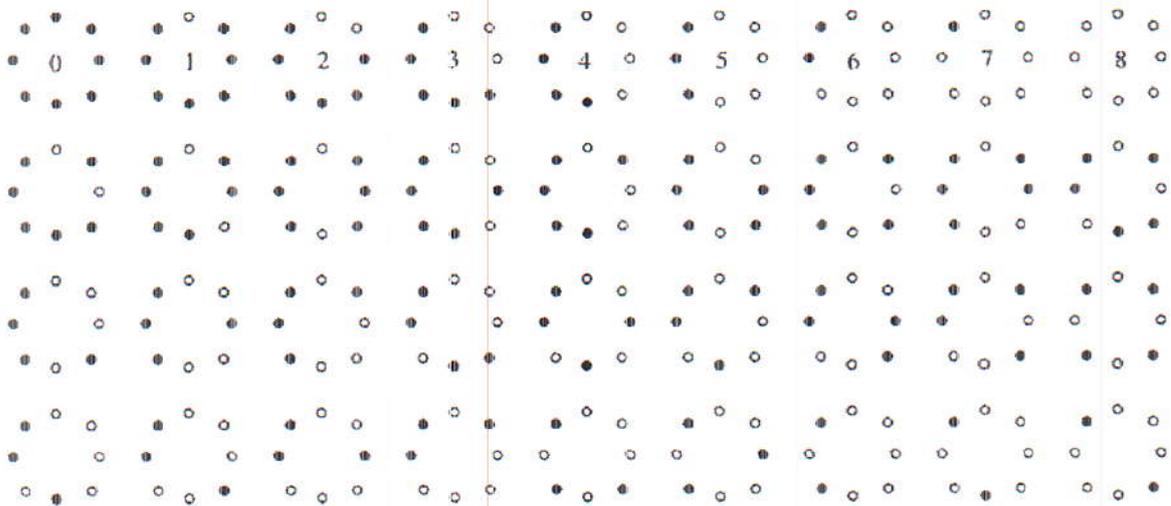


Figure I.13. Liste des 36 uniques motifs binaires locaux invariant en rotation possibles pour $LBP_{8,R}^{ri}$. Les points noirs et blancs correspondent respectivement aux valeurs de bit 0 et 1 dans la sortie sur 8 bits de l'opérateur $ROR(.)$ [29].

$R=1$, $LBP_{P,R}^{ri}$ ne permet pas d'obtenir de bonnes performances [30]. La principale raison est que l'occurrence des fréquences de certains des 36 motifs possibles varie fortement (instabilité). Pour résoudre ce problème, une mesure est effectuée sur les motifs afin de déterminer s'ils sont uniformes. Pour cela une fonction U retourne le nombre de transitions spatiales (passage de 0 à 1 ou de 1 à 0). Si le nombre de transitions est supérieur à deux, alors le motif est considéré comme non-uniforme. L'opérateur appelé $LBP_{P,R}^{riu2}$ est défini de la façon suivante :

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{p-1} s(g_p - g_c) & \text{Si } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{Sinon} \end{cases} \quad (I.16)$$

Avec

$$U(LBP_{P,R}) = |s(g_{p-1} - g_c) - s(g_0 - g_c)| + \sum_{p=0}^{p-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (I.17)$$

Une mesure d'uniformité.

Concernant les motifs uniformes, l'opérateur $LBP_{P,R}^{riu2}$ correspond à la somme des bits du voisinage. Au contraire si les motifs sont non-uniformes, l'opérateur est affecté à la valeur $P + 1$ (Figure I.13). Le nombre d'apparitions de chaque code est accumulé dans un histogramme de taille $P + 2$. Chaque tranche de cet histogramme donne une estimation de la probabilité d'apparition d'un motif dans une texture. Les textures sont caractérisées en utilisant directement l'histogramme des codes binaires.

Dans [31], les auteurs proposent une variation de cette méthode en coordonnant une invariance à la rotation à la fois globale et locale. Dans un premier temps, ils utilisent la distribution LBP pour déterminer l'orientation principale d'une texture et l'utilisent afin d'aligner l'histogramme LBP. Dans la figure (I.12), le motif uniforme obtient le même code LBP quelle que soit son orientation. Le fait d'aligner l'histogramme LBP selon l'orientation principale (information globale) permet de différencier ces 3 motifs uniformes en leur affectant des codes binaires différents.

I.5. Conclusion

L'analyse de texture fait état d'un grand nombre d'approche et de méthodes existantes dans la littérature. Il est témoin d'un besoin croissant de connaissance plus approfondie sur la notion même de texture et de mécanisme humaine de perception des textures. Ce chapitre donne une description de la notion de texture, et les différentes méthodes utilisées pour l'analyse de texture. Nous avons présenté la méthode LBP qu'est basée sur l'extraction des attributs spatiaux. Ou chaque pixel reçoit un code binaire qui dépend des valeurs des pixels voisins. Cette méthode LBP sert à classer les images texturées.

CHAPITRE II

La classification

Des données

II.1. Introduction

La reconnaissance de classe d'objets est l'un des problèmes les plus difficiles en reconnaissance de forme. Les approches les plus efficaces utilisent une description locale des images et des méthodes statistiques de classification pour l'étape de reconnaissance. En général la reconnaissance de forme est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique et aux statistiques.

II.2. Définition de la classification

L'objectif visé dans notre cas par la classification est de pouvoir distinguer entre des classes d'images correspondantes à des textures différentes dans une série d'images. Cette méthode est basée sur l'apprentissage suivi de la classification. La phase d'apprentissage correspond à l'extraction d'attributs caractéristiques à partir de l'image dans la base d'apprentissage. La classification c'est la phase au cours de laquelle sont utilisés les attributs précédemment extraits afin d'atteindre l'objectif initial.

La classification, en général, se réfère au classement ou groupement d'éléments de données dans des ensembles similaires. La classification a deux significations distinctes. On peut recevoir une série d'observations avec l'objectif d'établir l'existence des classes ou des groupes dans les données. On peut savoir avec certitude qu'il y'a tant de classes, et l'objectif consiste à instaurer une règle selon laquelle nous pouvons classer une nouvelle observation dans l'une des classes existantes. Le premier type est connu comme l'apprentissage non supervisé et le second comme l'apprentissage supervisé.

La classification d'image est en général similaire à la classification des données, mais elle peut être différente en fonction de l'application dans laquelle elle est utilisée. La classification est souvent la dernière étape d'un processus de diagnostic général (Figure II.1). Il s'agit généralement d'un tri d'objets dans une image ou plusieurs images dans des classes distinctes.

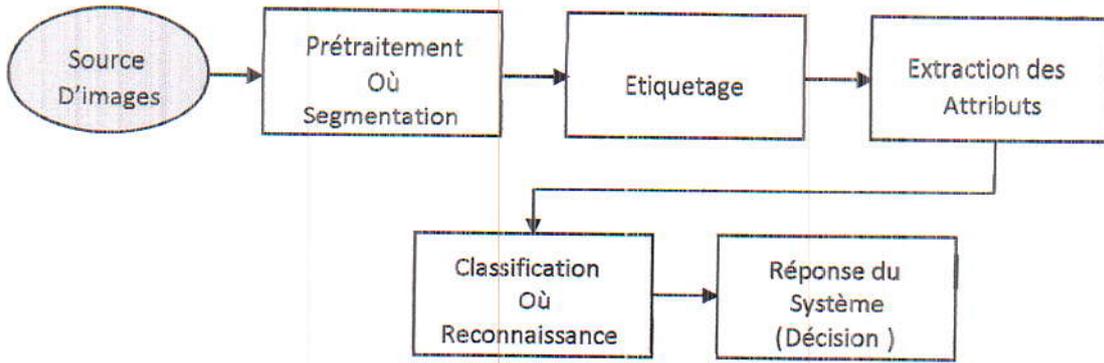


Figure II.1. Schéma de principe d'un système de classification.

II.3. Les types de la classification

Les algorithmes de classification peuvent être répertoriés de plusieurs manières (voir figure II.2). Un premier critère et le niveau de supervision, c'est-à-dire l'importance de l'intervention de l'utilisateur dans le processus de classification.

Il est également possible de séparer les algorithmes paramétriques associés à un modèle probabiliste des algorithmes non paramétriques [32].

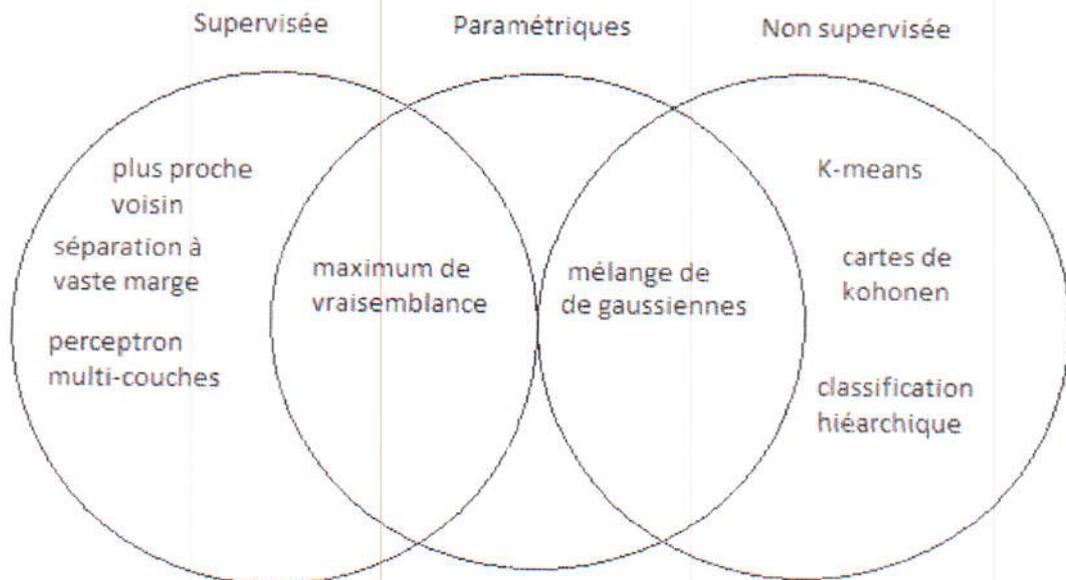


Figure II.2 Classification des types de la classification [31].

II.3.1. Classification non supervisée

Cette méthode est aussi appelée classification automatique (clustering en anglais) ou encore "regroupement". Aucune information a priori sur la texture n'est connue. On cherche alors à regrouper les différents exemples de textures à traiter en fonction de la valeur de leurs descripteurs de manière à créer des classes homogènes. On suppose qu'on dispose d'un ensemble d'objets que l'on note par $X = \{x_1, x_2, \dots, x_N\}$ caractérisé par un ensemble de descripteurs "D", l'objectif du regroupement est de trouver les groupes auxquels appartiennent chaque objet "x" qu'on note par $C = \{c_1, c_2, \dots, c_N\}$. Ce qui revient à déterminer une fonction notée "Ys" qui associe à chaque élément de X un ou plusieurs éléments de C. Il faut pouvoir affecter une nouvelle observation à une classe. Les observations disponibles ne sont pas initialement identifiées comme appartenant à telle ou telle population. L'absence d'étiquette de classe est un lourd handicap qui n'est que très partiellement surmontable. Ce procédé nécessite généralement de fixer au préalable le nombre de classes désirées, que ce soit de manière empirique ou automatique [33, 34]. Parmi les méthodes non-supervisées les plus utilisées, citons deux types d'approches : les centres mobiles (*k*-means) et la classification hiérarchique.

