

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERY de Tizi-Ouzou



Faculté de Génie électrique et D'informatique
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes
de MASTER ACADEMIQUE**
Spécialité : **Commande des systèmes**

Présentés par
Aghiles GOUNANE
Samira SADI

Mémoire dirigée par M^r Youcef MESSAR

Thème

**Contribution de l'approche LBP à la
classification des images multispectrales.
Application aux données TM de la région
du Hoggar.**

Mémoire soutenu publiquement le 03 juillet 2014 devant le jury composé de :

M Moussa DIAF
Pr, UMMTO, Président

M Youcef MESSAR
MAA, UMMTO, Rapporteur

Mme Sadia ALKAMA
MCB, UMMTO, Examineur

Mme Fadhila BOUDJEMAA
MAA, UMMTO, Examineur

Remerciements

Nous tenons tout d'abord à remercier notre professeure et promoteur Mr : MESSAR Youcef, pour son précieux conseil et son aide durant toute la période du travail et aussi pour l'orientation, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant d'examiner notre mémoire Et de l'enrichir par leurs propositions.

Nous tenons aussi à exprimer nos remerciements, d'une façon particulière à M^{elle} NAIT BELAID pour son aide ainsi qu'à toute la documentation qu'il a mit à notre disposition.

Nous remercions toutes celles et ceux qui de près ou de loin, nous ont prêté main forte (Ne serai-ce que par une parole), pour la réalisation et la finalisation de ce projet

Dédicaces

Je dédie ce présent travail à;

*Mes parents pour leur énorme soutien financier et moral
et qui m'ont permis d'arriver là où j'en suis ce jour, je
voudrai leur exprimer toute ma gratitude et tout amour, je
leur dis mille fois merci.*

*Toute ma famille et mes amis ainsi que mes camarades sans
oublier Mon binôme GOUNANE Aghiles*

Sadi Samira

Dédicaces

Je dédie ce présent travail à;

Mes parents, pour leur énorme soutien financier et moral et qui m'ont permis d'arriver là où j'en suis ce jour, je voudrai leur exprimer toute ma gratitude et tout amour, je leur dis mille fois merci ;

Mon frère Moussa, et ma sœur Amina

Toute ma famille et mes amis ainsi que mes camarades sans oublier Mon binôme Sadi Smira.

Gounane Aghiles

SOMMAIRE

SOMMAIRE

Chapitre I : La texture

Introduction générale.....	1
I.1. introduction	3
I.2. Définition de la texture.....	3
I.3. Les méthodes de l'analyse.....	4
I.3.1. Les méthodes structurelles.....	4
I.3.2. Les méthodes basées sur modèles.....	5
I.3.3. Les méthodes fréquentielles.....	5
I.3.4. Les méthodes fractales.....	5
I.3.5. Méthodes statistiques.....	6
I.3.5.1. Attributs statistiques du premier ordre.....	6
I.3.5.2. Attributs statistiques d'ordre 2.....	7
I.4. Approche proposé.....	10
I.5. Conclusion.....	10

Chapitre II : Locale Binary Pattern(LBP)

II.1. Introduction.....	11
II.2. Définition.....	11
II.3. LBP de base.....	11
II.4. Le LBPp.r.....	13
II.5. Dérivation.....	15
II.6. LBP uniforme (LBP^{u2}).....	16
II.6.1. LBP invariant par rotation.....	18
II.6.2 uniforme invariant.....	21
II.6.3. Centre Symétrique de la configuration binaire locale (CS-LBP).....	21
II.6.4. Combinaison orthogonale de motifs binaires locaux (OC-LBP).....	22
II.6.5 Quelques autres opérateurs.....	23
II.7. Modèle textural proposé.....	24
II.8 Mesure de texture.....	24
II.9. Modèles LBP multivariables.....	25

II.10. Extension à 4 et à 5 bandes spectrales.....	27
II.11. Conclusion.....	28

Chapitre III : segmentation et classification

III.1.Introduction	29
III.2.Définition de la segmentation	29
III.3. Différentes méthodes de segmentation	29
III.3.1. Approche globale de l'image (pixellaire).....	30
III.3.2. Approche contour	31
III.3.3.Approche région	32
III.3.4. Approche coopérative.....	33
III.4. Définition de classification	34
III.5. Les approche de classification	35
III.5.1.Approche classification supervisé	35
III.5.1.1. Les k-plus-proches-voisins	35
III.5.1.2. La méthode du maximum de vraisemblance	36
III.5.2. Approche classification non supervisé.....	37
III.5.2.1. La méthode des k-means.....	37
III.5.2.2. La méthode ascendante Hiérarchique	39
III.6. Évaluation d'une classification.....	41
III.7.Conclusion	43

Chapitre IV Implémentation et Expérimentation

Partie 1 : Implémentation

IV.1.1. Le langage C en traitement d'images numériques	44
IV.1.2. Caractéristiques de base des images numériques	44
IV.1.3. Du fichier source à l'exécutable	45
IV.1.4.Microsoft visuel studio.....	45
IV.1.5.Presentation des algorithmes.....	46

Partie 2 : Expérimentation

IV.2.1. Présentation de la zone d'étude	52
IV.2.2. Méthodologie	52

IV.2.3. Sélection des zones d'entraînements (échantillons terrains) et Analyse des canaux bruts.....	54
IV.2.4. Mesure de texture	55
IV.2.5. LBP multiples.....	56
VI.2.6. Classification multispectrale supervisée	58
IV.2.7. Interprétation et discussion	60
IV.3. Conclusion	61
Conclusion générale	62
Referance Bibliographique.....	63

Introduction générale

Introduction générale

Une image numérique est composée d'unités élémentaires appelées pixels, contenant chacune différentes informations (intensité, lumineuse, couleur, etc.).

L'arrivée de ces images constitue un véritable atout dans beaucoup de domaines scientifiques.

En effet, les images numériques apportent un support de représentation permettant de modéliser et de véhiculer beaucoup d'informations, les rendant ainsi accessibles à l'interprétation et ce, pour une utilisation optimale dans le domaine socio-économique.

L'analyse d'une image consiste souvent à extraire un certain nombre de propriétés caractéristiques et à les exprimer sous forme paramétrique.

La texture joue un rôle très important dans l'étape de l'identification et de l'extraction des informations thématiques contenues dans une image et il semble que l'avenir de la maîtrise de l'image réside dans la compréhension plus approfondie de sa texture, donc dans sa définition plus complète et cohérente.

Dans l'analyse des textures, on trouve une méthode théoriquement simple mais très puissante avec un opérateur appelé motif binaire locale (LBP). L'opérateur de texture LBP est devenu une approche populaire dans diverses applications. Grâce à ses récentes extensions, il a été transformé en une très puissante mesure de la texture de l'image.

Ce travail concerne l'étude de différentes méthodes de traitements d'images et d'analyse de texture et essentiellement l'application de l'opérateur Local Binary Pattern(LBP).

Le présent travail est structuré en quatre chapitres.

Le premier chapitre présentera les notions et les types de texture, ainsi que les paramètres qui permettent de les quantifier. L'accent est mis sur les méthodes statistiques qui contribueront d'avantage à mieux comprendre l'objet des approches proposées.

Le deuxième chapitre sera spécialement consacré pour la présentation du descripteur LBP, en précisant la manière de calcul de cet opérateur dans les images en niveaux de gris, et comment il joue le rôle dans l'extraction des textures. Nous allons voir aussi quelques extensions du LBP les plus récentes.

Dans le troisième chapitre, nous présenterons les principales approches de la segmentation des images, en se basant essentiellement sur l'approche classification.

Le dernier chapitre est réservé aux tests et à la présentation des résultats obtenus sur quelques images par les approches proposées.

Enfin, nous terminerons notre travail par une conclusion générale, clôturant ce mémoire et quelques perspectives.

Chapitre I

I.1.Introduction

La texture est une caractéristique importante de la surface et de la structure interne d'un objet. Son analyse est essentielle dans l'interprétation automatique d'une scène. L'étude des textures rencontrées dans les images est un thème qui suscite beaucoup d'intérêt dans la communauté du traitement du signal et des images.

I.2. Définition de la texture

Donner une définition précise et universelle de la notion de texture est un problème ardu qui n'a toujours pas été résolu, nous proposons deux définitions :

1- la texture est un phénomène à deux dimensions : la première concerne la description d'éléments de base ou primitives (le motif) à partir desquels est formée la texture, la deuxième dimension est relative à la description de l'organisation spatiale de ces primitives [1] ;

2- la texture est décrite comme étant un ensemble de primitives de taille et de forme variables, présentant une organisation spéciale particulière [2].

Nous distinguons au fait, selon l'élément de base et les règles de placement, 2 types de textures :

Les 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. Dans ce cas, la texture présente un aspect régulier, sous formes de motifs répétitifs spatialement placés selon une règle précise (ex: peau de lézard, mur de brique) donc, une approche structurelle déterministe [3] ;

Les micro-textures (ou textures aléatoires) présentent un aspect plus chaotique et plus désorganisé, mais l'impression visuelle reste globalement homogène. Dans ce cas, la texture présente des primitives "microscopiques" distribuées de manière aléatoire (ex: sable, laine tissée, herbe etc.) d'où, une approche probabiliste cherchant à caractériser l'aspect anarchique et homogène.

Les Textures microscopiques (Textures mixtes) sont décrites comme un mélange entre de macro-texture et micro-texture. En effet, à l'intérieur de ce type de textures, nous pouvons trouver une répétition de motifs très similaires, mais pas identiques.

La figure I.1 illustre les 03 types de texture.

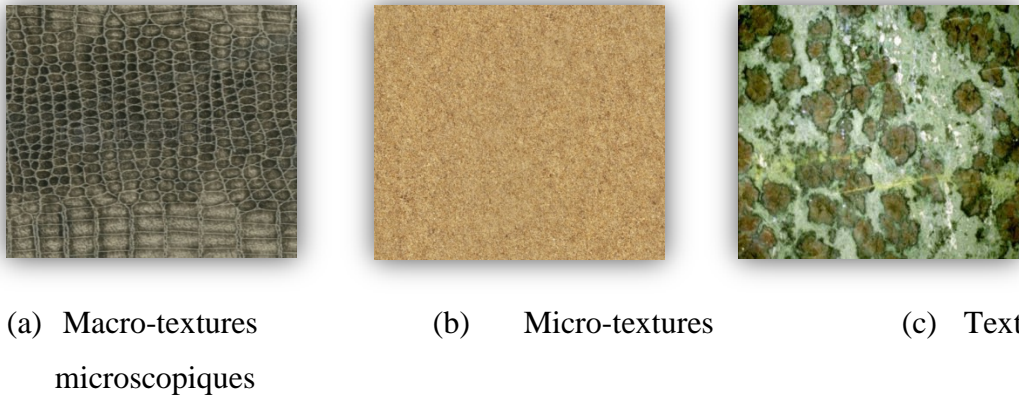


Figure I.1. : Exemple des types de textures.

I.3. Les méthodes de l'analyse

Dans une image, la texture est un paramètre très important pour la compréhension et l'interprétation d'une scène. C'est ainsi qu'elle est prise en compte dans nombre de méthodes d'analyse d'images. Si les méthodes permettant la mesure de textures sont nombreuses, aucune ne peut, aujourd'hui, prétendre généraliser un modèle de texture. En effet, de part sa complexité intrinsèque, l'on n'a pu trouver une définition formelle de ce qu'est la texture. On se contente donc de trouver un modèle adéquat pour l'étude à mener. C'est ainsi que l'on peut dresser une longue liste d'approches dans l'analyse de textures. Les différentes méthodes ont déjà été proposées dans la littérature et approuvées en pratique.

I.3.1. Les méthodes structurelles

On considère que les textures sont constituées de motifs de base se répétant dans des directions de l'espace de façon quasi-régulière. Dans les images déjà étudiées, les méthodes structurelles tiennent compte de l'information structurelle et contextuelle d'une forme et sont particulièrement bien adaptées aux textures macroscopiques. Les étapes d'analyse sont d'abord l'identification des éléments constitutifs, puis la définition des règles de placement. Les deux structures les plus importantes sont les structures de graphe et les structures syntaxiques [3].

I.3.2. Les méthodes basées sur modèles

Pour ces approches, les descripteurs sont les paramètres d'un modèle permettant de décrire ou de générer une texture. Plusieurs modèles peuvent être utilisés. Les plus connus sont les modèles Markoviens [4], les modèles autorégressifs et les modèles fractals [5]. L'estimation des paramètres de ces modèles permet de caractériser les textures.

I.3.3. Les méthodes fréquentielles

Ces méthodes, souvent utilisées en traitement du signal, permettent d'analyser une texture en identifiant les différentes fréquences qui la composent. Parmi elles, on retrouve notamment la transformée de Fourier, les filtres de Gabor ainsi que les ondelettes. De façon générale, la λ transformation linéaire d'une image f est obtenue en calculant la corrélation de f et d'une fonction ψ_λ .

$$\int_{-\infty}^{+\infty} f(x, y) \psi_\lambda^*(x, y) dx dy \quad (I.1)$$

Avec λ un ensemble de paramètres propre à ψ et $(^*)$ le complexe conjugué de la quantité en argument. Ces méthodes préservent à la fois les informations globales et locales. Elles sont bien adaptées aux signaux quasi périodiques. En effet, les textures sont des signaux quasi périodiques qui ont une énergie fréquentielle localisée. Ces méthodes permettent de caractériser la texture à différentes échelles.

Analyser une texture de part son spectre fréquentiel, nous donne beaucoup d'informations sur celle-ci.

I.3.4. Les méthodes fractales

Ces méthodes permettent de synthétiser des images très proches de la réalité. En analyse de texture, la dimension fractale, qui est une mesure du degré d'irrégularité d'un objet, décrit une certaine propriété de la texture. Le modèle fractal est basé essentiellement sur l'estimation par des méthodes spatiales de la dimension fractale de la surface représentant les niveaux de gris de l'image [6].

I.3.5. Méthodes statistiques

Les méthodes statistiques étudient les relations entre un pixel et ses voisins et définissent des paramètres discriminants de la texture en se basant sur des outils statistiques. Généralement, elles sont utilisées pour caractériser des structures fines, sans régularité apparente. Plus l'ordre du paramètre statistique est élevé et plus le nombre de pixels mis en jeu est important. Suivant la modalité des images à étudier, la signature la plus discriminante de la texture est à rechercher, soit dans des méthodes qui exploitent directement les propriétés statistiques de la texture (matrices de cooccurrences, matrice de longueurs de plages, matrice de voisinage, fonction d'auto-corrélation, modèle de Markov, modèle autorégressif, modèles issues de la morphologie mathématique), soit dans des méthodes qui exploitent les propriétés statistiques à partir d'un plan transformé dans lequel on réécrit l'image de texture. Dans ce dernier cas, il s'agit particulièrement de la technique de la densité spectrale, de la méthode des extrema locaux, de la transformation de Fourier et de l'approche de Karhunen Loeve [7] [8] [5].

I.3.5.1. Attributs statistiques du premier ordre

L'analyse de texture par les méthodes de premier ordre se fait au niveau de pixels individuels d'une région de l'image à traiter. Les paramètres tels que la moyenne, la variance, l'entropie, etc. sont généralement calculés à partir de l'histogramme des intensités (Attributs statistiques du premier ordre).

Les attributs les plus courants sont :

- **La moyenne**

La valeur moyenne $M1$ des niveaux de gris de tous les pixels de la région d'étude est donnée par :

$$M1 = \frac{1}{N * M} \sum_{x=1}^M \sum_{y=1}^N I(x,y) \quad (I.2)$$

$I(x,y)$ représente la valeur du niveau de gris du pixel (x,y) , M et N la taille de l'image.