II.3.2. Classification supervisée

Dans cette méthode de classification, on dispose déjà d'exemples dont la classe est connue et étiquetée. Une information sur les données à traiter est disponible et est utilisée pour entraîner le processus de classification, cela constitue la phase d'apprentissage du modèle. Cette information appelée ensemble d'apprentissage est généralement constituée d'un ensemble d'individus {caractéristiques, classe associée}. Dans le cas de la classification de textures, l'ensemble d'apprentissage est constitué d'un ensemble de textures "types". Chaque individu est donc composé du couple (caractéristiques de textures, texture associée). Cet ensemble est alors appris par un algorithme de classification supervisée classique parmi lesquels, on cite : les *k*-plus proches voisins (*k*-NN), les réseaux de neurones, les machines à support de vecteurs (SVM), etc. Une fois la phase d'apprentissage réalisée, l'algorithme de classification est alors utilisé afin de déterminer la classification d'un ensemble d'individus tests composé d'un grand nombre d'échantillons [33].

Le schéma présenté sur la figure (II.3) illustre de manière générale la méthode de classification supervisée.

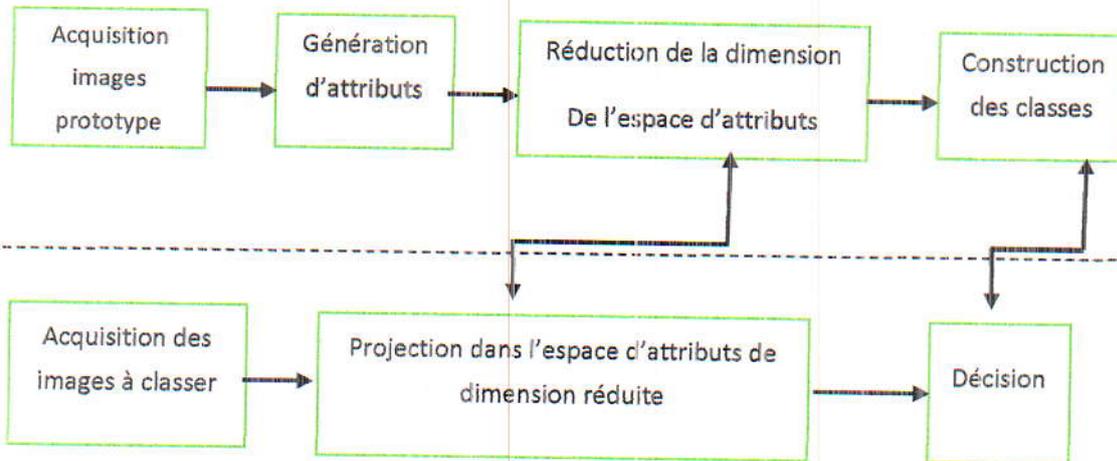
ApprentissageClassification

Figure II.3. Etapes de la classification supervisée.

Cette méthode est divisée en deux phases successives [35] :

- L'apprentissage, a pour but de construire des classes à partir de l'ensemble des prototypes (également appelé ensemble d'apprentissage) [36]. Dans notre cas, les textures présentes dans les images prototypes sont décrites par un ensemble d'attributs. Chaque texture est alors représentée par un vecteur d'attributs dans un espace d'attributs. Un classifieur est ensuite élaboré afin de déterminer la classe d'appartenance d'une image en fonction de ses caractéristiques.
- La classification, consiste à appliquer les règles de classification déterminées lors de l'apprentissage afin de classer l'image de la texture à contrôler.

II.4. Méthodes de classification

Une fois la sélection des caractéristiques effectuée, nous pouvons procéder à la classification des images texturées, Les méthodes utilisées pour la classification sont nombreuses, citons : la méthode des Machines à vecteurs supports (SVM), k plus proches voisins (K-NN).

II.4.1. Séparateurs à vaste marge (SVM)

Cette méthode de classification de données a été proposée par [37]. Initialement, elle a été définie pour un problème de discrimination à deux classes (cas binaire). De manière simple, il s'agit de construire, dans l'espace des attributs un hyperplan séparant les points représentatifs des prototypes de chacune des deux classes. Cette approche, également appelée méthode des séparateurs à vastes marges, repose sur deux notions clés : la notion de marge maximale et la notion de noyau [38].

La marge est la distance séparant la frontière de séparation (l'hyperplan) et les prototypes les plus proches de cette frontière. Ces prototypes sont appelés vecteurs supports.

Dans les SVM, la frontière de séparation est choisie comme étant celle qui maximise la marge (voir la Figure II.4) [38].

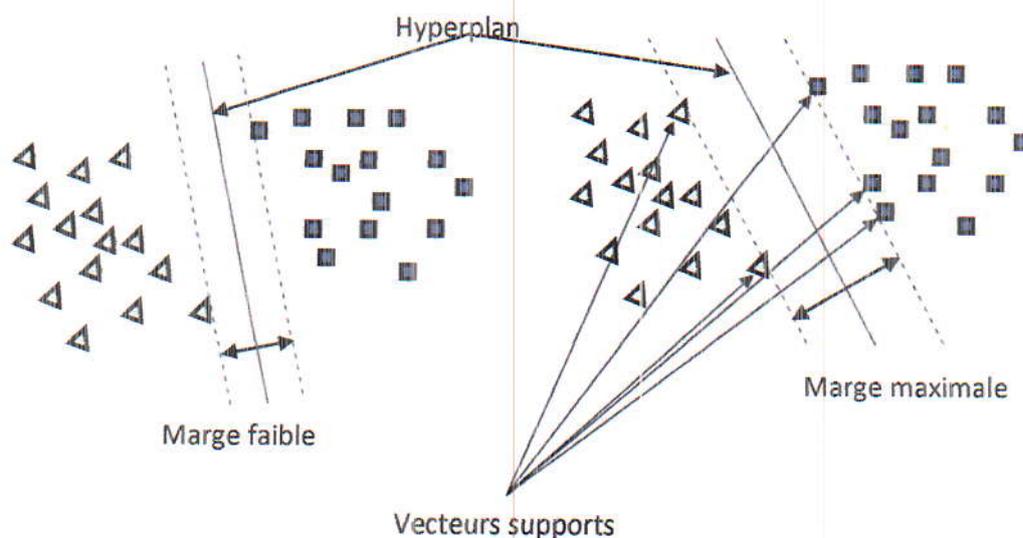


Figure II.4. Illustration de la marge maximale et les vecteurs supports qui sont lui associés.

Parmi les modèles des SVM, il y a les cas linéairement séparable (Figure II.4) et les cas non linéairement séparable. Les premiers sont les plus simples car ils permettent de trouver facilement le classifieur linéaire. Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace.

Il s'agit d'une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : Polynomiale, Gaussien, Sigmoidé et Laplacien.

II.4.2. K-plus proche voisins (k-NN)

L'algorithme des k plus proches voisins (noté k-NN) fait partie des méthodes de classification les plus couramment utilisées. Il permet, tout comme les SVM, de traiter des nuages de points non linéairement séparables. Cette approche a l'avantage d'être à la fois simple et efficace. L'algorithme k -NN figure parmi les algorithmes simples d'apprentissage artificiel. Dans un contexte de classification d'une nouvelle observation x , l'idée fondatrice est de faire voter les plus proches voisins de cette observation. La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x . La méthode k -NN est donc une méthode basée sur le voisinage, non-paramétrique ; ceci signifiant que l'algorithme permet de faire une classification sans faire d'hypothèse sur la fonction qui relie la variable dépendante aux variables indépendantes [39].

La méthode 1-NN : La méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche, vis-à-vis des co-variables utilisées. La détermination de leur similarité est basée sur des mesures de distance. Formellement, soit L l'ensemble de données à disposition ou échantillon d'apprentissage :

$$L = \{(y_i, x_i), i = 1, \dots, +n_L\} \quad (\text{II.1})$$

Où $y \in \{1, \dots, c\}$ désigne la classe de l'individu " i " et le vecteur $x_i = (x_{i1}, \dots, x_{ip})$ représente les variables prédictives de l'individu " i ". La détermination du plus proche voisin est basée sur une fonction distance notée d .

La distance euclidienne ou dis similarité entre deux individus caractérisés par " p " Co variables est définie par :

$$d((x_1, x_2, \dots, x_p), (u_1, u_2, \dots, u_p)) = \sqrt{(x_1 + u_1)^2 + (x_2 + u_2)^2 + \dots + (x_p + u_p)^2} \quad (\text{II.2})$$

Ainsi, pour une nouvelle observation (y, x) le plus proche voisin $(y_{(1)}, x_{(1)})$ dans l'échantillon d'apprentissage est déterminé par :

$$d(x, x_{(1)}) = \min_i (d(x, x_{(i)})) \quad (\text{II.3})$$

La méthode k-NN : la méthode des k plus proches voisins est une extension de l'idée précédente, qui est largement et communément utilisée en pratique. La plus proche observation n'est plus la seule observation utilisée pour la classification. Nous utilisons désormais les k plus proches observations. Ainsi la décision est en faveur de la classe majoritairement représentée par les k voisins. Soit k_A le nombre d'observations issues du groupe des plus proches voisins appartenant à la classe A .

$$\sum_{A=1}^c K_A = k \quad (\text{II.4})$$

Ainsi une nouvelle observation est prédite dans la classe l avec :

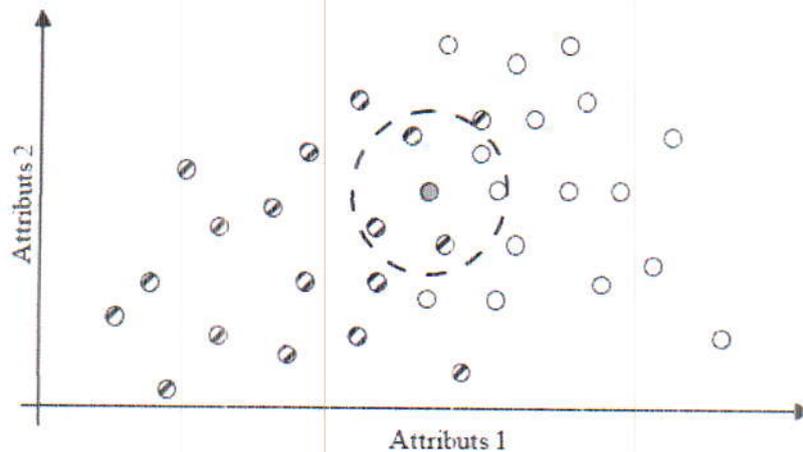
$$l = \max_A (K_A) \quad (\text{II.5})$$

Ceci évite que la classe prédite ne soit déterminée seulement à partir d'une seule observation. La mise en œuvre de cette technique ne dépend que du paramètre k : pour $k = 1$, on utilise la méthode du seul plus proche voisin comme technique locale maximale, pour $k = l_n$, nous utilisons la classe majoritaire sur l'ensemble intégral des observations.

Le paramètre k doit être déterminé par l'utilisateur : $k \in N$. En classification binaire, il est utile de choisir k impair pour éviter les votes égalitaires. Le meilleur choix de k dépend du jeu de données. En général, les grandes valeurs de k réduisent l'effet du bruit sur la classification et donc le risque de sur-apprentissage, mais rendent les frontières entre classes moins distinctes. [40, 41].

La figure (II.5) illustre cette méthode avec $k = 5$, pour des points image projetés dans un espace d'attributs de dimension $d = 2$ et pour un nombre de classe $N_C = 2$. Pour classer l'image test, dont le point représentatif dans le sous espace d'attributs est gris, on cherche tout d'abord

les 5 points prototypes les plus proches (au sens de la distance Euclidienne). Ces images sont celles présentes dans le cercle en pointillé centré sur le point représentatif de l'image à classer. Il y a 3 images appartenant à la classe 2 et 2 images appartenant à la classe 1. L'image test est donc assignée à la classe 2.



○ Point prototype appartenant à la classe 1 ● Point prototype appartenant à la classe 2
 ● Point image test

Figure II.5. Illustration de la méthode des k plus proches voisins (k -NN) avec $k = 5$.

II.3.5. Types d'informations fournies par les classificateurs

Les réponses fournies par les classifieurs peuvent être divisées en quatre types :

Type classe : la réponse du classifieur est une classe. Il considère que la forme d'entrée x appartient à cette classe.

Type ensemble : la réponse du classifieur est un ensemble de classes. Il considère que la forme x appartient à une des classes de cet ensemble, mais il ne me donne aucune indication supplémentaire qui pourrait nous aider à discriminer parmi ces classes.

Type rang : la réponse est un classement sur toutes les classes ou seulement sur une partie de classe. La classe mise au premier rang du classement est la classe qu'il considère comme la plus probable pour la x forme et la classe du dernier rang est la moins probable. Le classifieur indique ce classement en donnant comme sortie un vecteur des rangs.

Type Mesure : le classifieur donne, pour toutes les classes, des mesures qui indiquent en quelque sorte, directement ou indirectement à quel degré le classifieur considère chaque classe comme la vraie classe de x . la sortie du classifieur est donc un vecteur de mesures [42].

II.5. Evaluation de la classification

II.5.1. Base d'apprentissage et base de test

L'estimation des performances nécessite des exemples. Il est évident que ces exemples ne doivent pas être les mêmes que ceux sur lesquels l'apprentissage est effectué, sinon les performances estimées vont être surement trop optimistes puisque le classifieur a déjà appris ces exemples au cours de sa construction. Nous avons donc besoin de certains exemples utilisés pour l'apprentissage et d'autre pour le test.

L'ensemble des exemples utilisés pour l'apprentissage est nommé base d'apprentissage. L'ensemble des exemples utilisés pour tester le classifieur est nommé base de test. Si le nombre d'exemples disponible est limité, on utilise souvent une partie de la base disponible comme base d'apprentissage et le reste comme base de test.

II.5.2. La validation croisée

La validation croisée est une procédure qui structure les itérations nécessaires à l'évaluation des performances du classifieur. Avant toute chose, il faut déterminer le nombre de réplifications désirées. Typiquement, la validation croisée est effectuée sur dix itérations, on parle alors d'une validation croisée à dix partitions. Dans un premier temps, la base de données doit être séparée en dix partitions disjointes stratifiées d'une taille approximativement égale. À la première itération, les neuf premières partitions sont utilisées pour l'apprentissage du classificateur, et le dernier est utilisée pour les tests. Ensuite, les partitions 2 à 10 sont utilisées pour l'apprentissage et la partition 1 est utilisée pour les tests. À la troisième itération, les partitions 3 à 10 et la partition 1 sont utilisées pour l'apprentissage, et la partition 2 est utilisée pour les tests, ainsi de suite. La Figure (II.6) présente le principe de la validation croisée sur 10 partitions.

Ainsi, l'algorithme de classification est entraîné dix fois sur autant de segments de données différents et dix tests indépendants sont conduits par la suite. Le nombre de partitions (itérations) peut être modifié pour répondre aux besoins particuliers d'une expérimentation [43].



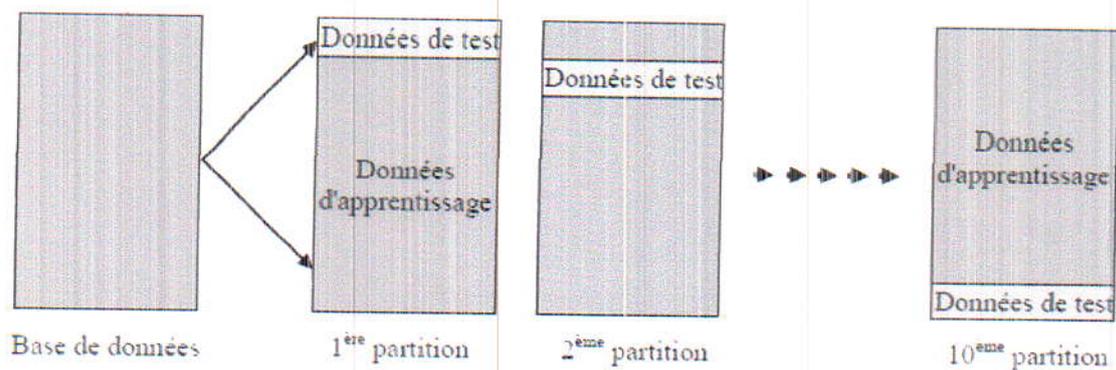


Figure II.6. Principe de la validation croisée pour $m = 10$.

En pratique, les jeux de données sont rarement suffisamment grands pour que l'erreur de test ainsi calculée estime correctement l'erreur de prédiction du modèle. La validation croisée est une alternative très populaire pour gérer la parcimonie des données. Il s'agit de découper le jeu de données en K groupes tirés aléatoirement qui font successivement servir d'ensemble de test. On peut alors calculer une erreur de test pour chacun des groupes et en faire la moyenne, ce qui constitue l'estimateur de l'erreur de test par validation croisée.

II.5.3. Matrice de confusion

L'outil le plus important pour déterminer la qualité d'un algorithme de classification supervisée est la matrice de confusion. Il s'agit d'un tableau de contingence à double entrées qui croise les groupes prédits par l'ordinateur avec ceux prédits manuellement pour ces mêmes objets [44].

En cas de concordance, les objets sont dénombrés sur la diagonale de cette matrice carrée. Donc, tous les dénombrements hors diagonale représentent des erreurs faites par l'ordinateur. Les statistiques tirées par cette matrice de confusion sont :

- **Le taux de reconnaissance correcte global** est la somme des individus sur la Diagonale divisée par le nombre total d'individus.
- **Le nombre de vrais positifs (VP)** est le nombre d'individus du groupe d'intérêt qui sont correctement classés.
- **Le nombre de vrais négatifs (VN)** est le nombre d'individus de tous les autres groupes qui sont correctement classés.

- Le nombre de faux positifs (FP) est le nombre d'individus des autres groupes qui sont classés dans le groupe d'intérêt.

- Le nombre de faux négatifs (FN) est le nombre d'individus du groupe d'intérêt qui sont classés dans les autres groupes.

		<i>Classification automatique</i>	
		<i>Positif</i>	<i>Négatif</i>
<i>Classification Manuelle</i>	<i>Positif</i>	<i>VP</i>	<i>FN</i>
	<i>Négatif</i>	<i>FP</i>	<i>VN</i>

Tableau II.1. Matrice de confusion.

II.5.4. Courbe ROC (Receiver Operating Characteristics)

La courbe Receiver Operating Characteristic (ROC) fut inventée durant la seconde guerre mondiale pour aider les opérateurs radar à décider si les spots apparaissant sur leur écran représentaient un avion ennemi, un avion ami ou juste du bruit. De nos jours, la courbe ROC est par exemple très utilisée en médecine pour évaluer les risques associés à une décision thérapeutique [45].

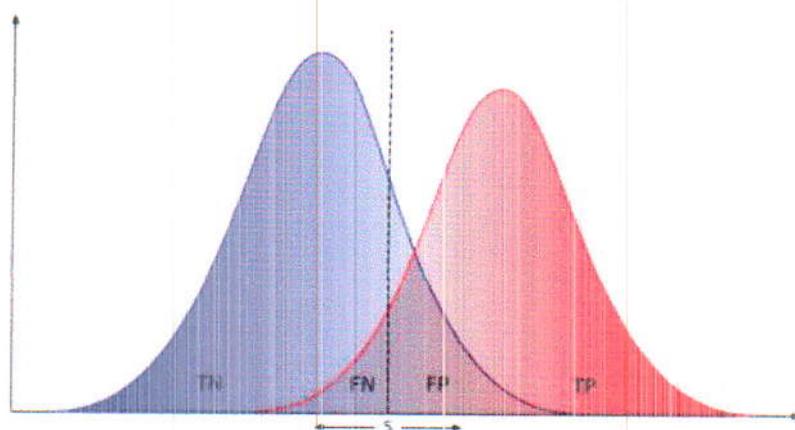


Figure II.7. Décision d'un classifieur binaire en fonction du seuil de discrimination s : la courbe bleue représente la probabilité a posteriori de la classe « négative » et la courbe rouge représente la probabilité a posteriori de la classe « positive ».

Avant d'entrer dans le détail de la construction des courbes ROC, il est nécessaire de définir certaines quantités entrant en jeu dans les deux cas. Un classifieur binaire, dont on souhaite évaluer la performance, va affecter chacune des observations à classer au groupe des positifs ou au groupe des négatifs. Cependant, l'origine des observations de l'échantillon de validation étant connues, il nous est possible de diviser le groupe des points positifs de la façon suivante [46] et [47] :

$$\# \text{ Positifs} = \text{TP} + \text{FN} \quad (\text{II.6})$$

Où TP est le nombre de vrais positifs, i.e. points classés positifs à juste titre, et FN est le nombre de faux négatifs, i.e. points classés négatifs à tort. De même, le groupe des points négatifs peut être divisé de la façon suivante :

$$\# \text{ Négatifs} = \text{TN} + \text{FP} \quad (\text{II.7})$$

Où TN est le nombre de vrais négatifs, i.e. points classés négatifs à juste titre, et FP est le nombre de faux positifs, i.e. points classés positifs à tort. La figure (II.7) permet de visualiser les différentes quantités entrant en jeu dans la construction de la courbe ROC d'un classifieur binaire.

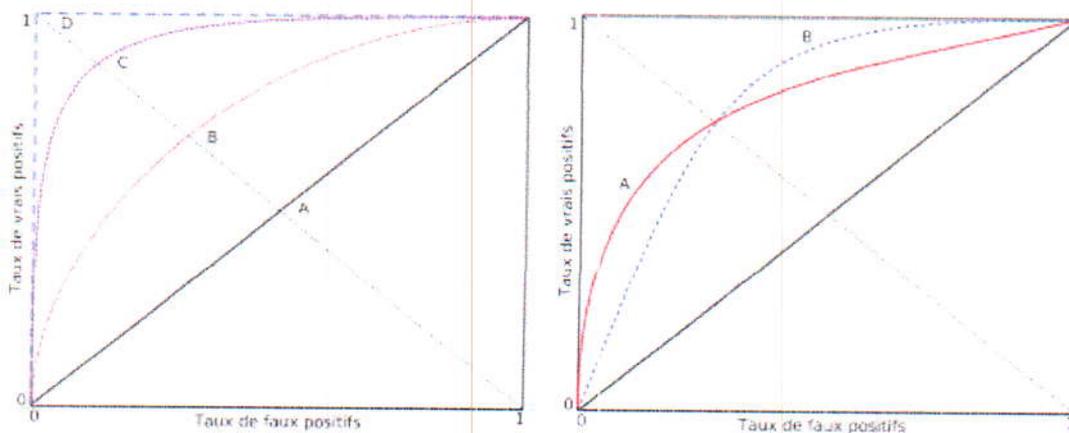


Figure II.8. Courbes ROC : à gauche, les courbes A, B, C et D sont respectivement associées à des classifieurs de performances croissantes. A droite, les courbes A et B ont la même aire (AUC) et la même valeur EER et sont pourtant associées à des classifieurs qui n'ont pas les mêmes qualités.

La Figure (II.8), présente des exemples de courbes ROC. Les courbes A, B, C et D, représentées sur l'image de gauche, sont respectivement associées à des classifieurs de performances croissantes. En particulier, la courbe A est associée au classifieur aléatoire, qui affecte chacune des observations au hasard, alors que la courbe D est associée au classifieur optimal. Les courbes ROC permettent donc de visualiser les caractéristiques d'un classifieur. Ainsi, le classifieur associé à la courbe A de l'image de droite de Figure II.8, est un classifieur peu performant mais relativement précis. A l'inverse, le classifieur associé à la courbe B est un classifieur performant mais au prix d'un grand nombre de fausses détections.