Une image claire possède une moyenne plus élevée qu'une image foncée

- **L'écart-type**

L'écart-type indique comment, en moyenne, les valeurs des niveaux de gris sont groupées autour de la tendance centrale (moyenne arithmétique). Un faible écart type signifie que les

valeurs sont peu dispersées autour de la moyenne (série homogène), et inversement (série hétérogène).

$$\sigma = \frac{1}{N * M} \sum_{x=1}^M \sum_{y=1}^N I(x, y) - M1 \quad (I.3)$$

- **La variance**

Elle correspond au moment d'ordre 2. Elle mesure la répartition des niveaux de gris (luminosité) autour de la valeur moyenne. Sa valeur est donnée par :

$$\sigma^2 = \frac{1}{N * M} \sum_{x=1}^M \sum_{y=1}^N (I(x, y) - M1)^2 \quad (I.4)$$

I.3.5.2. Attributs statistiques d'ordre 2

Ce type de méthodes prend en considération les interactions entre les niveaux de gris de deux pixels. On retiendra, en particulier la méthode des matrices de cooccurrence [1] connues sous le nom GLCM (Gray Level Cooccurrence Matrix). Du fait de leur richesse en information de texture, les matrices de cooccurrences sont devenues les plus connues et les plus utilisées pour extraire les caractéristiques des textures. Elles servent de référence à toute nouvelle méthode de caractérisation des textures [9].

Une matrice de cooccurrence est une matrice de taille $N \times N$, ou N est le nombre de niveaux de gris de l'image.

Pour un déplacement d (translations), un élément (i, j) de la matrice est définie par le nombre de pixels de l'image de niveau de gris j situés à une distance d d'un pixel de niveau de gris i .

Formellement, la formule suivante définit une matrice de cooccurrence :

$$Pd(i, j) = |\{(r, s), (t, v) : I(r, s) = i, I(t, v) = j\}| \quad (I.5)$$

Où :

- (r, s) et (t, v) sont des coordonnées de l'image avec $(t, v) = (r + dx, s + dy)$;
- d est un vecteur de déplacement (dx, dy) ;
- $||$ désigne le cardinal de l'ensemble.

Voici par exemple deux matrices de cooccurrence extraite d'une même image

I à trois niveaux de gris pour deux valeurs de translation :

$$I = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

Pour $d = (1, 0)$, on obtient la matrice 3×3 suivante :

$$M_{cc} = \begin{bmatrix} 4 & 0 & 2 \\ 2 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Et pour $d = (1, 1)$, on obtient la matrice 3×3 suivante :

$$M_{cc} = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Les matrices de cooccurrence contiennent beaucoup d'éléments (elles sont de taille 256×256 éléments pour une image sur 256 niveaux de gris) et sont donc difficilement manipulables. Il est à noter d'ailleurs que dans la pratique, les images sont souvent altérées afin de ne considérer qu'une partie des niveaux de gris (8, 16, 32 etc.).

Afin d'extraire des attributs texturaux de ces matrices, Quatorze indices prenant en compte la totalité d'une matrice ont été définis par Haralick .

Voici quelques exemples d'attributs donnés par Haralick :

- **La Moyenne :**

$$Moy = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} Pd(i, j) \quad (I.6)$$

$Pd(i, j)$: correspond aux éléments de la matrice de cooccurrence, c'est-à-dire à la probabilité de passer d'un pixel de niveaux de gris i à un pixel de niveau de gris j .

Ng : correspond au maximum des niveaux de gris de l'image.

- **La Variance :**

Ce paramètre mesure la distribution des niveaux de gris autour de la moyenne.

$$VAR = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i - MOY)^2 Pd(i, j) \quad (I.7)$$

- **Le Contraste:**

Il favorise les variations locales des niveaux de gris. Il est élevé quand on passe d'un pixel très clair à un pixel très foncé ou inversement (fortes transitions).

$$CON = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i - j)^2 Pd(i, j) \quad (I.8)$$

- **L'Energie:**

Ce paramètre mesure l'uniformité de la texture il atteint de fortes valeurs lorsque la distribution des niveaux de gris est constante (on trouve souvent les mêmes couples de pixels).

$$E = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} Pd(i, j)^2 \quad (I.9)$$

- **L'Entropie:**

Ce paramètre mesure le désordre dans l'image. Contrairement à l'énergie, l'entropie atteint de fortes valeurs lorsque la texture est complètement aléatoire (sans structure apparente)

$$Ent = - \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \text{Log}Pd(i, j)^2 Pd(i, j) \quad (I.10)$$

I.4. Approche proposé

On propose une approche qui peut être classée dans les méthodes statistiques du fait qu'elle s'appuie essentiellement sur la répartition spatiale des niveaux de gris dans la texture considérée. Grâce à son pouvoir discriminant et la simplicité de calcul, l'opérateur motif binaire local appelé communément LBP (« Local Binary Pattern »), est devenu une approche populaire dans diverses applications. Il peut être considéré comme une approche unificatrice pour les modèles statistiques et structurelles traditionnellement divergents de l'analyse de texture. La propriété la plus importante de l'opérateur de LBP dans les applications du monde réel est sa robustesse aux changements d'échelle de gris monotones causés, par exemple, par des variations d'éclairage. Une autre propriété importante est sa simplicité de calcul, ce qui permet d'analyser des images dans les milieux difficiles en temps réel.

L'opérateur LBP propose d'assigner un code binaire à un pixel en fonction de son voisinage, La valeur d'intensité d'un pixel central est utilisée pour seuiller ses pixels voisins afin de générer un motif binaire. Les pixels de ce motif binaire sont alors multipliés par des poids et sommés afin d'obtenir un code LBP.

I.5.Conclusion

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. Les méthodes d'analyse de la texture sont très nombreuses et très variés. Elles ont pour but de caractériser une image ou un pixel par un ensemble d'attributs ou signatures, la texture est ainsi définie et étudiée de manières différentes, en fonction de la façon dont on la caractérise. Parmi toutes ces méthodes, nous nous sommes focalisés sur les méthodes statistiques et on a proposé l'approche « Local Binary Patten » qui forme l'objet du chapitre suivant.

Chapitre II

II.1. Introduction

La texture est une composante riche en information dans l'interprétation d'une image. Nous proposons dans ce chapitre, l'étude d'une approche qui semble être la mesure la plus supérieure à la plupart des mesures de quantification traditionnelle. Il s'agit d'une approche à la fois efficace, simple à mettre en œuvre et invariante par rotation. Il s'agit du motif binaire local (LBP) qui est devenu une approche populaire dans plusieurs applications comme la télécommande, la télédétection, l'analyse de l'image biomédicale, la biométrie, l'analyse visuelle, la détection du mouvement etc...

Nous nous concentrons particulièrement dans cette étude, aux améliorations et aux extensions de cet opérateur.

II.2. Définition

Ce descripteur a été d'abord mentionné pour la première fois par Harwood en 1993 pour mesurer le contraste local d'une image monospectrale, mais réellement popularisé en 1996 par Timo Ojala pour analyser les textures [10]. Il a montré d'excellentes performances dans de nombreuses études comparatives, tant en termes de vitesse qu'en termes de discrimination des différentes textures.

Le LBP se définit aussi comme une caractéristique qui résume la structure spatiale locale d'une image utilisée en vision par ordinateur, pour reconnaître des textures ou pour détecter des objets dans les images numériques [11].

II.3. LBP de base

Le concept du LBP est le suivant : un code binaire décrivant la texture locale d'une région est calculé par seuillage d'un voisinage avec le niveau de gris du pixel central. Tous les voisins prendront alors une valeur 1 si leur valeur est supérieure ou égale au pixel courant et 0 sinon. On va alors multiplier cette matrice composée de 0 et 1 par les poids LBP et sommer tous ses éléments pour obtenir la valeur LBP du pixel courant [12].

On obtiendra donc des pixels dont l'intensité se situe entre 0 et 255 comme dans une image 8 bits ordinaire (Figure. II.1).

6	5	2	1	0	0	1	2	4			
7	6	1	1		0	128		8		241	
13	28	11	1	1	1	64	32	16			
image brute			image seuillée			poids			LBP		

Code binaire : 11110001 LBP=128+64+32+16+1=241

Figure II.1 : Illustration de calcul d'un LBP de base (8 voisins)

Le LBP de base est défini par :

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c) \times 2^p \quad (\text{II.1})$$

Où : g_c est le niveau de gris du pixel central de coordonnées (x_c, y_c) .

g_p ($p = 0, 1, \dots, 7$) est le niveau de gris de chaque pixel voisin.

Avec : $S(x)$ une fonction définie comme suit :

$$S(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Histogramme

Souvent la distribution des codes LBP sur l'image est utilisée pour décrire la texture sous forme d'histogramme.

Une fois le code LBP est calculé pour tous les pixels de l'image, on calcule l'histogramme de cette image LBP pour former un vecteur de caractéristiques représentant l'image.

En réalité, afin d'incorporer plus d'informations spatiales au vecteur représentant l'image (par exemple le visage), on divise tout d'abord cette image codée par l'opérateur LBP en petites régions et l'histogramme sera construit pour chaque région. Finalement, on concatène tous les histogrammes des régions afin de former un grand histogramme représentant l'image des caractéristiques faciales (voir la figure ci-après). L'efficacité du code LBP comme indice facial s'explique par le fait que le LBP permet de caractériser les détails fins d'un visage

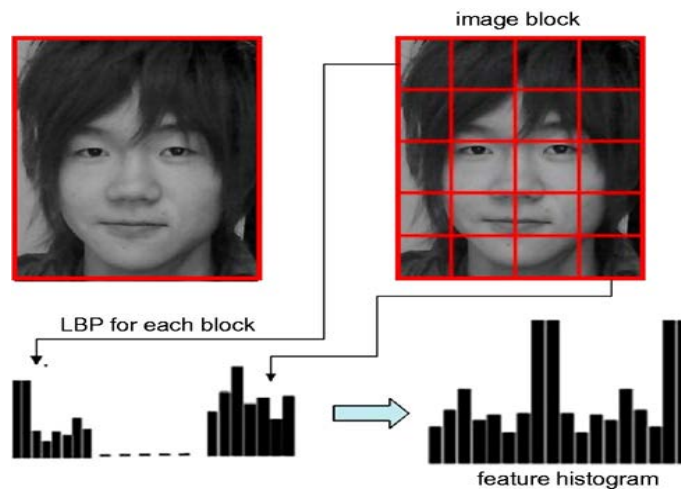


Figure II.2. : Exemple d'un histogramme LBP d'une image faciale.

L'histogramme LBP est connu sous cette forme :

$$h(i) = \sum_{x,y} B(LBP(x,y) = i) \quad (II.2)$$

Avec : $i \in [0, \dots, 2^p - 1]$ et $B(v) = \begin{cases} 1, & \text{lorsque } v \text{ est vraie} \\ 0, & \text{autrement} \end{cases}$

II.4. Le LBP_{P,R}

A l'origine, les codes de LBP ont été proposés seulement pour des régions 3 x 3. Ils ont été limités à de petites régions et ne peuvent capturer que des micro-textures locales. Ainsi, le LBP a été étendu à LBP (P, R) et cela en utilisant des voisinages de taille différente. Dans ce cas, un cercle de rayon R autour du pixel central est choisi. Les valeurs des P points échantillonnés sur le bord de ce cercle sont prises et comparées avec la valeur du pixel central. Pour obtenir les valeurs des P points échantillonnés dans le voisinage pour tout rayon R, une interpolation est nécessaire

Dans ce qui suit, la notation (P, R) définit le voisinage de P points sur un rayon R d'un pixel (Figure II.3).

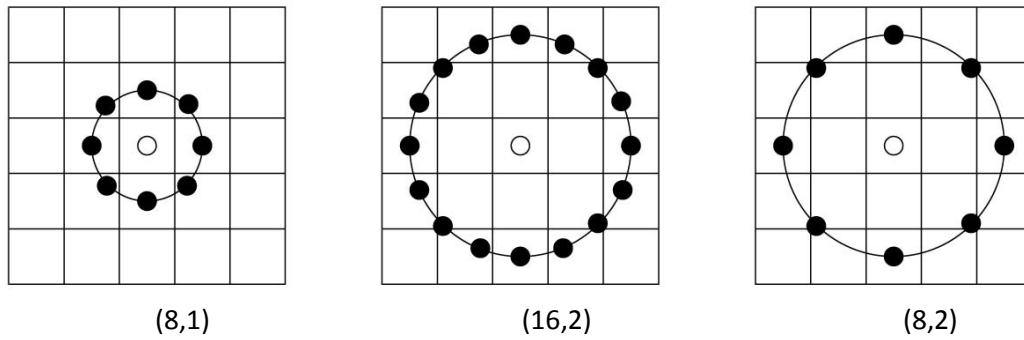


Figure II.3. : Exemples de voisinages avec différentes valeurs de (P, R).

Pour calculer un LBP dans un voisinage de P pixels, dans un rayon R, on compte simplement les occurrences de niveaux de gris plus grands que la valeur centrale.

$$LBP_{P,R} = \sum_{p=0}^{P-1} S(g_p - g_c) 2^p \quad (II. 3)$$

Où $s(x)$ est la fonction signe :

$$s(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

g_p et g_c sont respectivement les niveaux de gris d'un pixel voisin et du pixel central.

On peut illustrer la méthode calculant le $LBP_{P,R}$ par la figure suivante :

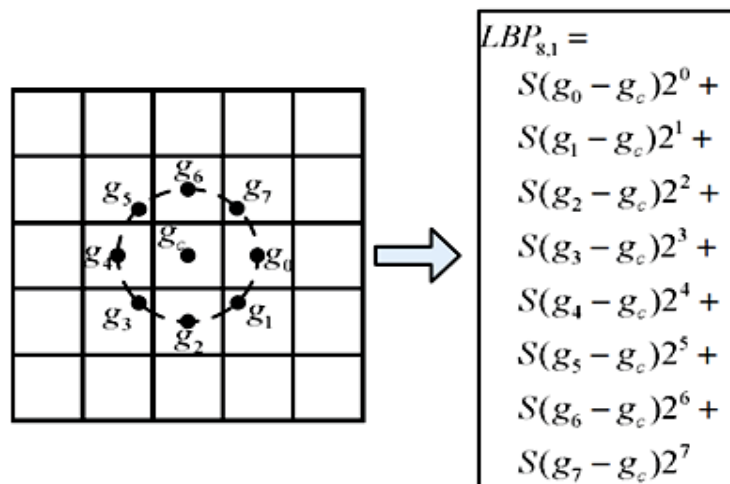


Figure II.4. : Calcul d'un $LBP_{P,R}$ (P=8, R=1).

II.5. Dérivation

En raison de l'absence d'une définition universellement acceptée de la texture, nous définissons donc la texture T dans le voisinage de rayon R d'une image (en niveaux de gris) comme la distribution jointe de $P + 1$ pixels:

La dérivation du LBP est représentée en 2002 par Ojala et al [13], en se basant sur le principe de la texture :

Soit une fenêtre de texture T d'une image et t l'ensemble des pixels dans T pour le calcul du LBP :

$$T = t(g_c, g_0, \dots, g_{P-1}) \quad (II.4)$$

g_c est le niveau de gris du pixel central de T , et $g_0 \dots g_{P-1}$ les niveaux de gris des pixels voisins.

Ainsi définie, la texture peut se représenter, sans perte d'informations par les différences de niveaux de gris entre le pourtour et le centre :

$$T = t(g_c, g_0 - g_c, \dots, g_{P-1} - g_c) \quad (II.5)$$

Si on suppose que ces différences sont localement indépendantes de la valeur du pixel central, on aura :

$$T = t(g_c)t(g_0 - g_c, \dots, g_{P-1} - g_c) \quad (II.6)$$

Dans la pratique, l'hypothèse de l'indépendance n'est pas toujours vraie.