La courbe ROC exprime la sensibilité en fonction de $(1 - \text{spécificité})$ pour différentes valeurs du seuil de discrimination. La sensibilité, qui est la probabilité de détecter correctement un positif, se définit comme le taux de vrais positifs :

$$\text{Sensitivité} = \frac{TP}{\#positifs} \quad (\text{II. 8})$$

La spécificité est quant à elle la probabilité de détecter correctement un négatif et elle se définit par la quantité $(1 - \text{spécificité})$ qui est égale au taux de faux positifs :

$$\text{Spécificité} = \frac{FP}{\#négatifs} \quad (\text{II. 9})$$

II.6. Conclusion

Le diagnostic de la reconnaissance de forme est passé par l'étape de la classification, les méthodes de la classification se divisent en deux familles : les méthodes supervisées et les méthodes non supervisées, dans ce chapitre, nous avons présenté les différentes approches de la classification à savoir la méthode K-NN ainsi que l'étape d'évaluation de taux de reconnaissance qui s'est effectuée par la validation croisé afin d'étudier les performances des méthodes utilisées pour classifier les images texturées et de les comparer.

III.1. Introduction

Dans ce chapitre nous allons présenter les résultats de la classification des images texturées en utilisant les caractéristiques obtenues par la méthode LBP. L'objectif d'utiliser la classification est de vérifier la performance des descripteurs de texture choisis et de trouver les classes appropriées aux images fournies au système. Afin de pouvoir évaluer notre méthode, nous avons utilisé une base de données constituée des images de différentes textures pour extraire les descripteurs LBP et nous avons effectué la classification par validation simple et validation croisée.

III.2. Présentation de la base de données

La base d'image de texture UIUCTex est considérée comme une base de référence, elle est employée dans la littérature afin de comparer expérimentalement les résultats de classification de texture obtenue par différentes méthodes [48].

La base de données d'UIUCTex contient 1000 images de taille 640x480 réparties sur 25 classes de textures avec 40 images par classe. Les textures sont vues sous différents échelles avec des changements, des déformations et des variations d'apparence. La figure (III.1) présente quatre images d'échantillon par classe, chacune montre une surface texturisée vue à des positions différentes.

III.3. Implémentation et résultats

L'extraction des caractéristiques de textures est la phase élémentaire de la classification de texture. De bons résultats à cette étape facilitent les étapes suivantes. Nous avons utilisé le descripteur LBP ainsi que ses variantes LBP uniforme, LBP invariant par rotation et LBP uniforme et invariant par rotation. Ces méthodes sont expliquées en détail dans le chapitre I. La figure (III. 2) résume l'implémentation de la méthode LBP et la figure (III.3) donne un exemple d'application du LBP sur une image de la base de données. Cette étape permet de créer la base de données image en la sauvegardant dans un fichier (-.mat) tout en extrant les caractéristiques visuelles pour chaque image (vecteur caractéristique) et le mettre dans une matrice.

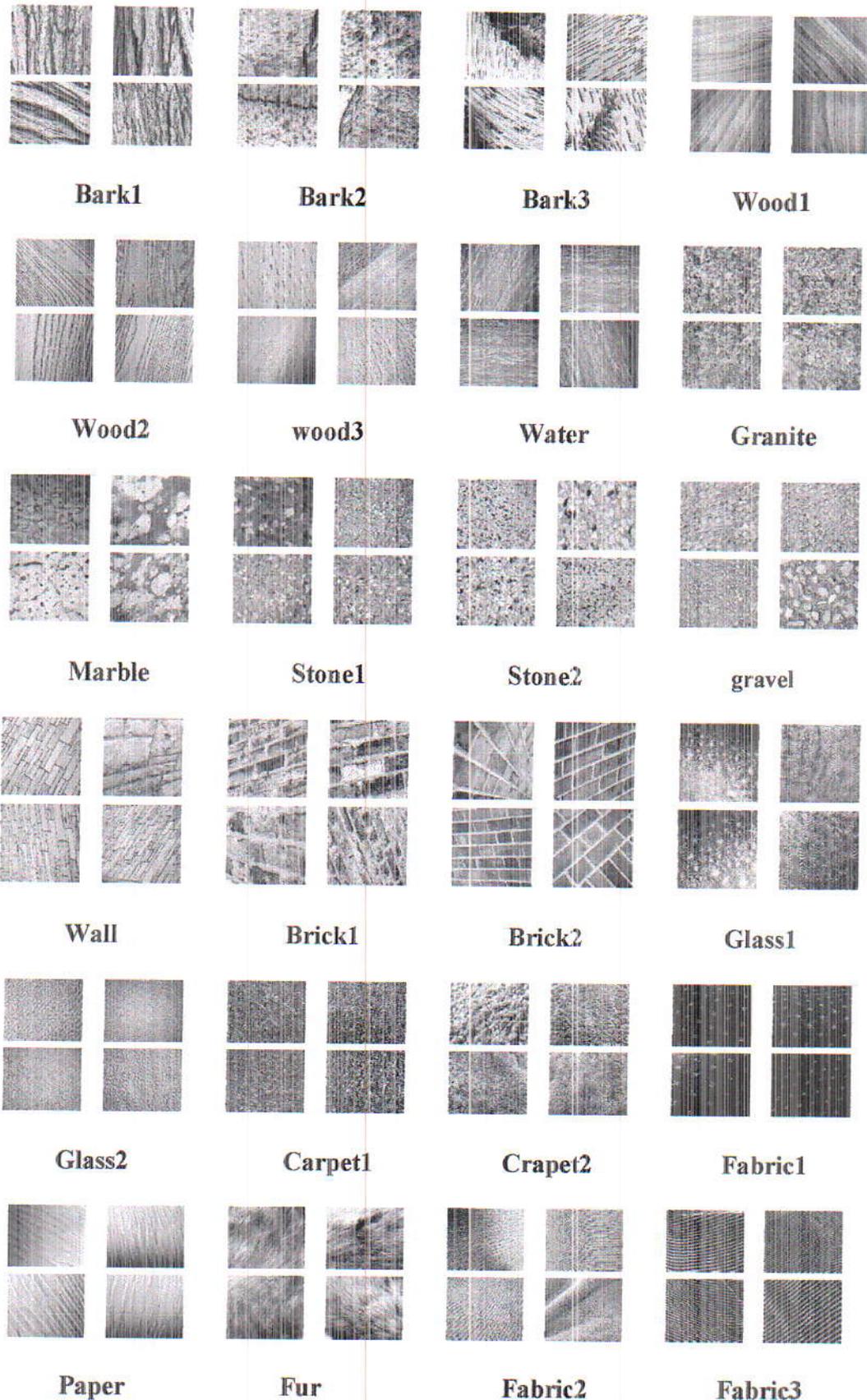


Figure III.1. Echantillons d'image de texture de la base UIUCTex[49].

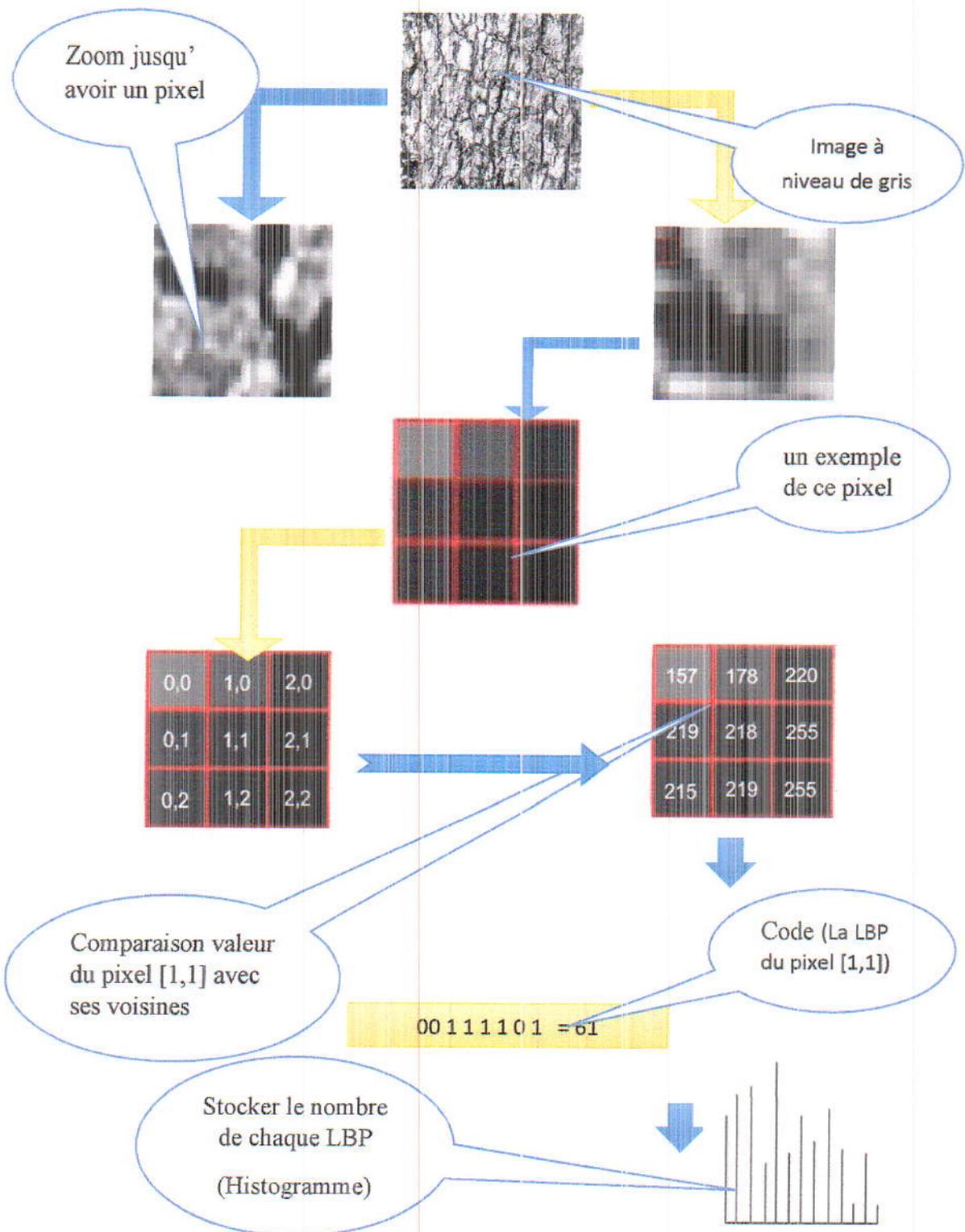


Figure III.2. Exemple d'application du code LBP sur une image de la base de données.

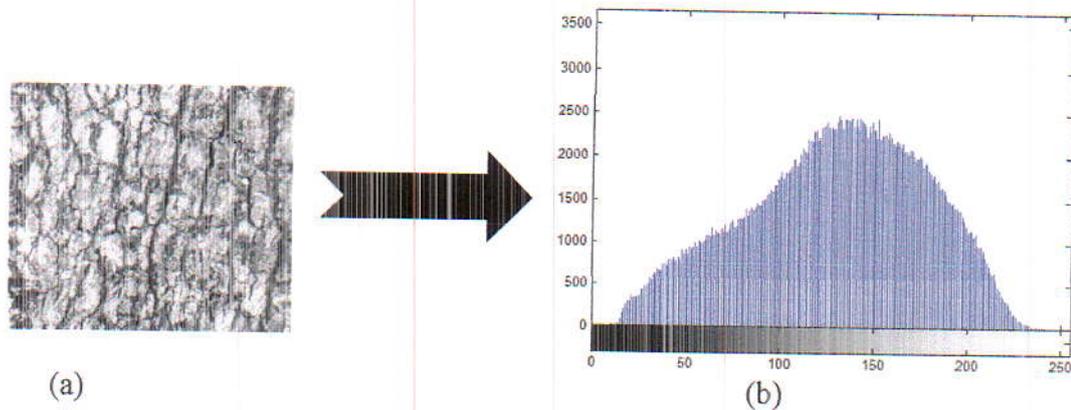


Figure III.3. Application du LBP sur une image de texture (a) : Image originale, (b) : Histogramme des codes LBP.