On trouve que $t(g_c)$ n'est qu'une description de luminance d'un pixel de l'image et ne fournit pas d'informations nécessaires pour analyser la texture [14]. Donc, la plupart d'informations caractéristiques de la texture sont préservées dans la distribution conjointe des différences :

$$T \approx t(g_0 - g_c, \dots, g_{P-1} - g_c) \quad (II.7)$$

Cette distribution de dimension P reflète les différences locales des niveaux de gris. Afin de renforcer l'invariance à des changements monotones des niveaux de gris, seul le signe de ces différences est pris en compte. Donc, la distribution devient :

$$T \approx t(\delta(g_0 - g_c), \dots, \delta(g_{P-1} - g_c)) \quad (II.8)$$

Où $\delta(\cdot)$ est une fonction échelon définie comme suit :

$$\delta(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Donc le code LBP caractérisant la texture locale autour d'un pixel (x_c, y_c) de l'image est :

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p \delta(g_p - g_c) \quad (II.9)$$

Ce motif est calculé en chaque point où le voisinage peut être défini, ce qui résulte en un descripteur global décrivant la texture de l'image. La distribution est représentée maintenant par un seul code LBP :

$$T \approx t(LBP_{P,R}(x_c, y_c)) \quad (II.10)$$

Remarque sur le voisinage

Les coordonnées des points p ($p = 0, \dots, P - 1$) du voisinage circulaire de rayon R autour du point (x_c, y_c) sont données par :

$$x_c + R \cos(2p\pi/P), y_c - R \sin(2p\pi/P) \quad (II.11)$$

La valeur du niveau de gris d'un point du voisinage qui ne «tombe» pas exactement sur un pixel réel est calculée par interpolation à la valeur du pixel le plus proche

II.6. LBP uniforme (LBP^{u2})

Une autre extension à l'opérateur d'origine est la définition de ce qu'on appelle les modèles uniformes noté LBP^{u2} [15]. Le terme uniforme se réfère à l'aspect uniforme de la configuration binaire locale. Cette extension a été inspirée par le fait que 90% des motifs rencontrés dans des images sont uniformes. Ces derniers contiennent au plus deux transitions de bits de 0 à 1 ou vice-versa quand la chaîne binaire est considérée comme circulaire. Par exemple les modèles de 00000000 (0 transitions), 01110000 (2 transitions) et 11001111 (2 transitions) sont des codes uniformes, alors que les modèles 11001001 (4 transitions) et 01010010 (6 transitions) ne sont pas.

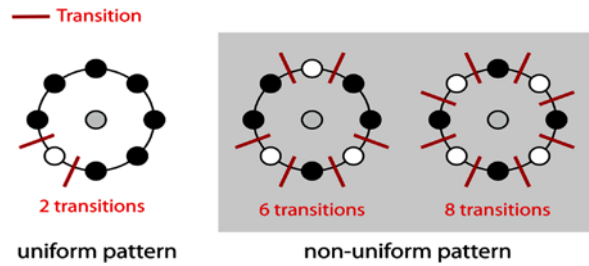


Figure II.5.: Exemples de motifs uniformes et non-uniformes

Le LBP^{u2} permet de détecter uniquement les textures locales importantes (ou les primitives de texture) comme les spots, les fins de ligne, les bords et les coins comme le montre la Figure II.6.

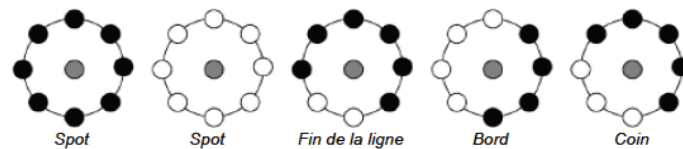


Figure II.6. : Textures particulières détectées par LBP^{u2}.

A l'origine, Ojala et al. [13] ont utilisé l'ensemble des 256 valeurs obtenues par seuillage des 8 voisins du pixel central 2^k .

Dans le calcul des codes LBP, chaque motif uniforme a son propre code de sortie et tous les motifs non-uniformes sont affectés à une seul code.

Pour une chaîne binaire de longueur k il y a $k(k-1)+2$ modèles uniformes, ce qui fait **58** pour une chaîne de 8 bits. En prenant un bac supplémentaire pour l'ensemble de ceux non uniformes, nous comptabilisons un total de $k(k-1)+3$, soit **59** valeurs au lieu de **256** et la figure suivante illustre les 58 modèles uniforme pour le quartier (8, R)

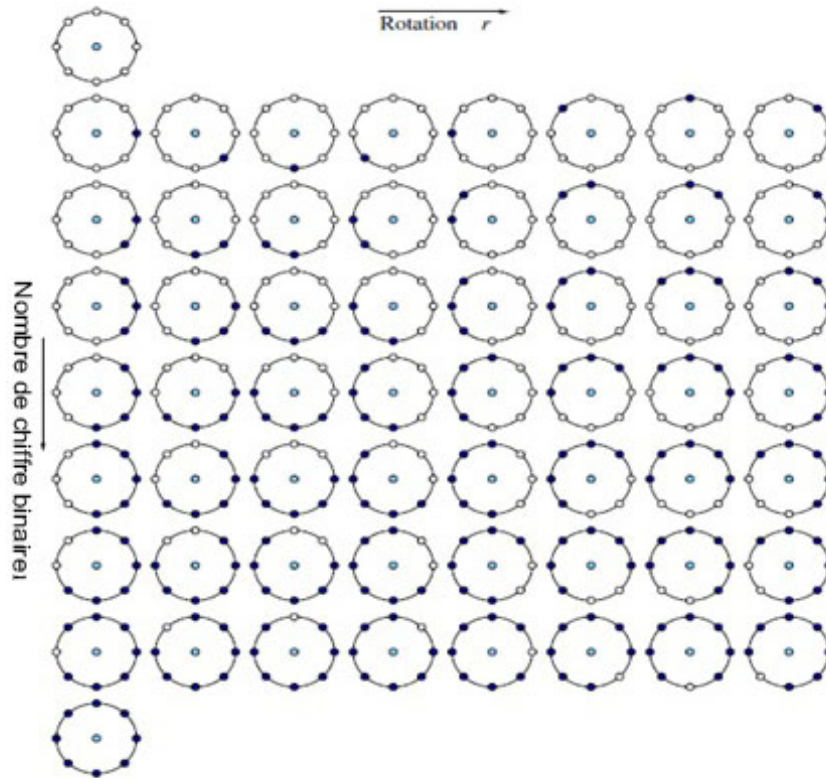


Figure II.7. Les 58 différents modèles uniformes du quartier (8, R).

II.6.1. LBP invariant par rotation

La rotation du code LBP est une méthode pour rendre les codes LBP plus invariant à l'orientation de l'image et pour faire une identification beaucoup plus efficace de la texture.

L'image d'entrée tourne d'un angle α dans un sens, les pixels voisins tournent autour du pixel central dans le sens contraire comme le montre la figure II.8:

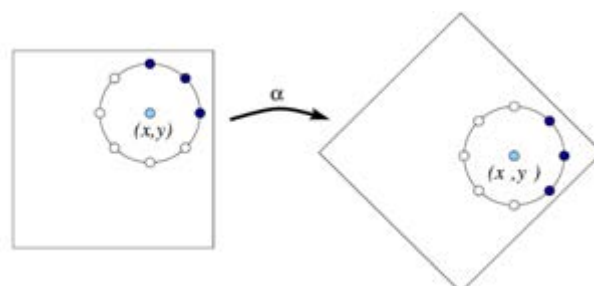


Figure II.8. : La rotation de l'image

On désigne par $I^{\alpha}(x, y)$, la rotation de l'image $I(x, y)$ par α degrés.

Les valeurs de gris des pixels voisins g_p sont mises en rotation autour du pixel central g_c . Cet effet de rotation cause une variation du code $LBP_{P,R}$ parce que les poids binomiaux 2^p affectés aux g_p lors du calcul du code sont différents, c.à.d., par exemple, au lieu d'affecter le poids 2^3 pour g_3 , on l'affecte pour g_2 à cause de la rotation.

Pour supprimer l'effet de la rotation, on définit le code LBP invariant à la rotation, notée $LBP_{P,R}^{ri}$ qui est définie en vue de retourner une valeur unique (valeur minimale du code $LBP_{P,R}$) quelle que soit la rotation appliquée [14] :

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \text{ avec } i = 0, 1, \dots, P - 1\} \quad (\text{II. 12})$$

L'exposant ri signifie «invariance par rotation».

Et $ROR(x, i)$ correspond au résultat de i décalages circulaires successifs ($|i| < P$) vers la droite (sens horaire) des bits du nombre binaire x (codé sur P bits) qui correspond à la valeur du $LBP_{P,R}$.

En bref, le code invariant à la rotation se produit en tournant le code original jusqu'à ce que sa valeur minimale soit atteinte.

Avec cette méthode, 36 motifs différents sont considérés au lieu de 256 ($P=8$).

L'avantage est que la texture sera plus efficace en réduisant le nombre des LBP.

La figure ci-dessous illustre les 36 uniques motifs binaires locaux invariants en rotation possibles dans le cas de $P = 8$. Il faut comprendre que la représentation suivante ne signifie pas un voisinage circulaire des niveaux de gris autour d'un pixel, mais c'est une représentation binaire circulaire du code binaire local (LBP) codé sur $P=8$ bits de chaque pixel central entouré par 8 voisins ($P=8$).

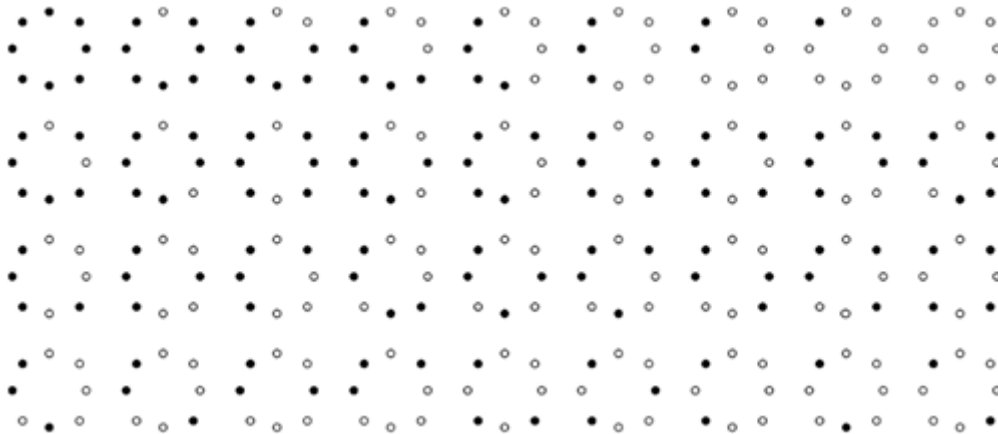


Figure II.9 : Liste des 36 formes binaires locales invariantes en rotation de $LBP_{8,R}^i$.

A titre d'exemple, le motif 0 permet de détecter les tâches claires, le motif 8 les tâches sombres et le motif 4 les contours.

Si le code est à 8 bits, la rotation se fait de 45° à chaque quart de travail.

En pratique, il est difficile de distinguer ces 36 formes lorsqu'une rotation est appliquée sur des images.

Etude d'un exemple

La Figure suivante illustre six rotations invariantes des codes dans la rangée du haut :

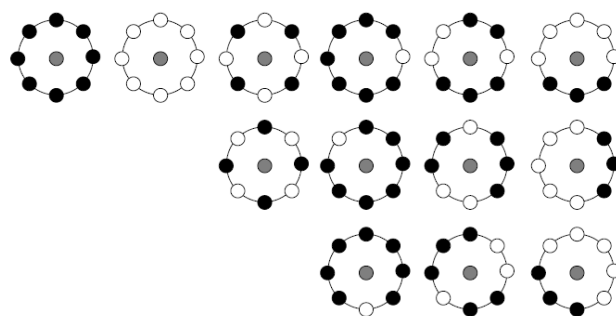


Figure II.10. : Exemple des rotations invariantes des codes LBP.

Les deux premiers quartiers sont immuables par rotation, et il n'y a pas par conséquent, d'autres quartiers qui pourraient produire les mêmes codes. Le troisième est particulier, il n'a que deux versions différentes de rotation. Les trois autres quartiers ont un total de huit versions différentes de rotation, dont deux sont présentées à titre d'exemples.

II.6.2 uniforme invariant

Le LBP invariant par rotation $\mathbf{LBP}_{p,R}^{ri}$ est souvent combiné avec le LBP uniforme \mathbf{LBP}^{u2} . Cela nous ramène à un opérateur à rotation invariant et uniforme notée $\mathbf{LBP}_{p,R}^{riu2}$ et défini comme suit :

$$\mathbf{LBP}_{p,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} S(g_p - g_c) & \text{si } U(\mathbf{LBP}_{p,R}) \leq 2 \\ P + 1 & \text{sinon} \end{cases} \quad (II.13)$$

Le code LBP invariant par rotation correspond à la somme des bits du voisinage s'il est uniforme. Tous les codes qui ne sont pas classés comme uniforme, l'opérateur les affecte à la valeur P+1 [15].

Dans la pratique, l'opérateur $\mathbf{LBP}_{p,R}^{riu2}$ est mieux mis en œuvre en créant une table de correspondance qui convertit les codes LBP «de base» dans leurs $\mathbf{LBP}_{p,R}^{riu2}$ correspondants.

Le point faible principal de $\mathbf{LBP}_{p,R}^{riu2}$, c'est qu'il ne peut pas décrire les caractéristiques des textures efficacement et complètement car le $\mathbf{LBP}_{p,R}^{riu2}$ fusionne tout les motifs non uniformes directement dans un modèle

II.6.3. Centre Symétrique de la configuration binaire locale (CS-LBP)

Le CS-LBP est une autre version modifiée de LBP. Il a été développé par Marko Heikkila [16] pour la reconnaissance des objets de base de données PASCAL.

Le LBP initial était très long, sa fonction n'est pas solide sur les images plates. Dans cette méthode, au lieu de comparer la valeur de niveau de gris de chaque pixel avec le centre pixel, les centres pairs symétriques de pixels sont comparés (Figure II.11).

CS-LBP présente plusieurs avantages comme la tolérance aux changements d'éclairage, la robustesse des zones de l'image et l'efficacité de calcul [17].

Les caractéristiques CS-LBP sont calculées par l'équation:

$$CS\ LBP_{P,r,T} = \sum_{i=0}^{\frac{N}{2}-1} S\left(g_i - g_{i+\frac{N}{2}}\right) 2^i, \quad (II.14)$$

Avec :

$$S(x) = \begin{cases} 1 & \text{si } x \geq T. \\ 0 & \text{autrement.} \end{cases}$$

g_i et $g_{i+\frac{N}{2}}$ correspondent au niveau de gris du centre-symétrique des paires des pixels

(N au total) régulièrement espacés sur un cercle de rayon r . T est une valeur faible utilisée comme seuil.

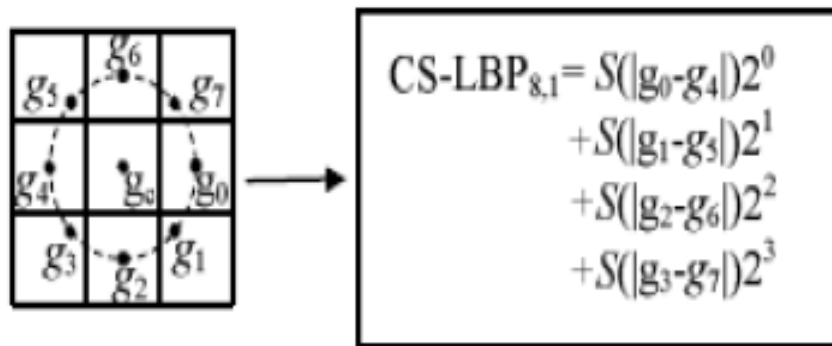


Figure II.11. CS-LBP caractéristique d'un quartier de 8 pixels

II.6.4. Combinaison orthogonale de motifs binaires locaux (OC-LBP)

OC-LBP propose une combinaison orthogonale de motifs binaires locaux, ce qui réduit considérablement le nombre de dimensions de l'histogramme LBP originale tout en gardant son pouvoir discriminant [18].