Pour réaliser notre travail, nous avons partitionné la base de données en deux groupes : un groupe d'apprentissage et un groupe de test, selon deux approches : une validation simple, et une validation croisée (voir chapitre II). Chaque image est encore divisée en blocs de 160×120 , on aura donc 4000 images de 25 classes, 160 images pour chaque classe.

Après avoir effectué l'extraction des caractéristiques, on procède à la classification en utilisant la méthode KNN. La méthode des K plus proches voisins (KNN) a pour but de classer des points cibles (classe méconnue) en fonction de leurs distances par rapport à des points constituant un échantillon d'apprentissage (c'est-à-dire dont la classe est connue a priori).

Ce travail consiste qu'à partir d'une matrice de données (dataset) dont chaque ligne présente un vecteur caractéristique d'une telle image où le nombre de lignes de cette matrice est le nombre total des images (4000), on peut représenter l'ensemble des images de cette matrice dans un espace vectorielle à N dimensions (selon le nombre de colonnes de cette matrice qui est le nombre de caractéristiques) d'où chaque ligne dessine un point dans cet espace.

Usuellement, on utilise la distance Euclidienne pour calculer la distance entre l'image à classer (qui est représenté par un point aussi dans cet espace) et les autres points en triant les distances d'ordre croissant, puis affichons la classe la plus proche élue.

La figure (III.4) illustre les différentes étapes de notre travail.

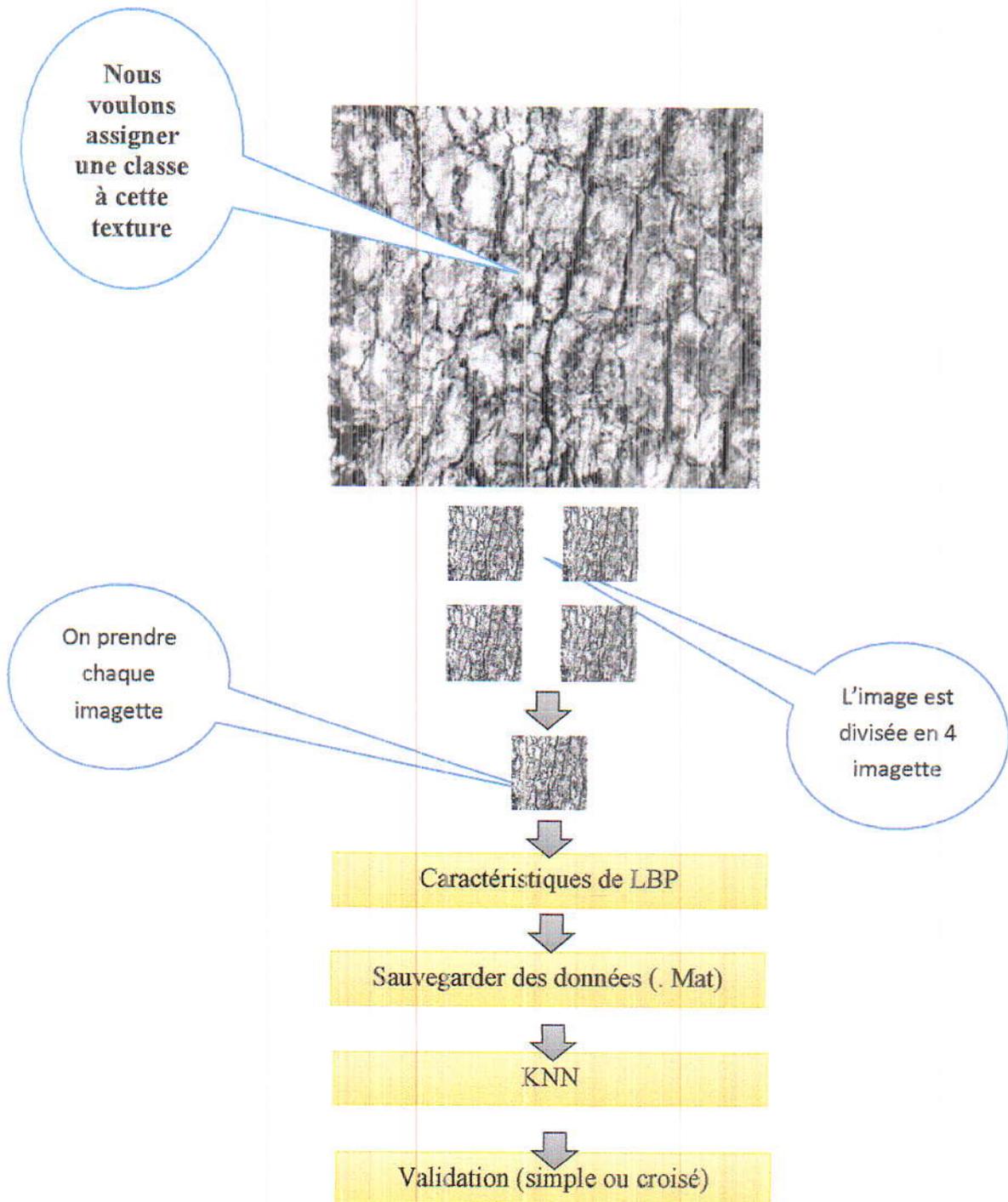


Figure III.4. Schéma générale du travail effectué.

III.4. Résultats et discussions

III.4.1. La validation simple

En divisant l'ensemble des échantillons en deux parties aléatoirement, une partie pour l'apprentissage (80%) et une partie pour le test (20%), nous aurons ainsi 3200 images pour l'apprentissage dont 128 pour chaque classe et 800 images pour le test dont 32 images pour chaque classe. Les résultats présentés dans les tableaux 1 et 2 sont exprimés en termes du taux d'erreur (%) : on présente aussi les valeurs de la sensibilité et de la spécificité pour $K=1$ et $K=3$ et ceci pour les 4 méthodes d'extraction de caractéristiques à savoir LBP, LBP uniforme, LBP invariante par rotation et LBP uniforme et invariante par rotation. Les paramètres utilisés sont :

- Le rayon $R=1, 2$.
- Le nombre de pixels voisins $P=8, 16$.
- Le $K= 1,3$, qui spécifie KNN.

Méthodes		1NN	3NN
LBP	R=1, P=8	<u>09.50</u>	<u>13.13</u>
	R=1, P=8	<u>11.87</u>	16.75
LBP Uniforme	R=2, P=16	19.36	26.25
	R=1, P=8 + R=2, P=16	<u>13.50</u>	18.63
	R=1, P=8	26.75	26.12
LBP Invariant rotation	R=2, P=16	16.50	16.50
	R=1, P=8 + R=2, P=16	15.50	16.00
	R=1, P=8	28.25	30.00
LBP Uniforme et Invariant par rotation	R=2, P=16	22.50	25.50
	R=1, P=8 + R=2, P=16	18.25	20.75
	R=1, P=8	28.25	30.00

Tableau III.1. Taux (%) d'erreur pour la validation simple.

Méthodes		Sensibilité		Spécificité	
		1NN	3NN	1NN	3NN
LBP	R=1, P=8	84.38	90.63	99.35	99.35
	R=2, P=16	84.38	78.13	98.70	98.83
LBP Uniforme	R=1, P=8	93.75	90.63	99.35	99.35
	R=2, P=16	84.38	78.13	98.70	98.83
LBP Invariant par rotation	R=1, P=8	81.25	78.13	98.44	99.09
	R=2, P=16	84.38	84.38	<u>99.22</u>	<u>99.22</u>
LBP Uniforme et Invariant par rotation	R=1, P=8	87.50	73.12	<u>97.92</u>	<u>98.52</u>
	R=2, P=16	84.38	71.13	<u>99.35</u>	<u>98.57</u>

Tableau III.2. La sensibilité et la spécificité (%) pour la validation simple.

III.4.2. La validation croisée

En divisant l'ensemble des échantillons en quatre parties aléatoirement, pour chaque itération (4 itérations) une partie est utilisée pour le test (1000 échantillons) et les 3 autres parties (3000 échantillons) sont utilisées pour l'apprentissage, l'erreur de classification est la moyenne des 4 erreurs obtenues pour les 4 parties de test.

Les résultats présentés dans les tableaux 3 et 4 sont exprimés en termes du taux d'erreur (%): on présente aussi les valeurs de la sensibilité et de la spécificité pour $K=1$ et $K=3$ et ceci pour les 4 méthodes d'extraction de caractéristiques à savoir LBP, LBP uniforme, LBP invariante par rotation et LBP uniforme et invariante par rotation. Les mêmes paramètres sont utilisés :

- Le rayon $R=1, 2$.
- Le nombre de pixels voisins $P=8, 16$.
- Le $K=1, 3$, qui spécifie KNN.

Méthodes		1NN	3NN
LBP	R=1, P=8	<u>09.35</u>	<u>13.96</u>
	R=1, P=8	<u>14.54</u>	17.98
LBP Uniforme	R=2, P=16	22.58	25.33
	R=1, P=8 + R=2, P=16	<u>14.32</u>	20.25
	R=1, P=8	26.86	26.99
LBP Invariant rotation	R=2, P=16	17.48	19.28
	R=1, P=8 + R=2, P=16	<u>16.30</u>	17.85
	R=1, P=8	28.23	30.95
LBP Uniforme et Invariant par rotation	R=2, P=16	22.68	19.90
	R=1, P=8 + R=2, P=16	<u>18.10</u>	21.30

Tableau III.3. Taux (%) d'erreur pour la validation croisée.

Méthodes		Sensibilité		Spécificité	
		1NN	3NN	1NN	3NN
LBP	R=1, P=8	93.13	91.25	99.58	99.35
	R=1, P=8	90.93	87.71	99.30	98.90
LBP uniforme	R=2, P=16	80.09	71.98	98.74	98.09
	R=1, P=8	73.75	71.25	98.88	98.80
LBP Invariant par rotation	R=2, P=16	87.50	84.38	99.32	99.11
	R=1, P=8	75.00	67.50	98.96	98.46
LBP Uniforme et Invariant par rotation	R=2, P=16	80.73	72.94	98.87	98.66

Tableau III.4. La sensibilité et la spécificité (%) pour la validation croisée.

Dans les tableaux ci-dessous (5 et 6), nous présentons les taux d'erreur pour chacune des 25 classes, nous utilisons la validation croisée et la validation simple pour les différentes méthodes de LBP pour $R=1,2$ et $P=8,16$ et $K=1$.

❖ Pour la Validation croisée

Méthode Classe	LBP base	LBP uniforme		LBP invariant par rotation		LBP uniforme et invariant rotation	
	R=1	R=1	R=2	R=1	R=2	R=1	R=1
	P=8	P=8	P=16	P=8	P=16	P=8	P=16
Bark1 	06.87	11.25	20.00	26.25	12.50	25.00	18.75
Bark2 	35.62	40.62	50.00	50.62	38.12	55.00	46.87
Bark3 	17.50	26.25	36.25	50.62	38.12	48.75	47.50
Wood1 	10.00	18.75	25.00	25.62	17.50	30.62	27.50
Wood2 	03.12	06.87	25.00	20.00	07.50	26.87	18.12
Wood3 	08.12	23.12	34.37	42.50	35.62	43.12	36.87
Water 	01.87	<u>02.50</u>	<u>01.87</u>	03.75	00.00	<u>04.37</u>	<u>06.25</u>
Granite 	16.25	14.37	23.75	28.75	15.62	25.00	28.75
Marble 	11.87	26.00	29.73	23.12	19.37	26.87	27.50
Stone1 	11.87	13.75	35.62	28.12	19.37	30.62	25.00
Stone2 	06.25	10.00	16.87	11.87	06.87	15.62	10.00

Gravel 	11.87	19.37	18.12	22.50	13.12	23.75	18.12
Wall 	<u>01.87</u>	05.00	14.37	12.50	08.75	16.87	15.00
Brick1 	03.75	13.12	26.87	29.37	15.00	33.75	20.62
Brick2 	03.12	06.25	14.37	14.37	15.00	20.62	22.50
Glass1 	03.12	06.25	15.00	10.00	08.12	13.75	12.50
Glass 2 	05.62	07.50	11.87	<u>08.75</u>	<u>05.00</u>	08.75	09.37
Carpet1 	04.37	09.37	33.75	10.62	04.37	07.50	08.75
Crapet2 	10.62	16.87	28.12	34.43	30.00	33.75	43.125
Fabric1 	12.50	15.00	25.62	32.50	20.62	31.25	29.37
Paper 	05.62	10.00	16.87	39.37	35.00	40.00	46.87
Fur 	23.75	28.12	44.37	41.25	16.25	36.87	20.62
Fabric2 	07.50	08.75	25.00	35.62	16.25	36.87	21.87
Fabric3 	08.75	14.37	16.25	23.75	15.62	29.37	20.62
Fabric4 	09.37	10.62	20.00	39.37	20.62	40.62	25.00

Tableau III.5. Taux d'erreur (%) des différentes classes pour la validation croisée.

❖ Pour La Validation simple

Méthode Classe	LBP base	LBP uniforme		LBP invariant par rotation		LBP uniforme et invariant rotation	
	R=1	R=1	R=2	R=1	R=2	R=1	R=1
	P=8	P=8	P=16	P=8	P=16	P=8	P=16
Bark1 	15.62	06.25	15.62	18.75	15.62	12.50	15.62
Bark2 	28.12	28.12	65.62	53.12	50.00	56.25	50.00
Bark3 	21.87	25.00	03.12	46.87	53.12	56.25	00.00
Wood1 	09.37	06.25	21.00	31.25	18.75	37.50	21.87
Wood2 	00.00	06.25	06.00	37.50	06.25	25.00	25.00
Wood3 	00.00	09.37	27.00	43.75	28.12	46.87	40.62
Water 	06.25	03.12	00.00	00.00	00.00	06.25	00.00
Granite 	15.62	25.00	31.00	37.50	09.37	25.00	31.25
Marble 	9.37	12.50	21.00	31.25	21.75	21.87	12.50
Stone1 	06.25	09.37	25.00	18.75	18.75	25.00	25.00
Stone2 	06.25	15.62	09.00	06.25	06.25	15.62	15.62
Gravel 	06.25	03.12	15.00	21.87	00.00	25.00	31.25
Wall 	03.12	03.12	21.00	15.62	6.25	12.50	15.62
Brick1 	00.00	12.50	06.00	18.75	09.37	31.25	09.37

Brick2 	03.12	06.25	15.00	15.62	06.25	18.75	06.25
Glass1 	<u>00.00</u>	<u>00.00</u>	<u>00.00</u>	09.37	03.12	25.00	06.25
Glass 2 	09.37	06.25	21.00	06.25	09.37	09.37	25.00
Carpet1 	12.50	18.75	<u>00.00</u>	15.62	06.25	<u>06.25</u>	12.50
Crapet2 	18.75	09.37	18.00	43.75	31.25	21.87	43.75
Fabric1 	09.37	31.25	18.00	28.12	09.37	34.37	21.87
Paper 	06.25	06.25	06.00	34.37	34.37	43.75	59.37
Fur 	25.00	18.75	46.00	34.37	09.37	43.75	15.62
Fabric2 	06.25	12.50	15.00	37.50	18.75	40.62	21.87
Fabric3 	03.12	15.62	03.12	28.125	15.62	21.87	12.50
Fabric4 	15.62	06.25	18.75	34.37	25.00	43.75	15.62

Tableau III.6. Taux d'erreur (%) des différentes classes pour la validation simple.

Pour examiner de proche les erreurs de la classification effectuée, nous donnons ci-dessous les matrices de confusion de la classification pour $K=1$ et $R=1$, $p=8$ et ceci pour les 4 méthodes d'extraction de caractéristiques à savoir LBP, LBP uniforme, LBP invariante par rotation et LBP uniforme et invariante par rotation.....

Classe	Bark1	Bark2	Bark3	Wood1	Wood2	Wood3	Water	Granite	Marble	Stone1	Stone2	Gravel	Wall	Brick1	Brick2	Glass1	Glass2	Carpet1	Carpet2	Fabric1	Paper	Fur	Fabric2	Fabric3	Fabric4
Bark1	146	2	1	0	0	0	0	1	0	0	0	2	0	3	0	0	0	0	0	0	0	2	0	0	
Bark2	102	102	1	0	0	0	0	21	0	3	0	20	1	7	0	0	0	0	5	0	0	0	0	0	
Bark3	5	3	129	1	1	3	1	0	0	3	2	4	1	1	0	0	0	1	0	1	0	3	0	0	
Wood1	0	0	0	139	1	14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
Wood2	0	0	0	157	2	1	157	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	0	0	
Wood3	0	0	0	14	1	150	0	0	1	0	150	0	1	0	0	0	0	0	0	0	1	2	0	0	
Water	0	0	1	0	0	0	151	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Granite	1	21	0	0	0	0	0	137	0	0	0	7	1	0	0	1	0	1	4	0	0	2	0	0	
Marble	0	0	0	0	0	0	0	0	143	7	0	0	0	0	0	0	3	0	0	1	1	0	0	0	
Stone1	0	3	3	0	0	0	0	0	7	144	4	3	0	0	0	0	1	0	0	1	0	3	0	0	
Stone2	0	0	0	0	0	0	0	1	0	3	152	0	0	0	0	0	0	4	0	0	0	2	0	0	
Gravel	2	20	4	0	0	0	0	0	0	2	0	139	0	0	0	1	0	1	7	0	9	1	0	0	
Wall	0	1	1	5	1	3	0	0	0	0	0	1	154	1	0	0	0	0	0	1	0	2	1	0	
Brick1	3	7	1	0	0	0	0	0	0	0	0	0	0	148	0	0	0	0	0	0	1	1	0	0	
Brick2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	1	0	0	0	
Glass1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	
Glass2	0	0	0	0	0	0	2	0	5	4	1	3	0	0	0	0	155	0	0	0	0	1	0	0	
Carpet1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	149	0	0	0	0	0	0	
Carpet2	0	5	0	0	0	0	0	0	0	0	0	4	0	0	0	0	2	0	143	0	1	1	0	0	
Fabric1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	139	0	0	0	0	
Paper	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	151	0	0	0	
Fur	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	124	0	0	
Fabric2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	150	0	
Fabric3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	145	0	
Fabric4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	145	4	

Tableau III.7. Matrice de confusion pour LBP simple.

Classe	Bark1	Bark2	Bark3	Wood1	Wood2	Wood3	Water	Granite	Marble	Stone1	Stone2	Gravel	Wall	Brick1	Brick2	Glass1	Glass2	Carpet1	Carpet2	Fabric1	Paper	Fur	Fabric2	Fabric3	Fabric4
Bark1	146	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bark2	1	78	1	0	0	0	0	10	0	2	1	6	2	8	0	0	0	1	9	0	0	0	0	5	0
Bark3	0	1	127	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	4	0	0	2
Wood1	1	0	2	138	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
Wood2	0	0	1	1	155	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Wood3	0	0	2	12	1	140	0	0	0	0	0	0	6	1	2	0	0	0	0	0	1	0	0	0	1
Water	0	0	0	0	1	0	157	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
Granite	1	22	0	0	0	0	0	140	0	0	0	3	0	0	0	1	0	3	0	2	0	0	1	0	0
Marble	0	1	0	0	0	0	0	0	143	6	1	0	0	0	0	0	5	0	3	0	2	0	1	0	0
Stone1	1	4	2	0	0	1	0	1	5	139	8	2	0	0	0	2	4	0	0	3	0	3	2	0	1
Stone2	0	1	1	0	0	0	0	0	1	5	145	3	0	0	0	2	0	3	0	0	0	3	0	0	0
Gravel	1	26	5	0	0	0	2	0	1	2	3	131	1	0	1	0	1	3	7	3	0	0	0	0	0
Wall	0	1	1	9	0	0	0	0	0	0	0	0	151	6	0	0	0	0	0	1	0	0	0	0	0
Brick1	1	14	2	0	0	0	1	0	0	0	0	1	0	145	1	0	0	0	0	1	0	0	0	0	1
Brick2	0	0	1	0	0	0	0	0	0	0	0	0	0	147	0	0	0	0	0	0	3	0	0	0	0
Glass1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	155	0	0	0	0	0	2	0	0	0
Glass2	0	1	0	0	0	0	2	10	2	0	1	1	0	0	0	0	149	0	0	4	1	0	1	0	0
Carpet1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	146	0	1	0	0	0	0	0
Carpet2	0	11	0	0	0	0	4	0	0	0	12	0	0	0	0	0	0	0	140	4	0	1	0	0	0
Fabric1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	138	1	0	0	4	0
Paper	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	149	1	0	0	0	0
Fur	3	0	4	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	122	0	2	0	0
Fabric2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	147	2	0	
Fabric3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	135	4	2	
Fabric4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	141	5	2	

Tableau III.8. Matrice de confusion pour LBP uniforme.

Classe	Bark1	Bark2	Bark3	Wood1	Wood2	Wood3	Water	Granite	Marble	Stone1	Stone2	Gravel	Wall	Brick1	Brick2	Glass1	Glass2	Carpet1	Carpet2	Fabric1	Paper	Fur	Fabric2	Fabric3	Fabric4
Bark1	125	2	16	0	1	0	0	0	0	0	0	7	0	6	0	4	0	9	1	3	0	3	2	0	
Bark2	0	80	2	0	0	0	0	21	1	2	0	20	4	16	0	0	0	0	12	1	0	0	2	0	
Bark3	0	0	79	2	1	4	1	0	0	3	4	4	13	4	0	3	0	2	0	0	2	0	3	0	
Wood1	0	0	3	121	5	10	6	0	0	0	0	0	8	0	0	2	0	0	0	0	0	0	0	0	
Wood2	0	0	0	3	125	1	0	0	0	0	0	0	3	0	3	0	0	0	0	0	0	0	0	0	
Wood3	0	0	0	0	0	86	0	0	11	3	0	0	13	1	14	0	0	0	0	0	2	13	0	0	
Water	0	0	0	0	0	0	151	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Granite	0	0	0	0	0	0	0	121	1	0	0	3	0	0	0	0	1	0	31	0	7	0	11	0	
Marble	0	0	0	0	0	0	0	0	126	10	2	0	0	0	0	0	3	0	0	2	17	0	0	0	
Stone1	0	2	3	0	0	0	0	0	14	126	8	2	0	0	0	0	1	1	0	3	7	3	1	0	
Stone2	0	0	0	0	0	0	0	0	0	139	9	0	0	0	0	0	0	1	0	3	1	5	3	10	
Gravel	9	0	0	0	0	0	0	0	0	0	0	117	0	5	0	0	0	2	3	0	0	4	2	5	
Wall	0	4	0	0	0	0	0	0	0	0	0	0	135	2	0	0	0	0	1	1	2	5	0	1	
Brick1	1	0	0	0	0	0	0	0	0	0	0	0	0	118	0	2	0	0	0	0	0	4	0	1	
Brick2	0	0	0	0	0	0	0	0	1	0	0	0	4	0	134	0	0	0	0	1	6	0	0	1	
Glass1	5	0	0	0	2	0	0	0	0	0	0	0	0	0	0	139	0	0	0	0	0	8	1	1	
Glass2	0	0	0	1	0	0	0	1	3	1	1	0	0	0	0	149	0	1	0	3	4	0	1	4	
Carpet1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	134	0	0	3	0	0	7	0	
Carpet2	0	12	0	0	0	0	0	16	0	0	0	3	0	0	0	0	0	103	0	3	0	1	2	3	
Fabric1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	115	2	1	0	0	6	
Paper	0	0	2	1	0	0	0	10	0	1	1	0	0	0	0	0	0	0	100	2	1	0	0	4	
Fur	2	0	6	0	8	0	0	0	0	3	1	0	0	12	0	0	0	0	0	3	97	1	0	19	
Fabric2	2	2	0	0	2	0	0	0	0	0	2	6	0	0	0	0	3	0	0	0	1	108	5	0	
Fabric3	1	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	0	3	
Fabric4	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	4	0	0	91	

Tableau III.9. Matrice de confusion pour LBP invariante par rotation.