La Figure (II.12) illustre le processus d'un opérateur OC-LBP avec huit pixels voisins. Dans ce cas, deux opérateurs de LBP 4 voisins réguliers sont considérés. Le premier traite les voisins horizontaux et verticaux et le second est constitué des voisins diagonaux. En concaténant ces deux histogrammes LBP, nous obtenons l'histogramme OC-LBP avec 32 niveaux, ce qui est huit fois plus compact que l'original 8-voisin LBP histogramme (256 niveaux).

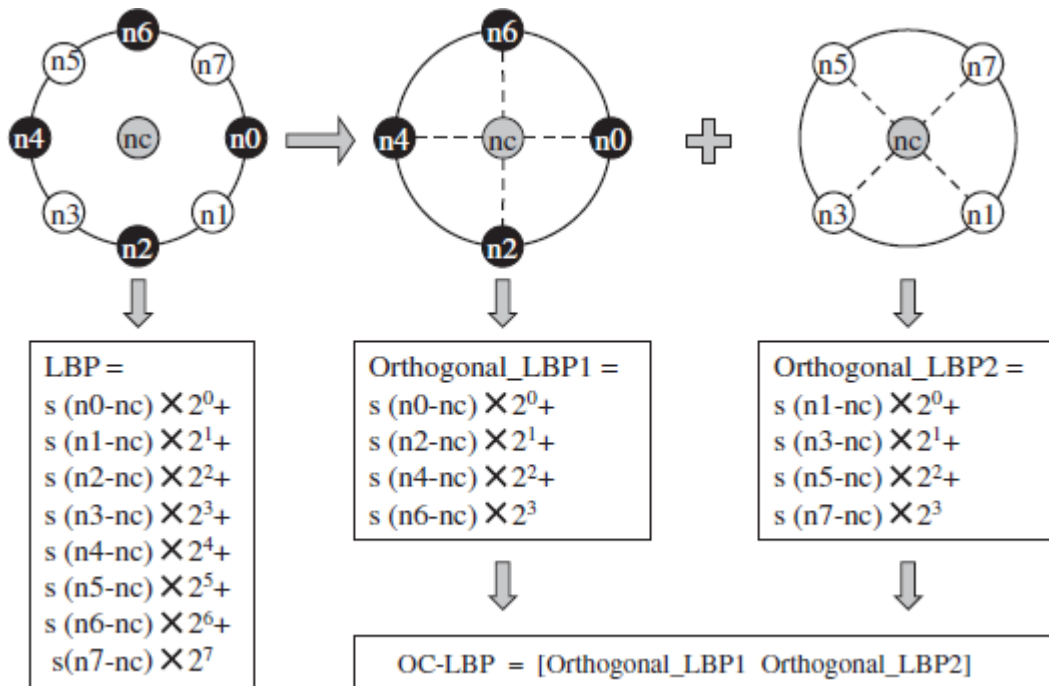


Figure II.12. Le processus d'un opérateur OC-LBP

II.6.5 Quelques autres opérateurs

Pour la reconnaissance des visages, d'autres opérateurs ont donné de bons résultats, nous pouvons citer :

- Le motif local quantifié **LQP** (Local Quantized Patterns) : Récemment Hussain et al [19] ont introduit les motifs locaux quantifiés. Ceux-ci permettent d'utiliser les LBP ou les LTP sur des voisinages plus complexes, c'est à dire des voisinages constitués de plusieurs cercles concentriques et avec un plus grand nombre d'échantillons.
- Le motif ternaire local **LTP** (Local Ternary Pattern) : une extension introduite en 2010 par Tan et Triggs [20] pour sa plus grande résistance au bruit. Au lieu de binariser les valeurs échantillonnées, celle-ci peut prendre trois valeurs selon leur distance à la valeur du pixel central
- Amélioration LBP (**ILBP**) : introduit par Jin et al [21], la principale différence entre ILBP et LBP réside dans la comparaison de tous les pixels (incluant le pixel central) avec la moyenne de tous les pixels dans le noyau pour l'ILBP.

II.7. Modèle textural proposé

Ojala et al [13] ont de même constaté que tous les motifs binaires locaux ne décrivent pas bien les bonnes propriétés d'une texture. LBP capte l'uniformité du pixel central à l'égard de son voisinage, mais il ne capte pas l'uniformité de la région elle-même. Par conséquent, ils ont introduit une mesure d'uniformité U dans un voisinage. U correspond au nombre de transitions spatiales 0/1 dans le motif. Avec $g_p = g_0$, l'uniformité est décrite par :

$$U_c = \sum_{i=1}^p |\text{sign}(g_i - g_c) - \text{sign}(g_{i-1} - g_c)| \quad (II.15).$$

Les formes avec $U_c \leq j$ sont désignées comme uniformes. D'après les études déjà réalisées [14], $j = 2$ correspond au meilleur modèle textural. Cela se traduit par l'opérateur suivant, qui offre une description texturale invariante par rotation.

$$LBP_{c,j} = \begin{cases} \sum_{i=0}^{p-1} \text{sign}(g_i - g_c) & \text{si } U_c \leq j \\ p + 1 & \text{sinon} \end{cases} \quad (II.16)$$

L'opérateur $LBP_{c,j}$ seuille les pixels dans un voisinage de P équidistants pixels sur un rayon R , à la valeur du pixel central. Il permet de détecter les motifs uniformes de toute quantification de l'espace angulaire et pour toute résolution spatiale. Les motifs non uniformes sont regroupés sous une seule étiquette $P+1$.

II. 8 Mesure de texture

Le $LBP_{c,j}$ mesure la structure spatiale de la texture locale de l'image mais, n'étudie pas le contraste, donnant une autre propriété importante de la texture locale de l'image. Dans la plupart des cas, les performances peuvent être améliorées en le combinant avec mesure de variance invariant en rotation qui caractérise le contraste de la texture de l'image définie par :

$$VAR_c = \sum_{i=0}^{p-1} (g_i - \mu_c)^2 \quad AVEC \quad \mu_c = \frac{1}{p} \sum_{i=0}^{p-1} g_i \quad (II.17)$$

Les deux valeurs $LBP_{c,j}$ et VAR_c sont calculées et affectées à chaque pixel d'image individuellement, illustrant ainsi des informations de texture locale. Par conséquent, deux nouvelles images sont dérivées de l'image originale. Ces dernières forment la base pour la mesure finale de texture.

La plupart des méthodes d'analyse de la texture consiste à quantifier les mesures de texture par des valeurs simples (moyennes, écarts, entropie, etc.). Cependant, beaucoup d'informations importantes contenues dans les distributions des valeurs de caractéristiques peuvent être perdues. Dans cette étude, la dernière caractéristique de texture est l'analyse simultanée des deux variables obtenues.

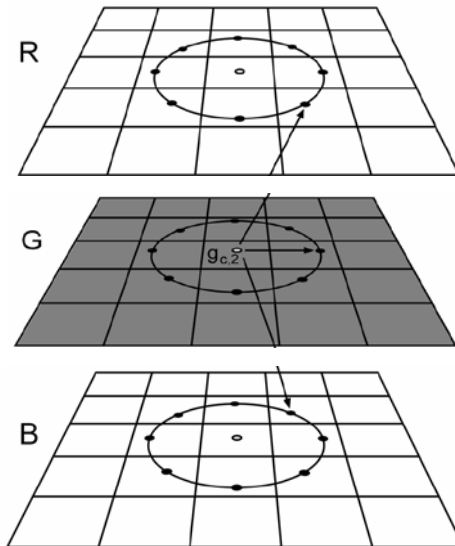
II.9. Modèles LBP multivariables

Grace à ses extensions récentes, l'opérateur LBP est devenu une très puissante mesure de la texture des images, voici quelques extensions les plus récentes.

➤ L'opérateur binaire local multi-varié (**MLBP_c**):

Poirson et al [15] ont montré que les informations de couleur et de motif sont traitées séparément par le système visuel humain. Pietikäinen et al ont montré que la mesure de la texture LBP peut également être appliquée à des images multispectrales [13]

MLBP_c a été développé par Arco Lucifer [22], un opérateur qui décrit les relations des pixels locales dans trois bandes. En plus des interactions spatiales des pixels au sein d'un groupe, les interactions entre les bandes sont prises en considération. Ainsi, l'ensemble de voisinage pour un pixel est constitué des voisins locaux dans les trois groupes



La figure II.13. Mesure de texture $MLBP_c$ décrit les relations spatiales au sein d'un groupe et entre les bandes

L'opérateur $MLBP_c$ pour une description de la couleur de texture locale se traduit par la formule suivante

$$\begin{aligned}
 MLBP_c = \sum_{i=0}^{p-1} & \text{sign}(g_i^{b1} - g_c^{b1}) + \text{sign}(g_i^{b2} - g_c^{b1}) + \text{sign}(g_i^{b3} - g_c^{b1}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b2}) + \text{sign}(g_i^{b2} - g_c^{b2}) + \text{sign}(g_i^{b3} - g_c^{b2}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b3}) + \text{sign}(g_i^{b2} - g_c^{b3}) + \text{sign}(g_i^{b3} - g_c^{b3})
 \end{aligned} \quad (II.18)$$

b1 est la première bande, b2 la deuxième et b3 la troisième.

La première partie de l'équation calcule les valeurs de LBP pour le pixel central de la première bande (b1) basé sur les relations avec les voisins dans le premier groupe et les deux autres groupes (b2, b3), la seconde partie de l'équation calcule les valeurs de LBP pour le pixel central de la deuxième bande et la troisième partie de l'équation (II.18) calcule les valeurs de LBP pour le pixel central de la troisième bande en relation avec les deux autres bandes .

Chacun des trois pixels centraux est par conséquent, calculé par rapport aux pixels voisins dans les autres groupes. Au total, neuf valeurs de LBP sont additionnées pour obtenir le $MLBP_c$

MLBP_c n'est pas seulement une sommation des LBP des bandes individuellement mais, il modélise également les relations de pixels entre les bandes. Ces relations croisées sont très importantes dans la distinction des différentes textures.

II.10. Extension à 4 et à 5 bandes spectrales

La quasi-totalité des capteurs embarqués ont 4, 6, 8, voir 9 bandes spectrales dans le visible et le proche-infrarouge. C'est pourquoi, nous tenterons dans ce travail étendre l'analyse multispectrale du LBP à 4 et 5 bandes.

MLBP 4 Bandes

$$\begin{aligned}
 MLBP_c = \sum_{p=0}^{p+1} & \text{sign}(g_i^{b1} - g_c^{b1}) + \text{sign}(g_i^{b2} - g_c^{b1}) + \text{sign}(g_i^{b3} - g_c^{b1}) + \text{sign}(g_i^{b4} - g_c^{b1}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b2}) + \text{sign}(g_i^{b2} - g_c^{b2}) + \text{sign}(g_i^{b3} - g_c^{b2}) + \text{sign}(g_i^{b4} - g_c^{b2}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b3}) + \text{sign}(g_i^{b2} - g_c^{b3}) + \text{sign}(g_i^{b3} - g_c^{b3}) + \text{sign}(g_i^{b4} - g_c^{b3}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b4}) + \text{sign}(g_i^{b2} - g_c^{b4}) + \text{sign}(g_i^{b3} - g_c^{b4}) + \text{sign}(g_i^{b4} - g_c^{b4})
 \end{aligned}
 \tag{II.19}$$

MLBP 5 Bandes

$$\begin{aligned}
 & \text{sign}(g_i^{b1} - g_c^{b1}) + \text{sign}(g_i^{b2} - g_c^{b1}) + \text{sign}(g_i^{b3} - g_c^{b1}) + \text{sign}(g_i^{b4} - g_c^{b1}) + \text{sign}(g_i^{b5} - g_c^{b1}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b2}) + \text{sign}(g_i^{b2} - g_c^{b2}) + \text{sign}(g_i^{b3} - g_c^{b2}) + \text{sign}(g_i^{b4} - g_c^{b2}) + \text{sign}(g_i^{b5} - g_c^{b2}) + \\
 = \sum_{p=0}^{p+1} & \text{sign}(g_i^{b1} - g_c^{b3}) + \text{sign}(g_i^{b2} - g_c^{b3}) + \text{sign}(g_i^{b3} - g_c^{b3}) + \text{sign}(g_i^{b4} - g_c^{b3}) + \text{sign}(g_i^{b5} - g_c^{b3}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b4}) + \text{sign}(g_i^{b2} - g_c^{b4}) + \text{sign}(g_i^{b3} - g_c^{b4}) + \text{sign}(g_i^{b4} - g_c^{b4}) + \text{sign}(g_i^{b5} - g_c^{b4}) + \\
 & \text{sign}(g_i^{b1} - g_c^{b5}) + \text{sign}(g_i^{b2} - g_c^{b5}) + \text{sign}(g_i^{b3} - g_c^{b5}) + \text{sign}(g_i^{b4} - g_c^{b5}) + \text{sign}(g_i^{b5} - g_c^{b5})
 \end{aligned}
 \tag{II.20}$$

Dans le cas des données à très haute résolution, ces variables sont intégrés dans le processus de classification par le modèle hiérarchique

II.11. Conclusion

Ce chapitre définit et explique l'opérateur (LBP) à partir d'une définition générale de la texture dans un quartier local. Un certain nombre d'extensions de l'opérateur de base ont été abordées, ce qui nous ouvrira la voie à exploiter les données issues des capteurs multispectraux où l'analyse s'avère complexe à cause de la résolution et de la multitude de thème à reconnaître. Cette phase est systématiquement suivie par le processus de segmentation pour aboutir à des documents thématiquement exploitables. C'est ce que nous aborderons dans le chapitre suivant.

Chapitre III

III.1.Introduction

L'interprétation d'images est souvent composée d'une étape de segmentation et d'une étape de classification, ces étapes constituent le cœur de tout système de vision.

La segmentation a pour objectif l'extraction d'indices visuels dans une image, elle permet de partitionner une image en ensembles de points appelés régions, homogènes pour une ou plusieurs caractéristiques (intensité, couleur, texture etc.).

La classification est similaire à la segmentation telle qu'on la connaît dans d'autres domaines. À chaque région, représentée par un vecteur d'attributs, est affectée une étiquette de classe.

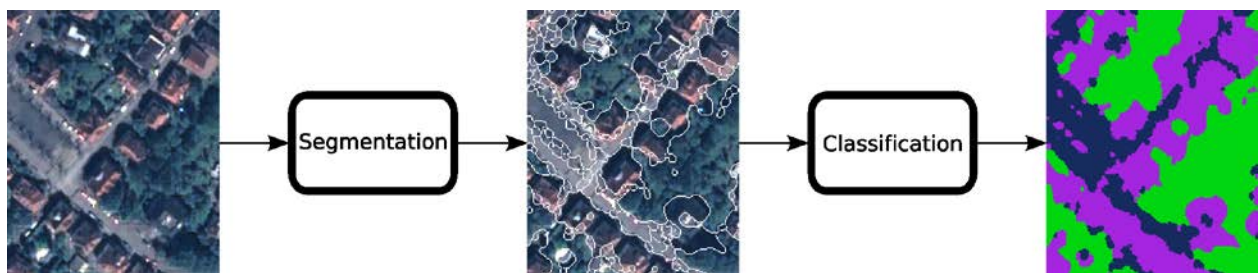


Figure III.1. : Processus d'interprétation d'images par segmentation puis classification.

III.2.Définition de la segmentation

La segmentation d'une image texture consiste à diviser cette même image en plusieurs régions homogènes comportant des caractéristiques de texture identique, nous pouvons adopter la définition suivante :

"La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image A en sous-ensembles R_i , appelés régions tels qu'aucune région ne soit vide. Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines telles que l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image" [23].

Un traitement visant à extraire uniquement deux régions de l'image initiale peut être vu comme une binarisation.

III.3. Différentes méthodes de segmentation

Dans l'objectif d'analyser une image, la segmentation est alors une étape primordiale. On regroupe de façon usuelle les méthodes de segmentation en 4 groupes :

1. Approche globale de l'image (pixellaire) ;
2. La recherche de discontinuités afin de mettre en évidence les contours ;
3. La recherche d'homogénéité locale pour définir les régions ;
4. Ou encore la coopération des deux principes (frontière-région).

III.3.1. Approche globale de l'image (pixellaire)

Le principe consiste à regrouper les pixels selon leurs attributs (intensité, contraste etc.) sans tenir compte de leur localisation au sein de l'image. Cela permet de construire des classes de pixels ; les pixels adjacents, appartenant à une même classe, forment alors des régions. Parmi les approches pixellaires, on distingue les méthodes par seuillage et des méthodes permettant une classification multi-attributs [24].

On prend comme exemple la méthode du seuillage des niveaux de gris.

Le principe général du seuillage consiste à chercher une valeur appropriée du seuil, puis de classifier tous les pixels de l'image selon la valeur de leurs niveaux de gris par rapport à ce seuil, afin de séparer les régions d'intérêt et l'arrière plan de l'image. Plusieurs méthodes de détermination de cette valeur de seuil ont été appliquées pour la segmentation d'image.

Le seuillage peut être :

- Global : un seuil pour toute l'image.
- Local : un seuil pour une portion de l'image.
- Adaptatif : un seuil qui s'ajuste selon les images, ou parties de l'image.

Le seuillage le plus simple est le seuillage global, il a pour principe :

Soit p un pixel et V une fonction définissant le niveau de gris et S un seuil :

$$V(p) = \begin{cases} 0 & \text{si } V(p) < S \\ 1 & \text{si } V(p) \geq S \end{cases}$$

Le résultat sera évidemment une image binaire

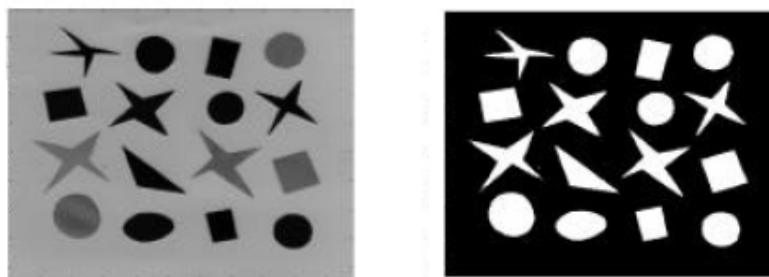


Image originale

image seuillée (binaire)

Figure III.2 : Exemple de segmentation pixellaire.

Le seuillage d'histogramme est une méthode basée sur l'hypothèse qu'à chaque mode de l'histogramme correspond une région.

Si un seuillage simple est utilisé, on obtient une séparation de l'histogramme en deux mi lieux uniquement. Pour les cas plus complexes, d'histogrammes présentant plus de deux classes, un seuillage multiple devient nécessaire. Ainsi, chaque pixel de l'image est classé selon son niveau de gris.

Exemple

L'histogramme des niveaux de gris est utilisé afin d'en déduire le seuil. Si un seuillage simple est utilisé, on obtient une séparation de l'histogramme en deux milieux uniquement.

Soit I une image composée d'un objet clair sur un fond sombre, dont l'histogramme des niveaux de gris est représenté par la figure III.3 :

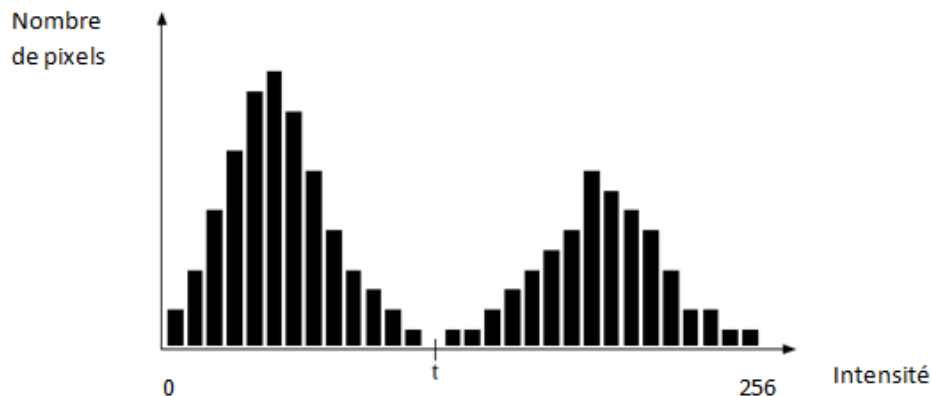


Figure III.3. : Exemple d'histogramme d'une image.

On place le seuil t entre les deux modes. Les pixels dont le niveau de gris est supérieur à t seront attribués à l'objet, les autres pixels au fond.

III.3.2. Approche contour

Les approches contour ne cherchent pas à trouver les régions elles même mais leurs frontières. On peut définir les frontières ou les contours comme des zones de l'image où l'intensité des pixels change brusquement. Cette variation dans l'image est le passage d'un niveau de gris à un autre.

La variation peut être détectée par le maximum d'une dérivée première de la fonction niveau de gris d'un point, ou par le passage par zéro de la seconde dérivée [24].

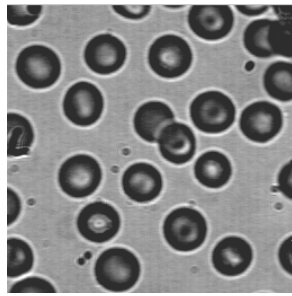
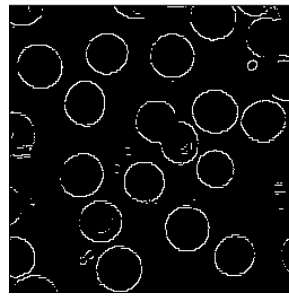


Image original



segmentation en contours

Figure III.4. : Exemple de segmentation par approche contour.

On discerne trois grandes familles d'approches, les méthodes de calculs de gradients discrets, les détecteurs de contour et les modèles déformables [25]

III.3.3.Approche région

Contrairement à l'approche frontière qui recherche les discontinuités, l'approche région recherche plutôt les continuités. Les méthodes de cette approche fournissent une carte de régions fermées. Cependant la localisation des frontières reste généralement peu précise.

L'approche région a pour objectif de décomposer l'image en un ensemble de régions connexes, les plus homogènes possibles et les plus différentes par rapport aux autres régions. Chaque pixel est affecté à une région unique après une partition de l'image. Notons qu'une région est dite homogène si les caractéristiques des pixels la constituant sont sensiblement identiques. Or pour une image en niveau de gris, un pixel peut être caractérisé soit par son niveau de gris, soit par un ensemble de paramètres de texture [26].

On distingue deux familles d'algorithmes pour l'approche région :

Les méthodes de croissance de régions : la croissance s'effectue à partir de pixels initiaux appelés « germes » lors d'une itération du processus, les pixels adjacents à la région sont étudiés. S'ils vérifient les conditions d'homogénéité, le pixel sera alors agrégé dans la région. Et les pixels non intégrés aux régions peuvent générer eux-mêmes de nouvelles régions [24].

Les méthodes de décomposition et fusion : Le principe consiste en une alternance de phases de division et de fusion de régions jusqu'à optimiser un critère d'homogénéité choisi préalablement. Pour l'étape de fusion, on recherche dans les couples de régions possibles (régions adjacentes quasi similaires), des candidats à un possible regroupement. Chaque couple retenu est noté en fonction de l'impact qu'aurait sa fusion sur le critère d'homogénéité

global. Les couples les mieux notés sont alors fusionnés. L'étape de décomposition (division) agit de façon opposée.

Les méthodes qui fusionnent ou divisent les régions en fonction du critère choisi sont dites descendantes [27].

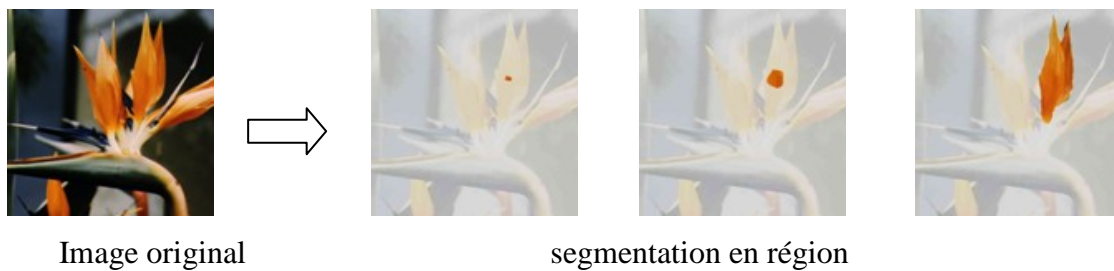


Figure III.5. : Exemple de segmentation par la méthode de croissance de région.

III.3.4. Approche coopérative

Les approches frontières et régions s'appuient sur des informations différentes et complémentaires. Cela a incité plusieurs chercheurs à développer des systèmes de segmentation par coopération de méthodes. Ainsi, pallier les faiblesses de chacune des techniques : la faible précision du contour (approche région) et l'obtention de régions non fermées (approche contour) [28].

On distingue en principe trois types de coopération [29]:

1. La coopération série : donne naissance à des algorithmes de segmentation puissants ;
2. La coopération parallèle : elle fait appel à trois mécanismes, la fusion, l'adaptation et la correction. Généralement, la modélisation utilisée favorise un de ces trois mécanismes, ce qui ne contribue pas d'une manière générale à des résultats très probants.
3. La coopération hybride : permet de solutionner quelques lacunes des deux coopérations précédentes.

III.4. Définition de classification

Le principe de la classification de textures est de déterminer la classe associée à un type de texture particulier. Elle consiste à organiser un ensemble de données multidimensionnelles en un ensemble fini de classes selon un ou plusieurs critère(s) de classification à l'aide d'un classifieur. Il existe de nombreuses méthodes de classification de données multidimensionnelles (multispectrales). La figure III.6 illustre le principe de la classification. Les données sont représentées par des points dans un espace à n dimensions (dans notre exemple n=2). En sortie de la classification, on obtient M classes (ou familles) de points selon les critères donnés et le classifieur choisi (dans notre exemple, M=3).

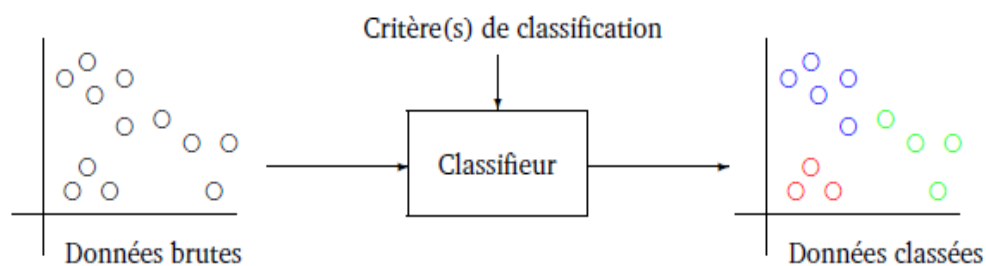


Figure III.6. Principe de la classification

Les pixels de l'image sont regroupés en un ensemble de M classes disjointes. A chaque pixel de l'image, on attribue une étiquette parmi les M étiquettes qui correspondent chacune à une classe.

Mathématiquement, la classification est définie de la manière suivante [30] :

Soit $X = \{X_1, X_2, \dots, X_N\}$ l'ensemble des N observations (pixels) à classer.

Chaque observation X_i est caractérisée par p paramètres $X_i = \{x_{i1}, x_{i2}, \dots, x_{ip}\}$.

Soit $C = \{c_1, c_2, \dots, c_k\}$, l'ensemble des k classes de l'image.

La classification consiste à répartir l'ensemble des N pixels en k classes tel que :

- $c_i \neq \emptyset$ pour $i = 1, 2, \dots, k$
- $c_i \cap c_j = \emptyset \quad \forall i \neq j$
- $\cup c_i = C$

III.5. Les approche de classification

Une première partition des méthodes de classification peut être faite, entre les méthodes dites de classification supervisées et les méthodes non supervisées.

III.5.1.Approche classification supervisé

La classification supervisée est une technique de classification automatique où l'on cherche à produire automatiquement des règles pour regrouper des individus en un certain nombre de classes, à partir d'une base de données d'apprentissage contenant des échantillons (les régions d'entraînement).

L'utilisation des techniques supervisées nécessite l'intervention d'un utilisateur et aussi une première phase d'apprentissage durant laquelle le système apprend les caractéristiques associées à chaque classe d'une base d'apprentissage. Une fois cette étape réalisée, ces méthodes supervisées sont capables de classifier les individus d'une base de test en fonction des caractéristiques de textures qui leurs sont associées.

Il existe plusieurs méthodes de classification supervisé on parlera particulièrement de la méthode K plus proches voisins et maximum de vraisemblance

III.5.1.1. Les k-plus-proches-voisins

Cette approche est supervisée, c'est une approche très simple et directe. Son principe est le suivant :

Une donnée d'une distance entre les éléments que l'on veut classifier. Nous considérons une base d'apprentissage de n éléments $Z = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, où \mathbf{x}_i est le vecteur descripteur de l'élément i et \mathbf{y}_i est sa classe. Pour la classification d'une nouvelle observation \mathbf{x} , nous calculons d'abord sa distance à tous les éléments dans la base d'apprentissage $\mathbf{d}_i = \|\mathbf{x} - \mathbf{x}_i\|$.

Nous obtenons donc n valeurs réelles et positives des distances \mathbf{d}_i . De plus, nous trouvons les k plus petites valeurs parmi ces n valeurs \mathbf{d}_i .

Ces k valeurs correspondent aux k éléments $\{\mathbf{x}_i, \mathbf{y}_i\}$ de la base d'apprentissage qui sont les plus proches de la nouvelle observation \mathbf{x} .

En fin, la classe de \mathbf{x} que nous souhaitons classifier est attribuée à la classe majoritaire \mathbf{y}_i parmi ces k éléments [30].

La Figure III.7 présente un exemple visuel de l'algorithme des k-plus proches voisins.

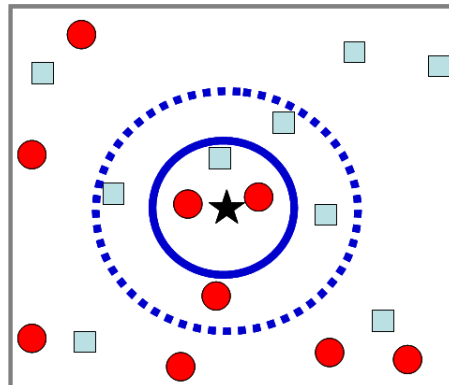


Figure III.7. : Illustration de la classification par la méthode des k-plus proches voisins.