Classe	Bark1	Bark2	Bark3	Wood1	Wood2	Wood3	Water	Granite	Marble	Stone1	Stone2	Gravel	Wall	Brick1	Brick2	Glass1	Glass2	Carpet1	Carpet2	Fabric1	Paper	Fur	Fabric2	Fabric3	Fabric4
Bark1	124	2	9	0	0	0	0	0	0	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0	
Bark2	2	74	1	0	0	0	0	10	0	0	0	6	0	5	0	0	0	0	13	2	3	0	4	3	
Bark3	9	0	74	1	3	0	0	0	0	0	0	0	2	0	0	0	0	1	1	0	3	5	0	0	
Wood1	0	0	0	107	6	0	8	0	0	0	0	0	0	1	7	0	0	0	0	0	0	0	0	0	
Wood2	0	0	5	4	116	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	3	0	0	
Wood3	0	0	6	14	1	88	0	0	11	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	
Water	0	0	1	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
Granite	0	21	0	0	0	0	0	110	0	0	0	6	0	0	0	0	1	0	25	9	0	0	0	0	
Marble	0	0	0	0	0	0	0	0	125	0	0	0	0	0	0	0	3	0	0	19	0	0	0	0	
Stone1	0	2	3	1	0	1	0	1	11	18	0	0	0	0	0	0	1	1	0	10	2	3	0	0	
Stone2	0	0	5	0	0	0	0	0	13	115	9	1	0	0	0	0	0	1	4	1	5	3	2	6	
Gravel	9	17	3	0	0	0	5	0	0	0	0	122	0	3	0	0	0	4	0	0	7	0	3	4	
Wall	0	4	9	11	6	13	0	0	0	0	0	0	140	0	0	0	0	0	0	3	3	0	0	0	
Brick1	0	14	5	0	0	1	1	0	0	0	0	2	6	112	0	0	0	0	0	0	7	0	0	2	
Brick2	0	0	0	14	4	18	0	1	0	0	0	0	4	0	127	0	0	0	0	5	0	0	1	1	
Glass1	6	0	2	0	1	1	0	0	0	0	0	1	0	9	0	137	0	0	0	0	11	2	0	0	
Glass2	0	0	0	3	0	0	1	0	6	0	1	0	0	0	0	145	0	0	0	4	0	0	0	0	
Carpet1	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	141	0	0	0	0	0	0	0	
Carpet2	0	15	0	0	0	0	22	0	1	0	1	0	0	0	0	0	0	104	3	0	1	3	2	3	
Fabric1	1	3	0	0	0	0	0	2	1	0	0	0	0	0	0	5	0	2	116	1	94	0	8	4	
Paper	0	1	2	1	0	0	0	0	7	2	0	0	0	0	4	0	0	0	0	0	2	1	0	0	
Fur	3	0	10	0	8	1	0	0	0	3	4	0	0	0	0	0	0	0	0	0	104	3	1	14	
Fabric2	1	4	3	0	0	0	8	0	0	2	2	1	0	0	0	0	4	6	0	0	0	0	4	2	
Fabric3	0	2	2	0	0	0	0	0	0	1	3	0	0	0	0	2	0	0	0	0	0	0	115	4	
Fabric4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86	4	

Tableau III.10. Matrice de confusion pour LBP uniforme et invariante par rotation.

III.4.3. Discussion

D'après les résultats expérimentaux obtenus, nous pouvons conclure les points suivants :

- La meilleure classification est obtenue par la méthode LBP basique avec un taux d'erreur de 9.11%, et des valeurs de sensibilité et de spécificité élevées.
- La diminution des performances pour les variantes de LBP (uniforme et invariante par rotation) est due à la réduction des nombres de caractéristiques (les formes uniformes) utilisées pour modéliser le voisinage des pixels qui sont insuffisantes dans les cas des changements de rotation, d'échelle et d'orientation.
- La fusion des caractéristiques LBP pour les deux paramètres P et R a permis d'améliorer le taux de classification par rapport aux performances individuelles de chacune des 3 méthodes LBP. Cette amélioration prouve la complémentarité entre les caractéristiques provenant des deux rayons $R=1,2$ et les deux voisinages $P=8, 16$, ce qui met l'accent sur l'importance de l'information du voisinage des pixels pour modéliser des images texturées réelles.
- Les tableaux des erreurs de classification pour chaque classe donnent une idée sur les classes qui sont difficiles à la reconnaissance. Nous pouvons remarquer que les classes Bark2, Bark3, présentent des taux d'erreurs élevés par rapport aux autres classes et ceci pour toutes les méthodes d'extraction LBP utilisées.
- Les matrices de confusion donnent plus d'information sur le nombre d'échantillons mal classés. Nous constatons que les images des classes Bark2 et 3 sont classées comme des images appartenant aux classes Granite, Fur, Gravel. Ces classes présentent une grande similitude même pour l'œil humain (figure III.5).
- La base de données utilisée contient des images avec des rotations significatives, des points de vue et des résolutions différentes pour chaque classe et des conditions d'illumination qui ne sont pas contrôlées, ce qui explique la difficulté de la classification de certaines classes.

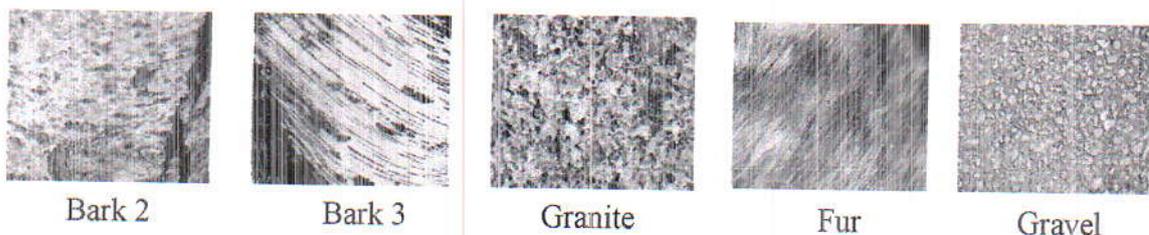
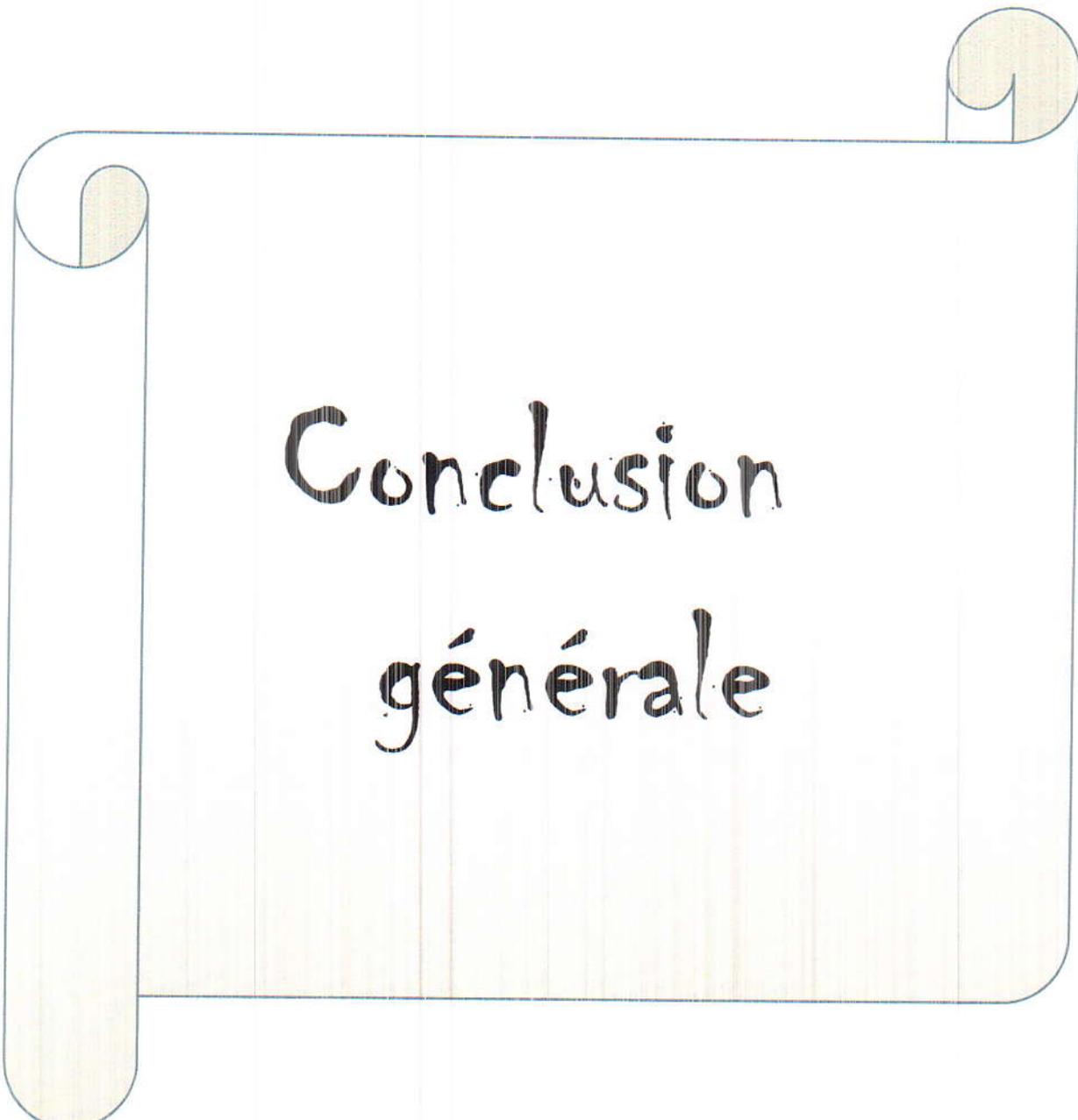


Figure III.5. Représentation des images de classe : Bark2, Bark3, Granite, Fur, Gravel.