Considérant une base de données de 18 éléments de deux classes : cercles et carrés, un nouvel élément étoile est attribué à la classe des cercles si on considère la valeur de $k = 3$, parce que ses trois plus proches voisins sont constitués de deux cercles et d'un carré. Par contre, si on considère ses $k = 7$ voisins, le nouvel élément est attribué à la classe des carrés car ses 7 premiers voisins sont constitués de 4 carrés pour 3 cercles.

III.5.1.2. La méthode du maximum de vraisemblance

La méthode du maximum de vraisemblance (Maximum Likelihood) minimise le risque d'erreur, elle se base sur des lois de probabilité et la règle de Bayes. Cette loi calcule pour chaque pixel sa probabilité d'appartenir à une classe plutôt qu'une autre. Le pixel est assigné à la classe pour laquelle la probabilité d'appartenance est la plus élevée, la totalité de l'image est classée sur cette base [31].

D'après la règle de Bayes, la probabilité qu'un pixel x appartienne à une classe C_q est :

$$P(C_q / x) = \frac{P(x/C_q) \times P(C_q)}{P(x)} \quad (\text{III.1})$$

$$\text{Avec } P(x) = \sum_{q=1,2,\dots} P(x/C_q) \times P(C_q) \quad (\text{III.2})$$

Soient :

$P(C_q / x)$: La probabilité que le pixel x appartienne à la classe C_q .

$P(x / C_q)$: La probabilité d'être le pixel x dans la classe C_q .

On affecte x à la classe C_q si et seulement si la probabilité que x appartienne à la classe C_q est supérieure à la probabilité que x appartienne à n'importe quelle autre classe j :

$$P(C_q / x) > P(C_j / x) \text{ quelque soit } j.$$

Ou $[P(x / C_q) \times P(C_q)] > [P(x / C_j) \times P(C_j)]$ (conformément à la règle de Bayes).

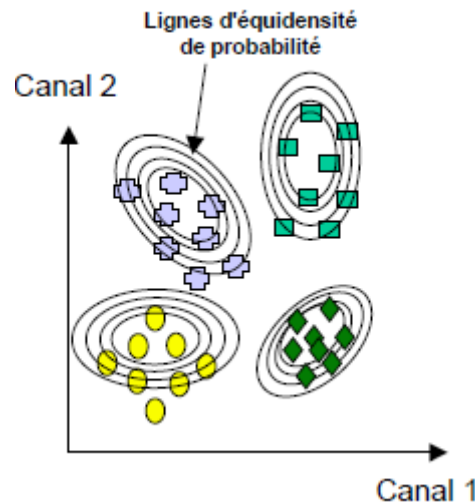


Figure III.8. : Exemple de la classification par la méthode du maximum de vraisemblance

III.5.2. Approche classification non supervisé

Lorsque le type des objets dans l'image n'est pas connu, cela résulte d'un manque d'information ou d'incertitude sur la réalité du terrain, il existe des algorithmes de classification qui permettent de créer un regroupement de pixels similaires. Dans le cadre de la classification non-supervisée, il est généralement nécessaire d'indiquer le nombre de classes recherchées. Les textures similaires sont alors regroupées à l'intérieur d'une même classe sans l'intervention de connaissances a priori et simplement à partir d'estimations de similarité entre caractéristiques [32].

III.5.2.1. La méthode des k-means

k-means est l'une des méthodes les plus connues pour la classification non supervisée créée par J. Mac Queen [33] en 1967. Elle nécessite comme unique paramètre, le nombre de classes. L'algorithme k-means permet de répartir un ensemble de données (individus ou pixels de l'image) en k classes homogènes. La valeur de k n'est pas connue, elle est choisie selon une connaissance préalable (combien de régions sont souhaitées) ou de manière aléatoire. Chaque

groupe est représenté par son centre de gravité qui correspond à la moyenne des éléments (pixels) du groupe. Cette méthode de classification est basée sur le mouvement de ces centres, qui se déplacent vers une position de stabilité après chaque itération.

Le centre de gravité d'une classe C_k est calculé comme suit :

$$\mu_k = \frac{1}{N} \sum_{x_i \in C_k} x_i \quad (\text{III.3})$$

Avec : N est le nombre d'observations (individus ou pixels).

k est le nombre de classes souhaitées.

$i = 1, \dots, k$ attribut de la classe d'un individu x .

Après avoir déterminé les paramètres (intensité par exemple) de toute région, chaque individu est affecté au centre le plus proche (généralement selon une mesure de distance euclidienne). Après l'affectation de toutes les données, les centres de gravités sont recalculés, ils constituent les nouveaux représentants des groupes. Le critère d'arrêt de cet algorithme correspond à la stabilité des classes (où aucune donnée ne change pas de groupe), c.à.d. lorsque les centres des classes ne changent pas durant deux itérations successives : $\mu(n, p) = \mu(n, p+1)$

avec $n=1, \dots, k$ et $\mu(n, p)$: centre de gravité de classe n à la $p^{\text{ème}}$ itération.

L'algorithme de k -means minimise la fonction suivante :

$$J = \sum_i^k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (\text{III.4})$$

Où $\| \cdot \|$ représente la norme euclidienne, qui mesure la distance entre un point et un centre.

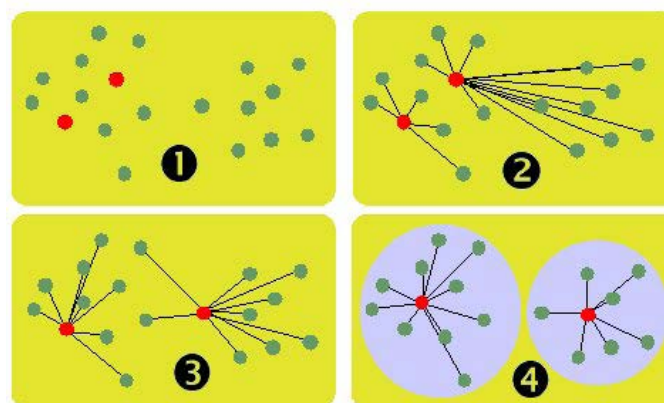


Figure III.9. : Exemple de classification avec la méthode de k -means.

À la fin de la classification par les k-means, on obtient des classes strictes, c'est-à dire que chaque individu n'appartient qu'à une seule classe. Les individus situés en périphérie d'une classe, près de la frontière d'une autre classe, posent le problème d'appartenance stricte à une classe unique [32].

III.5.2.2. La méthode ascendante Hiérarchique

La notion de regroupement hiérarchique recouvre différentes méthodes de classification.

La classification ascendante hiérarchique est dite ascendante car elle part d'une situation où tous les individus sont seuls dans une classe, puis sont rassemblés en classes de plus en plus grandes. Le qualificatif "hiérarchique" vient du fait qu'elle produit une hiérarchie H , l'ensemble des classes à toutes les étapes de l'algorithme [33].

Initialement, chaque individu forme une classe, soit n classes. On cherche à réduire le nombre de classes à $N_{bclasse} < n$, ceci se fait itérativement. La première itération a un indice d'agrégation (valeur de dissimilarité) faible, mais celui-ci va croître d'itération en itération.

Itération 1 : k éléments sont à classer. Les k éléments forment chacun une classe,

Itération 2 : l'algorithme calcule les distances entre chacun des k éléments et regroupe les deux plus proches. On obtient ainsi une première partition en $(k-1)$ classes ;

Itération 3 : l'algorithme recalcule les distances entre le nouvel élément et les éléments restants, puis regroupe à nouveau les deux éléments les plus proches. On obtient alors une deuxième partition à $(k-2)$ classes ;

Itération n : le processus est réitéré jusqu'à la dernière partition qui constitue une seule grande classe regroupant l'ensemble des éléments de l'image.

Les itérations sont présentées dans la figure suivante.

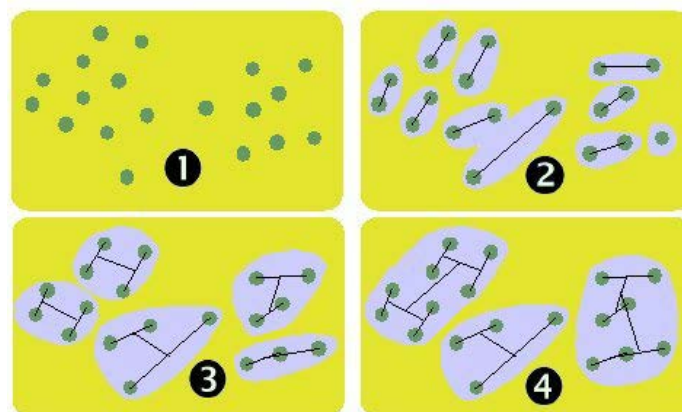


Figure III.10. Exemple de classification avec la méthode ascendante Hiérarchique.

Donc le principe est assez simple : à chaque itération, on commence par réunir, dans une même classe, les deux pixels les plus proches en distance. Ensuite, on remplace cette classe par un pixel moyen qui la représente pendant les itérations suivantes, jusqu'à ce que la classe soit avalée par une autre classe, puis on itère ce processus, jusqu'à ce qu'il n'y ait plus qu'une seule classe.

Les étapes de la classification sont modélisées sur un diagramme, appelé dendrogramme : on peut y identifier les classes qui ont fusionné pour créer une classe plus importante et évaluer la distance entre les différentes classes (deux classes extrêmement éloignées tarderont à être réunies dans une classe plus importante) [34]. L'analyse de ce diagramme peut aider à déterminer le nombre de classes le plus intéressant à obtenir, par visualisation du résultat sur l'image.

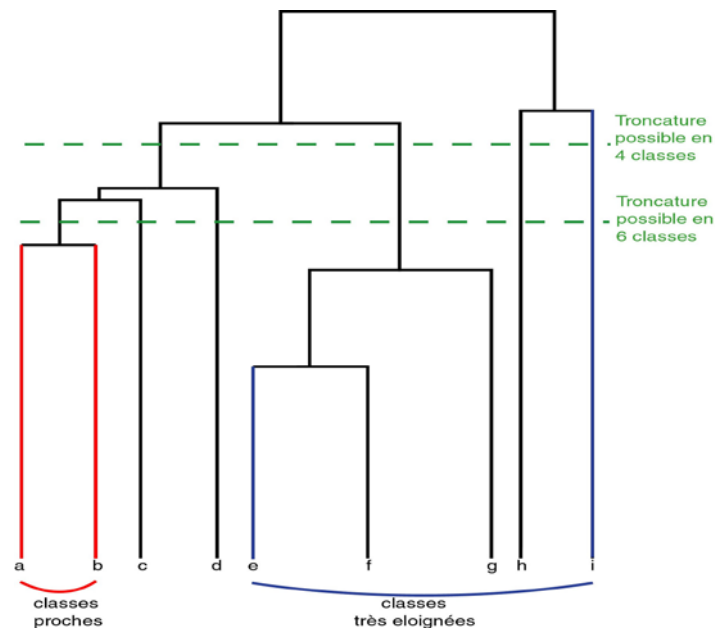


Figure III.11. : Exemple de dendrogramme.

Ces hiérarchies sont généralement appréciées puisqu'elles permettent une visualisation de l'organisation des données et du processus de classification. A partir de ce dendrogramme, il est possible d'obtenir une partition de X en coupant l'arbre à un niveau l donné. Par exemple, le choix de $l = 4$ dans le dendrogramme de la figure III.11 renvoie le partitionnement suivant :

$$C = \{(a,b,c,d) (e,f,g) (h) (i)\}$$

Cette méthode a l'avantage de présenter des temps de calcul relativement courts. Toutefois, il faut se méfier du choix des paramètres sur la distance, ils peuvent conduire à des résultats très différents, et il faut tenir compte de la difficulté d'interprétation,

III.6. Évaluation d'une classification

- **Matrice de confusion**

Pour bien analyser les performances d'un système de classification plusieurs outils sont utilisés [34]. La matrice de confusion est la base de plusieurs évaluations statistiques des performances, elle permet de détecter les confusions entre les classes à partir d'une vérité terrain. Cette dernière peut être caractérisée soit à partir de relevés sur le terrain, soit en utilisant les zones test saisies, donc la matrice de confusion compare le résultat de la classification avec l'information de réalité de terrain. C'est une matrice carrée de taille égale au nombre de classes, et dont un élément m_{ij} est égal au pourcentage de pixels classés λ_j qui auraient dû appartenir à la classe λ_i [35]

Normalement, une matrice de confusion a des valeurs proches de 1 sur la diagonale, et des valeurs proches de 0 ailleurs. Idéalement, c'est la matrice identité.

Pour construire cette matrice, il suffit de produire un tableau qui met en relation le nombre d'échantillons bien classés et mal classés.

A partir de cette matrice, plusieurs indicateurs permettent d'évaluer la qualité de la classification, soit pour chaque classe prise individuellement ou globalement sur l'ensemble des éléments

1- Précision globale (overall accuracy) :

Le taux moyen de pixels correctement classés OA est obtenue en divisant le nombre de pixels classés correctement (diagonale de la matrice) par la somme des pixels de toutes les classes d'apprentissage. Ceci, exprime par conséquent, les pixels correctement classés par rapport à ceux mal classés

2- Coefficient Kappa : l'analyse KAPPA est une technique discrète multivariable , l'analyse KAPPA produit une statistique KHAT(une estimation de kappa) qui est une mesure de précision. La statistique KHAT se calcule suivant l'expression :

$$KHAT = \frac{N \sum_{i=0}^r x_{ii} - \sum_{i=0}^r (x_{i+} * x_{+i})}{N^2 - \sum_{i=0}^r (x_{i+} * x_{+i})} \quad (III.5)$$

N : est le nombre total des pixels ;

r : est le nombre des lignes de la matrice de confusion ;

x_{ii} : est la somme des pixels placer correctement (la diagonale de la matrice) ;

x_{i+} : est le totale des pixels de chaque classe d'apprentissage ;

x_{+i} : est le total des pixels affecter a chaque classe.

Plus la statistique KHAT est proche de 1, plus la classification est correcte

Vérité terrain

				Total
<i>Classification</i>	4	1	0	4/5 80%
	1	1	1	1/3 33%
	0	0	1	1/1 100%
Total	4/5 80%	1/2 50%	1/2 50%	

$$K = (9 \cdot (4 + 1 + 1) - (1 \cdot 1 + 0 \cdot 0 + 0 \cdot 1)) / (92 - (1 \cdot 1 + 0 \cdot 0 + 0 \cdot 1)) = 53/80 = 66,2\%$$

Figure III.12. : Exemple de matrice de confusion et le calcul de KAPPA

III.7. Conclusion

Dans ce chapitre nous avons décrit le lien qui existe entre deux domaines segmentation et classification et leurs différentes méthodes qui conduisant à l'interprétation de l'image.

Dans un premier temps, nous avons introduit la notion de la segmentation, puis nous avons donné un aperçu sur les différentes approches de cette dernière puis on a vu la segmentation supervisée et non supervisée et particulièrement, la méthode du maximum de vraisemblance qu'on mettra en pratique dans le chapitre suivant

Avec ces méthodes, on vu que c'est possible de définir les objets présents dans la scène. Or il n'existe pas de technique de reconnaissance capable de recréer le processus d'analyse qu'un enfant de quatre ans arrive à faire au premier coup d'œil.

Chapitre IV

Partie 1 : Implémentation

Après avoir présenté les différentes techniques LBP, nous exposerons dans cette partie l'implémentation de nos algorithmes sous l'environnement Windows en mettant l'accent sur le choix de C++ sous l'environnement Windows

IV.1.1. Le langage C en traitement d'images numériques

Le langage C est un langage de programmation inventé par MM. Kernighan et Ritchie au début des années 70. Au début des années 90, Bjarne Stroustrup fait évoluer le langage vers le langage C++ en lui rajoutant notamment les notions orientées objet. Toutefois, bien que le C++ ait évolué à partir du C, et ait gardé un grand nombre de notions et de syntaxes de son «ancêtre»,

Le langage C++ a été retenu pour le traitement d'image, car il est le langage le plus communément utilisé dans le domaine du traitement d'images numériques. La principale motivation du courant emploi de ce dernier, est sa rapidité d'exécution, car le traitement d'images requiert des ressources conséquentes, tant au niveau de la gestion de la mémoire, que de la charge processeur. En effet, la majeure partie des traitements est basée sur le principe d'itérations (assez souvent des boucles imbriquées sont utilisées), dans la mesure où il faut traiter tous les pixels d'une image, qui est représentée par une matrice (ou un vecteur de vecteurs). En outre, le langage C permet une grande maîtrise de la mémoire (gestion par pointeurs), et également, une optimisation des accès à cette mémoire (accès directs par pointeur). De plus, il est également possible de grandement optimiser les algorithmes.

IV.1.2. Caractéristiques de base des images numériques

Une image numérique est une matrice à deux dimensions, représentée du point de vue du langage C++, par un tableau à deux dimensions "[]" (double crochets). Du point de vue de la mémoire. Nous conviendrons donc, que chaque image correspond à un emplacement mémoire de taille prédéterminée (allocation de mémoire statique, par opposition à l'allocation de mémoire dynamique). Définir une variable pour une image s'effectue par conséquent de la façon suivante : `unsigned char image[256][256];` Cette première déclaration permet de construire une image d'une taille de 256 pixels par 256. Le type utilisé, « unsigned char », indique qu'il s'agit d'une image dont chaque pixel est codé sur 1 octet, soit 8 bits.

8 bits permettent de coder 28 niveaux, soit 256 (valeurs comprises entre 0 et 255, 0 représentant le noir, et 255 le blanc). L'image possède pour conclure, 256*256 pixels, codés chacun sur 256 niveaux (du gris dans notre cas). Pour information, chaque image "pèsera" ainsi 65536 octets, soit 64 Ko.

Accéder à un pixel est relativement simple. Suivant la déclaration précédente, si l'on souhaite obtenir la valeur du pixel présent à la position (115, 28), il suffit d'écrire :

```
unsigned char pixel ;  
pixel = image[115][28];
```

Ou encore, "balayer" tous les pixels d'une image s'effectue de la façon suivante (balayage ligne par ligne) :

```
for( i=0; i<HAUTEUR_IMAGE; i++ )  
    for( j=0; j<LARGEUR_IMAGE; j++ )  
        pixel = image[i][j];
```

IV.1.3. Du fichier source à l'exécutable

Nombreux programmeurs utilisent cependant un **EDI** (Environnement de Développement Intégré) au lieu d'un simple compilateur pour compiler leurs programmes. Un EDI est un logiciel qui intègre un éditeur de texte pour taper les codes sources, un compilateur pour les traduire en exécutable ainsi que d'autres outils aidant à la mise au point et à la distribution des programmes. Il existe de nombreux EDIs pour le langage C++. Certains sont gratuits (Code::Blocks Studio, Visual C++ Express, etc.), d'autres payants (CodeGear RAD Studio, Microsoft Visual Studio, etc.). , nous avons opté pour l'environnement de Microsoft, Visual Studio.

IV.1.4. Microsoft visuel studio

Microsoft Visual Studio est un environnement de développement intégré (IDE) de Microsoft. Il peut être utilisé pour développer des consoles et des applications graphiques de l'interface utilisateur ainsi que les applications Windows Forms, des code natif avec le code managé pour toutes les plateformes prises en charge par Microsoft Windows, Windows Mobile,

Windows CE, .NET Framework, .NET Compact Framework et Microsoft Silverlight. Visual Studio inclut un éditeur de code IntelliSense soutien ainsi que la refactorisation de code. Le débogueur intégré fonctionne à la fois comme un débogueur de niveau source et un débogueur de niveau de la machine.

IV.1.5. Présentation des algorithmes

Algorithme LBP de base : application de l'approche LBP sur l'image

Entrée :

1. créer une matrice de N lignes et M colonnes image [N][M].
2. créer une matrice de N+2 lignes et M+2 colonnes image1[N+2][M+2].

Début :

- Charger l'image à traiter en lecture.
- Charger l'image pour l'écriture.
- Charger les pixel de l'image sur la matrice image [N][M].
- Charger le contenu de la matrice image [ligne][colonne] sur la matrice image1[N+2][M+2].
- Applique l'effet miroir pour la matrice image1[N+2][M+2].
- **Pour** (ligne =1 jusqu'à nombre total N+1) Faire

Pour (colonne =1 jusqu'à nombre total M+1) Faire

$$LBP = \sum_{p=0}^{p-1} S(g_p - g_c) * 2^p$$

$$\text{avec } S(g_p - g_c) = \begin{cases} 1 & \text{si } g_p \text{ et plus grand que } g_c \\ 0 & \text{si } g_p \text{ et plus petit que } g_c \end{cases}$$

Charger le contenu de LBP sur le pixel de l'image qui est en écriture.

Fin Pour

Fin Pour

Fin.

Algorithme LBP_{c,j}**Entrée :**

1. créer une matrice de N lignes et M colonnes image [N][M].
2. créer une matrice de N+2 lignes et M+2 colonnes image1 [N+2][M+2].

Début :

- Charger l'image à traiter en lecture.
- Charger l'image pour l'écriture.
- Charger les pixels de l'image sur la matrice image[N][M].
- Charger le contenu de la matrice image [ligne][colonne] sur la matrice image1[N+2][M+2].
- Appliquer l'effet miroir pour la matrice image1 [N+2][M+2].
- **Pour** (ligne = 1 jusqu'à nombre total N+1) Faire

Pour (colonne = 1 jusqu'à nombre total M+1) Faire

$$U_c = \sum_{i=1}^p |\text{sign}(g_i - g_c) - \text{sign}(g_{i-1} - g_c)|$$

$$LBP_{c,j} \begin{cases} \sum_{i=0}^{p-1} \text{sign}(g_i - g_c) & \text{si } U_c \leq j \\ p + 1 & \text{sinon} \end{cases} \quad \text{avec } j = 2$$

Charger le contenu de LBP_{c,j} sur le pixel de l'image qui est en écriture.

Fin Pour

Fin Pour

Fin

Algorithme VAR_c**Entrée :**

1. créer une matrice de N lignes et M colonnes image [N][M].
2. créer une matrice de N+2 lignes et M+2 colonnes image1 [N+2][M+2].

Début :

- Charger l'image à traiter en lecture.
- Charger l'image pour l'écriture.
- Charger les pixels de l'image sur la matrice image [N][M].
- Charger le contenu de la matrice image [ligne][colonne] sur la matrice image1[N+2][M+2].
- Applique l'effet miroir pour la matrice image1 [N+2][M+2].
- **Pour** (ligne =1 jusqu'à nombre total N+1) Faire
 - **Pour** (colonne =1 jusqu'à nombre total M+1) Faire

$$VAR_c = \sum_{i=0}^{p-1} (g_i - \mu_c)^2 \quad AVEC \quad \mu_c = \frac{1}{p} \sum_{i=0}^{p-1} g_i$$

Charger le contenu de VAR_c sur le pixel de l'image qui est en écriture.

Fin Pour

Fin Pour

Fin.

Algorithme MLBP 3 Bandes

Entrée :

1. crier trois matrices de N lignes et M colonnes $R[N][M]$, $V[N][M]$, $B[N][M]$.
2. crier trois matrices de N+2 lignes et M+2 colonnes $R1[N+2][M+2]$, $V1[N+2][M+2]$, $B1[N+2][M+2]$.

Début :

- Charger les trois bandes à traiter en lecture.
- Charger l'image pour l'écriture.
- Charger les pixel de la bande rouge sur la matrice $R[N][M]$.
- Charger les pixel de la bande verte sur la matrice $V[N][M]$.
- Charger les pixel de la bande bleu sur la matrice $B[N][M]$.
- Charger le contenu de la matrice $R[\text{ligne}][\text{colonne}]$ sur la matrice $R1[N+2][M+2]$.
- Charger le contenu de la matrice $V[\text{ligne}][\text{colonne}]$ sur la matrice $V1[N+2][M+2]$.
- Charger le contenu de la matrice $B[\text{ligne}][\text{colonne}]$ sur la matrice $B1[N+2][M+2]$.
- Applique l'effet miroir pour les matrices $R1[N+2][M+2]$, $V1[N+2][M+2]$, $B1[N+2][M+2]$.
- **Pour** (ligne =1 jusqu'à nombre total N+1) Faire
 - Pour** (colonne =1 jusqu'à nombre total M+1) Faire

$$MLBP_c = \sum_{i=0}^{p-1} \begin{matrix} \text{sign}(g_i^R - g_c^R) + & \text{sign}(g_i^V - g_c^R) + & \text{sign}(g_i^B - g_c^R) + \\ \text{sign}(g_i^R - g_c^V) + & \text{sign}(g_i^V - g_c^V) + & \text{sign}(g_i^B - g_c^V) + \\ \text{sign}(g_i^R - g_c^B) + & \text{sign}(g_i^V - g_c^B) + & \text{sign}(g_i^B - g_c^B) \end{matrix}$$

Charger le contenu de $MLBP_c$ sur le pixel de l'image qui est en écriture.

Fin Pour

Fin Pour

Fin.

Algorithme MLBP 4 bande

Entrée :

1. crier cinq matrices de N lignes et M colonnes $b_{11}[N][M]$, $b_{22}[N][M]$, $b_{33}[N][M]$, $b_{44}[N][M]$.
- 2.crier cinq matrices de N+2 lignes et M+2 colonnes $b_1[N+2][M+2]$, $b_2[N+2][M+2]$, $b_3[N+2][M+2]$, $b_4[N+2][M+2]$.

Début :

- Charger quatre bande à traiter en lecture.
- Charger l'image pour l'écriture.
- Charger les pixel de la bande 1 sur la matrice $b_{11}[N][M]$.
- Charger les pixel de la bande 2 sur la matrice $b_{22}[N][M]$.
- Charger les pixel de la bande 3 sur la matrice $b_{33}[N][M]$.
- Charger les pixel de la bande 4 sur la matrice $b_{44}[N][M]$.
- Charger le contenu de la matrice $b_{11}[ligne][colonne]$ sur la matrice $b_1[N+2][M+2]$.
- Charger le contenu de la matrice $b_{22}[ligne][colonne]$ sur la matrice $b_2[N+2][M+2]$.
- Charger le contenu de la matrice $b_{33}[ligne][colonne]$ sur la matrice $b_3[N+2][M+2]$.
- Charger le contenu de la matrice $b_{44}[ligne][colonne]$ sur la matrice $b_4[N+2][M+2]$.
- Applique l'effet miroir pour les matrices $b_1[N+2][M+2]$, $b_2[N+2][M+2]$, $b_3[N+2][M+2]$, $b_4[N+2][M+2]$.

Pour (ligne =1 jusqu'à nombre total N+1) Faire

Pour (colonne =1 jusqu'à nombre total M+1) Faire

$$MLBP_c = \sum_{p=0}^{p+1} \begin{matrix} sign(g_i^{b^1} - g_c^{b^1}) + sign(g_i^{b^2} - g_c^{b^1}) + sign(g_i^{b^3} - g_c^{b^1}) + sign(g_i^{b^4} - g_c^{b^1}) + \\ sign(g_i^{b^1} - g_c^{b^2}) + sign(g_i^{b^2} - g_c^{b^2}) + sign(g_i^{b^3} - g_c^{b^2}) + sign(g_i^{b^4} - g_c^{b^2}) + \\ sign(g_i^{b^1} - g_c^{b^3}) + sign(g_i^{b^2} - g_c^{b^3}) + sign(g_i^{b^3} - g_c^{b^3}) + sign(g_i^{b^4} - g_c^{b^3}) + \\ sign(g_i^{b^1} - g_c^{b^4}) + sign(g_i^{b^2} - g_c^{b^4}) + sign(g_i^{b^3} - g_c^{b^4}) + sign(g_i^{b^4} - g_c^{b^4}) \end{matrix}$$

Charger le contenu de $MLBP_c$ sur le pixel de l'image qui est en écriture.

Fin Pour

Fin Pour

Fin.

Algorithme MLBP 5 Bandes

Entrée :

1. crier cinq matrices de N lignes et M colonnes $b11[N][M]$, $b22[N][M]$, $b33[N][M]$, $b44[N][M]$, $b55[N][M]$.
2. crier cinq matrices de N+2 lignes et M+2 colonnes $b1[N+2][M+2]$, $b2[N+2][M+2]$, $b3[N+2][M+2]$, $b4[N+2][M+2]$, $b5[N+2][M+2]$.

Début :

- Charger les cinq à traiter en lecture.
- Charger l'image pour l'écriture.
- Charger les pixel de la bande 1 sur la matrice $b11[N][M]$.
- Charger les pixel de la bande 2 sur la matrice $b22[N][M]$.
- Charger les pixel de la bande 3 sur la matrice $b33[N][M]$.
- Charger les pixel de la bande 4 sur la matrice $b44[N][M]$.
- Charger les pixel de la bande 5 sur la matrice $b55[N][M]$.
- Charger le contenu de la matrice $b11[ligne][colonne]$ sur la matrice $b1[N+2][M+2]$.
- Charger le contenu de la matrice $b22[ligne][colonne]$ sur la matrice $b2[N+2][M+2]$.
- Charger le contenu de la matrice $b33[ligne][colonne]$ sur la matrice $b3[N+2][M+2]$.
- Charger le contenu de la matrice $b44[ligne][colonne]$ sur la matrice $b4[N+2][M+2]$.
- Charger le contenu de la matrice $b55[ligne][colonne]$ sur la matrice $b5[N+2][M+2]$.
- Applique l'effet miroir pour les matrices $b1[N+2][M+2]$, $b2[N+2][M+2]$, $b3[N+2][M+2]$, $b4[N+2][M+2]$, $b5[N+2][M+2]$.
- **Pour** (ligne =1 jusqu'à nombre total N+1) Faire
Pour (colonne =1 jusqu'à nombre total M+1) Faire

$$MLBP_c = \sum_{p=0}^{p+1} \begin{aligned} & \text{sign}(g_i^{b^1} - g_c^{b^1}) + \text{sign}(g_i^{b^2} - g_c^{b^1}) + \text{sign}(g_i^{b^3} - g_c^{b^1}) + \text{sign}(g_i^{b^4} - g_c^{b^1}) + \text{sign}(g_i^{b^5} - g_c^{b^1}) + \\ & \text{sign}(g_i^{b^1} - g_c^{b^2}) + \text{sign}(g_i^{b^2} - g_c^{b^2}) + \text{sign}(g_i^{b^3} - g_c^{b^2}) + \text{sign}(g_i^{b^4} - g_c^{b^2}) + \text{sign}(g_i^{b^5} - g_c^{b^2}) + \\ & \text{sign}(g_i^{b^1} - g_c^{b^3}) + \text{sign}(g_i^{b^2} - g_c^{b^3}) + \text{sign}(g_i^{b^3} - g_c^{b^3}) + \text{sign}(g_i^{b^4} - g_c^{b^3}) + \text{sign}(g_i^{b^5} - g_c^{b^3}) + \\ & \text{sign}(g_i^{b^1} - g_c^{b^4}) + \text{sign}(g_i^{b^2} - g_c^{b^4}) + \text{sign}(g_i^{b^3} - g_c^{b^4}) + \text{sign}(g_i^{b^4} - g_c^{b^4}) + \text{sign}(g_i^{b^5} - g_c^{b^4}) + \\ & \text{sign}(g_i^{b^1} - g_c^{b^5}) + \text{sign}(g_i^{b^2} - g_c^{b^5}) + \text{sign}(g_i^{b^3} - g_c^{b^5}) + \text{sign}(g_i^{b^4} - g_c^{b^5}) + \text{sign}(g_i^{b^5} - g_c^{b^5}) \end{aligned}$$

Charger le contenu de $MLBP_c$ sur le pixel de l'image qui est en écriture.