III.5. Conclusion

Dans ce chapitre nous avons testé la méthode de classification K-NN pour classer les images texturées de la base de données UIUCTex. Nous avons effectué l'extraction des caractéristiques par les méthodes LBP en transformant les images en des vecteurs d'attributs. L'évaluation du système effectuée en calculant les différents taux d'erreur a permis d'estimer la capacité des attributs LBP à discriminer les différentes classes de texture. la méthode LBP basique avec le rayon $R=1$ et le nombre de voisin considéré $P=8$ donne des résultats meilleurs par rapport aux autres LBP utilisés. La concaténation des attributs avec les rayons $R=1,2$ et $P=8,16$ a réduit le taux d'erreur de classification pour les méthodes LBP uniforme et LBP invariant par rotation.



Conclusion générale

Conclusion générale

Dans ce travail, nous avons fait la classification des images texturées en utilisant une méthode d'extraction des caractéristiques qui est la méthode des motifs binaires locaux LBP. Cette méthode fait comparer le niveau de gris de chaque pixel avec les niveaux de gris de ses voisins. Cela rend donc compte d'une information relative à des motifs réguliers dans l'image, autrement dit une texture. Selon l'échelle du voisinage utilisé, certaines zones d'intérêt tel des coins ou des bords peuvent être détectées par ce descripteur LBP.

L'étape de la classification vise à déterminer la classe de l'image. L'algorithme de k plus proche voisin KNN est une méthode permettant de classer les objets en fonction des exemples d'apprentissage étiqueté les plus proches dans l'espace des caractéristiques. Elle se base sur une comparaison directe entre le vecteur de caractéristiques de l'image à classer et les vecteurs de caractéristiques des images de références. La comparaison consiste en un calcul de distance entre ces vecteurs. L'image à classer est assigné à la classe majoritaire parmi les classes des K exemples les plus proches au sens de la distance utilisée qui est la distance euclidienne dans notre cas.

Dans ce mémoire nous avons appliqué les méthodes ci-dessus sur la base de données UIUCTex constitué de 1000 images de 25 types de textures (bois, marbre, tissu, eau, verre)

L'évaluation est effectuée par deux types de validation qui sont la validation simple et la validation croisée, les résultats montrent des bons taux de classification pour la méthode d'extraction LBP et les trois autres méthodes de la famille LBP qui sont LBP uniforme, LBP invariante par rotation et LBP uniforme et invariante par rotation malgré la diversité et la complexité des images de textures utilisées. La méthode de motif binaire local simple LBP donne des résultats meilleurs par rapport aux autres variantes du LBP.

Comme perspective on propose :

- D'utiliser d'autres variantes des motifs binaires locaux LBP qui tiennent compte de d'autres informations lors du calcul des codes binaires.
- De combiner la méthode LBP qui est une méthode spatiale avec une méthode fréquentielle tel que les filtres de Gabor pour extraire plus d'information

Bibliographie

- [1] J. Sklansky. Image segmentation and feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pages 237–247, 1978.
- [2] H. Tamura, S. Mori and Y. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, pages 460–473, 1978.
- [3] R. M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, vol. 67, pages 786–804, 1979.
- [4] VisTex. Vision texture data base. Massachusetts Institute of Technology, 1995.
- [5] L. J. V. Gool, P. Dewaele, and A. Oosterlinck, Texture analysis anno 1983. *Computer Vision, Graphics, and Image Processing*, 29(3):336–357, 1985.
- [6] V. Vapnik, "Consistency of Learning Processes," in *The Nature of Statistical Learning Theory*: Springer New York, page. 33-64, 1995
- [7] M. Varma and A. Zisserman, "A Statistical Approach to Material Classification Using Image Patch Exemplars," *IEEE Transactions on, Pattern Analysis and Machine Intelligence*”, vol. 31, page. 2032-2047, 2009.
- [8] K. Brady, “A probabilistic framework for adaptive texture description”. Thèse de doctorat, Université de Nice-Sophia antipolis, 2003
- [9] M. Tuceryan and A. K. Jain. Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248. Editions World Scientific Publishing Co, 1998.
- [10] C. Zheng, D.W. Sun and L. Zheng. “A new region-primitive method for classification of colour meat image texture based on size”, orientation and contrast”. *Meat Science*, vol. 76, pages 620–627, 2007.

- [11] C. Palm and T. Lehmann. "Classification of color textures by gabor filtering Machine Graphics & Vision International", Journal, vol. 11, pages 195–219, 2002.
- [12] N Richard P. Carré, et C. Fernandez-Maloigne. Analyse et caractérisation des signaux bidimensionnels, chapitre Représentation en ondelettes. Traité I2C, 2001.
- [13] A. Pentland, Fractal based description of natural scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.6 page 661–674. 1984
- [14] Haralick 1973 R. M. Haralick, K. Shanmugan and I. Dinstein. Textural features for image classification. IEEE Trans. On Systems, Man, and Cybernetics, vol. 3, no. 6, page 610–621, Novembre 1973.
- [15] C. Palm. "Color texture classification by integrative co-occurrence matrices". Pattern recognition, vol. 37, pages 965–976, 2004.
- [16] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural Features for Image Classification, *IEEE Transactions on, Systems, Man and Cybernetics*", vol. SMC-3, page. 610-621, 1973.
- [17] C. Chen, L. F. Pau, P. S. Wang, and S. Wang, "Texture analysis," in Handbook of Pattern Recognition & Computer Vision, 2nd ed: World Scientific, 1999, page. 207-248.
- [18] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, page. 786-804, 1979.
- [19] A. Baraldi and F. Parmiggiani, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters, *IEEE Transactions on, Geoscience and Remote Sensing*", vol. 33, page. 293-304, 1995.
- [20] K. D. Fritscher, B. Schuler, A. Grünerbl, M. Hänni, K. Schwieger, N. Suhm, and R. Schubert, "Assessment of femoral bone quality using co-occurrence matrices and adaptive regions of interest," page. 65141K-65141K, 03/08 2007.
- [21] C. Hum-Yan, W. Lai-Khin, S. Tan-Tian, S. Sh-Hussain, and A. A.K Kamarulafizam, "Gray-Level Co-occurrence Matrix Bone Fracture Detection " *American Journal of Applied Sciences*, vol. 8, page. 26-32, 2011.

- [22] E. Lespessailles, C. Gadois, G. Lemineur, J. P. Do-Huu, and L. Benhamou, "Bone Texture Analysis on Direct Digital Radiographic Images: Precision Study and Relationship with Bone Mineral Density at the Os Calcis," *Calcified Tissue International*, vol. 80, page. 97-102, 2007.
- [23] T. K. Reddy and N. Kumaravel, "Segmentation and Classification of Jaw Bone CT images using Curvelet based Texture features," *Bangladesh Journal of Medical Science*, vol. 09, 2010.
- [24] A. Porebski, N. Vandenbroucke and L. Macaire. Attributs de texture couleur. In Proceedings de l'École d'Hiver sur l'Imagerie Numérique Couleur (EHINC'09), 2009. Ecole d'Ingénieurs du Pas-de-Calais (EIPC), Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS UMR CNRS), Université des Sciences et Technologies de Lille.2009
- [25] Tupin et M. Sigelle. Markov random fields. ISTE Ltd and John Wiley & Sons, Digital Signal and Image Processing Series, 2008.
- [26] O.J. Hernandez et A. Khotanzad. Color image segmentation using multispectral random field texture model and color content features. *Journal of Computer Science et Technologie*, vol. 4, pages 141–146, 2004.
- [27] T. Ojala, M. Pietikinen et D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, vol. 29, pages 51–59, 1996.
- [28] M. Pietikäinen, T. Mäenpää and J. Viertola. "Color texture classification with color histograms and local binary patterns". In Proceedings of the 2nd International Workshop on Texture Analysis and Synthesis, pages 109–112, 2002.
- [29] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, page. 971-987, 2002.
- [30] Pietikäinen, M., Ojala, T. et Xu, Z. (2000). Rotationinvariant texture classification using feature distributions. *Pattern Recognition*,33 page 43–52.
- [31] Z. Guo, LeiZhang and DavidZhang. Rotation invariante texture classification using lbp variance (lbpv) withglobalmatching. *Pattern Recognition*, 2009

- [32] Guillaume ROUSSEL. "Développement et évaluation de nouvelles méthodes de classification spatiale-spectrale d'images hyper spectrale", doctorat de l'université de Toulouse, 2012.
- [33] M. W. Ashour, M. F. Hussin, and K. M. Mahar, "Supervised texture classification using several features extraction techniques based on ANN and SVM," in *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*, 2008, page. 567-574.
- [34] H. Park and K. R. Park, "Iris recognition based on score level fusion by using SVM," *Pattern Recognition Letters*, vol. 28, page. 2019-2028, 2007.
- [35] A.K. Jain, R.P. W. Duin, et J. Mao "statistical pattern recognition: A review. IEE transactions on pattern Analysis and Machine intelligence ", vol. 22 page 4-37, 2000.
- [36] A.K. Jain, M.N. Murty, ET P.J. Flynn. "Data clustering: ACM computing Surveys", vol.31 page 264-323, 1999.
- [37] V. Vapnik, "Consistency of Learning Processes," in *The Nature of Statistical Learning Theory*: Springer New York, 1995, page. 33-64.
- [38] K. M. Rajpoot and N. M. Rajpoot, "Wavelets and support vector machines for texture classification," in *Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International*, 2004, page. 328-333.
- [39] M. Goldstein, "k-Nearest Neighbor Classification, IEEE Transactions on," *Information Theory*, vol. 18, page. 627-630, 1972.
- [40] K. Hechenbichler and K. Schliep, "Weighted k-Nearest-Neighbor Techniques and Ordinal Classification." *journal*. 399, 2004.
- [41] Y.-L. Qiao, Z.-M. Lu, J.-S. Pan, and S.-H. Sun, "Fast k-nearest neighbor search algorithm based on pyramid structure of wavelet transform and its application to texture classification," *Digital Signal Processing*, vol. 20, page. 837-845, 2010.

- [42] C. Rosenberger and K. Chehdi, "Genetic Fusion Application to multi-components image segmentation", ICASSP 2000, Vol. 3, page. 2223-2226, Istanbul, Turkey, 2000.
- [43] H. W. Ian and F. Eibe, Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems): Morgan Kaufmann Publishers Inc., 2005.
- [44] T. Fawcett and Povost Fawcett, Adaptative Fraud Detection, vol 1 of Data Mining and Knowledge Discovery, pages 291– 316. Springer Netherlands. 1997
- [45] Charles Bouveron. Modélisation et classification des données de grande dimension : application à l'analyse d'images. Doctorat, Université Joseph-Fourier - Grenoble I, Français. 2006.
- [46] J. Davis and M. Goadrich. "The relationship between precision-recall and ROC curves". In Proceedings of the Twenty-Third International Conference on Machine Learning, Pittsburgh, USA, 2006.
- [47] T. Fawcett. ROC graphs: notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories, Palo Alto, USA, 2004.
- [48] L. A. Wald, "European proposal for terms of reference in data fusion", interbational archives of photogrammetry and Remote Sensing, Vol. XXXII, Part 7, page. 651-654,1998.
- [49] http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database.

Résumé

Ce mémoire concerne le sujet de la reconnaissance des formes de textures qui est un ensemble de techniques et méthodes visant à identifier des motifs dans images texturées. Pour ce faire ; deux étapes sont indispensables, la première est l'extraction des caractéristiques qui est effectué en utilisant les motifs binaires locaux LBP. Cette méthode transforme l'image de texture en un vecteur de caractéristiques en se basant sur l'information spatiale contenue dans le voisinage de chaque pixel de l'image. La deuxième étape est la classification qui a pour but d'évaluer la capacité des LBP à discriminer les différents types de textures dans les images en utilisant la méthode des k plus proche voisin KNN. L'évaluation des résultats s'estime par la validation croisée et la validation simple, qui sont des procédures qui structurent les itérations nécessaires à la validation des performances du classifieur utilisé.