Fin Pour

Fin Pour

Fin.

Partie 2 : Expérimentation

Cette deuxième partie est consacrée d'abord, à décrire brièvement les données qui ont fait l'objet des traitements, ainsi que la région d'étude qui rentre dans un contexte d'une région aride algérienne, située dans le Hoggar. La phase suivante consiste à tester et à évaluer les approches développées.

Afin de quantifier nos approches développées, nous les avons testées sur des images multispectrales de Landsat 7 ETM+ sur la région du Hoggar. Le but de l'analyse sur cette région aride, partiellement accessible, est de cartographier la géologie des sols. Les images Landsat 7 ETM+ sont idéales pour une cartographie d'échelles variant entre 1/50.000 et 1/200.000.

La région d'Amsinnassène a été choisie d'une part, pour sa diversité des thèmes géologiques caractérisés par de divers textures et d'autres parts, pour la connaissance a priori du terrain et l'existence de quelques cartes géologiques. L'investigation du terrain et l'élaboration du fichier « échantillons » ont été réalisés par l'équipe de l'USTHB [36], en collaboration étroite avec notre promoteur.

IV.2.1. Présentation de la zone d'étude

La région d'Amsinassène est comprise entre la faille Ouest-Atakor et la bande de mylonite allongée NNW- SSE, sur laquelle s'appuie le village d'In Amguel. Elle fait partie d'un domaine plus vaste désigné sous le voculaire de l'Arechchoum qui présente une grande variété lithologique. Elle est située à environ 90Km au nord ouest de la ville de Tamanrasset entre les coordonnées 23°20' et 23°38' de latitude Nord et entre 5°8' et 5°25' de longitude Est (Figure IV.2).

IV.2.2. Méthodologie

Nous avons utilisé une fenêtre multispectrale de 512 x 512, extraite de la scène *Thematic Mapper* (TM) de référence Path 192 et Row 44 du 27 novembre 2000, géoréférencée en UTM 31 N (figure IV.1). Nous rappelons que les données TM présentent 3 bandes dans le visible (TM1, TM2, TM3) et 3 bandes dans le proche infrarouge (TM4, TM5, TM7), le tout avec une résolution de 30 x 30 mètres, ce qui nous permettra de travailler aisément à l'échelle du 1 : 50 000. Le but ultérieur est de procéder à l'usage du système d'information

géographique (SIG) pour des opérations d'analyse afin d'apporter des informations sur les différentes structures géologiques.

La méthode utilisée dans notre cas, est l'intégration des variables décrivant la textures par les techniques LBP dans des processus de classification multispectrale, afin d'améliorer la convergence de ces derniers.

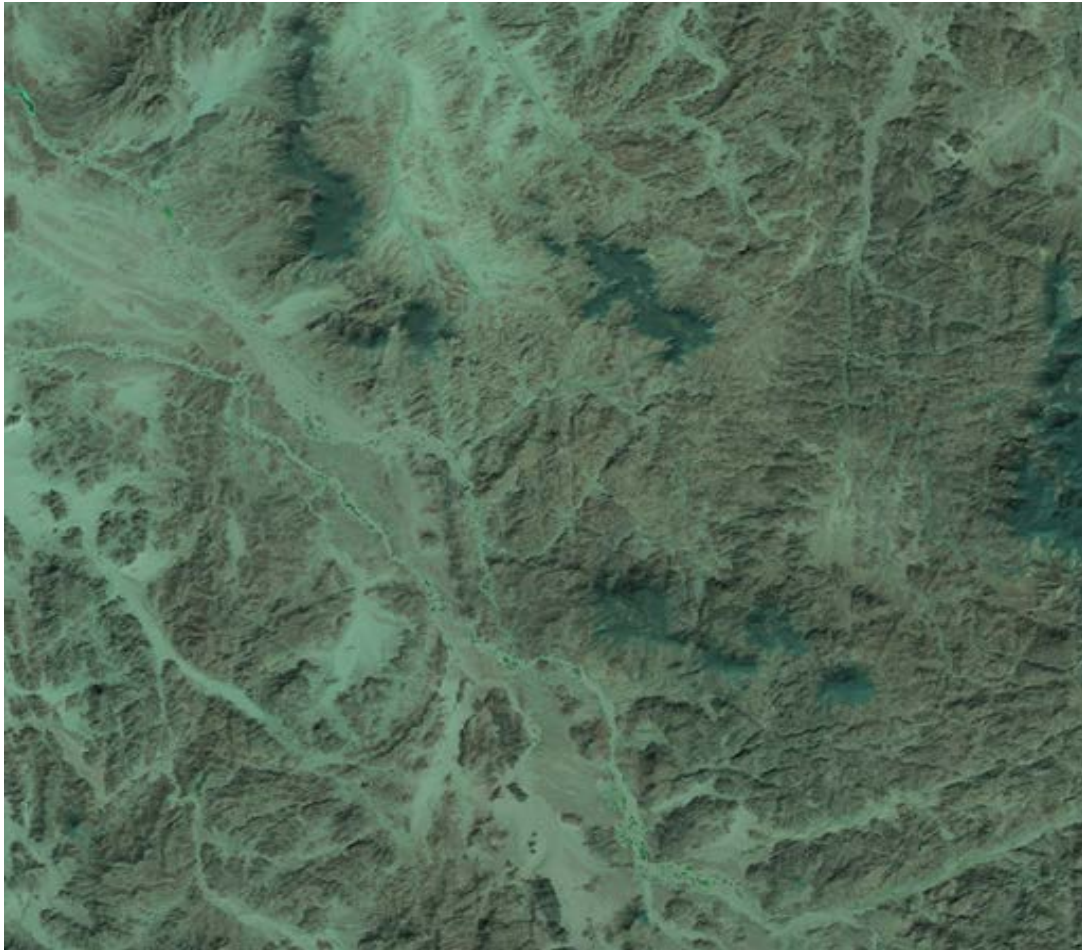


Figure. IV.1. : Composition colorée issue des 3 canaux TM3 TM4 et TM7

Pour la validation et la quantification des résultats, les documents thématiques suivants ont été utilisés

- ❖ La carte géologique d'Amsinassène au 1/50000.
- ❖ Une image radar Alos-Scansar N23E005 dont la résolution est de 100 m .
- ❖ La carte topographique d'Assekrem à l'échelle 1/200 000.

IV.2.3. Sélection des zones d'entraînements (échantillons terrains) et Analyse des canaux bruts

Cette partie a été réalisée par l'équipe de l'USTHB en étroite collaboration avec notre promoteur [36].

En se servant des documents thématiques cités ci-dessus et des missions sur le terrain, 5 classes d'occupation des sols ont été recensées. Il s'agit : 1. Basaltes, 2. Granites monzonitiques, 3. Marbres-pyroxénites-quartzites, 4. Diorite, 5. Tonalite et Sable des oueds

(La figure IV.2) représente le « fichier échantillons », calé sur l'image TM7. Ce dernier sera utilisé pour la classification supervisée qui utilise des variables (des néo-canaux) texturales, calculées par les opérateurs LBP

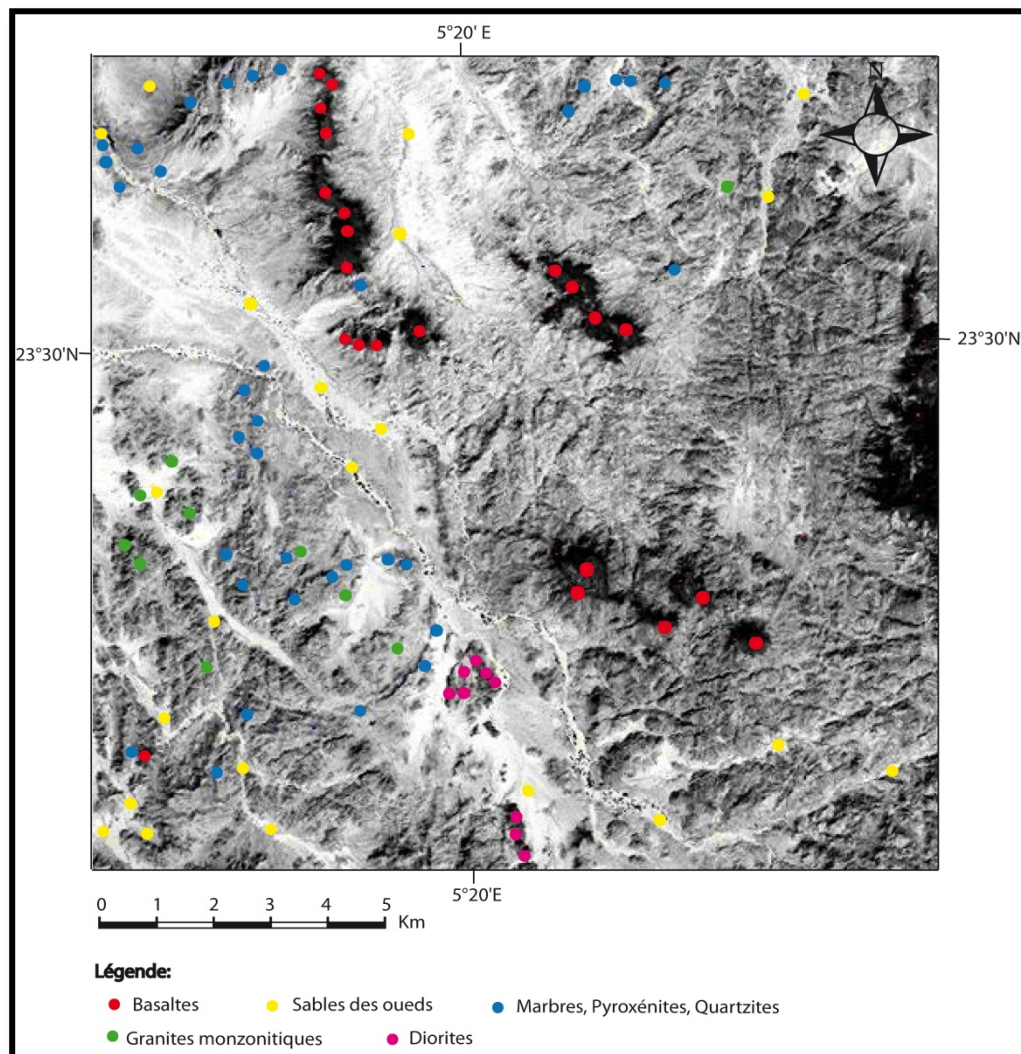


Figure IV.2. : Fichier échantillons, calé sur l'image du canal TM7

IV.2.4. Mesure de texture

(La figure IV.3) représente l'illustration du résultat du $\mathbf{LBP}_{c,j}$ calculé sur l'image TM7 avec $R=2$ et $P=16$. Chaque pixel est quantifié par rapport aux radiométries de ces voisins, ce qui nous a permis de former des régions élémentaires de petites tailles ayant une texture plus au moins identique selon le principe LBP. Avec, $R=1$ et $P=8$, nous avons obtenue de micro-structures difficilement interprétables. Le traitement a été réalisé avec $j=2$, qui semble le mieux adapté pour ce type d'images. Nous rappelons que l'image de sortie est codée sur 17 niveaux seulement, ce qui nous rend difficile, son interprétation visuelle.

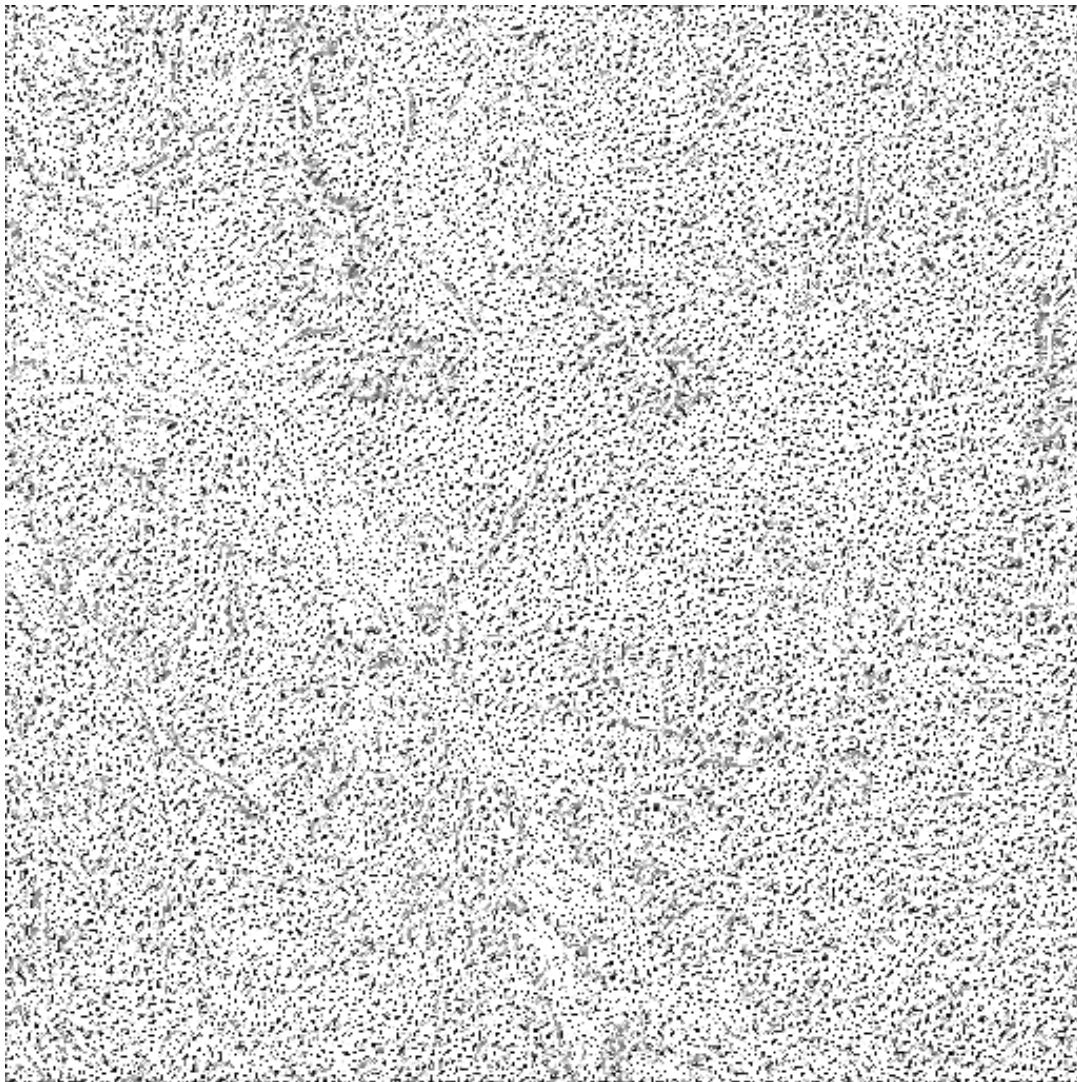


Figure IV.3 : Résultat du $\mathbf{LBP}_{c,j}$ appliqué à la bande TM7 ($R=2$, $P=16$ et $j=2$).

Nous avons de même calculé sur la même image TM7, la variance \mathbf{VAR}_c avec les mêmes paramètres que le traitement précédent, soit $R=2$ et $P=16$. Le résultat est illustré dans

(la figure IV.4) Nous pouvons constat déjà visuellement, la formation de différentes textures des formations géologiques d'intérêt.

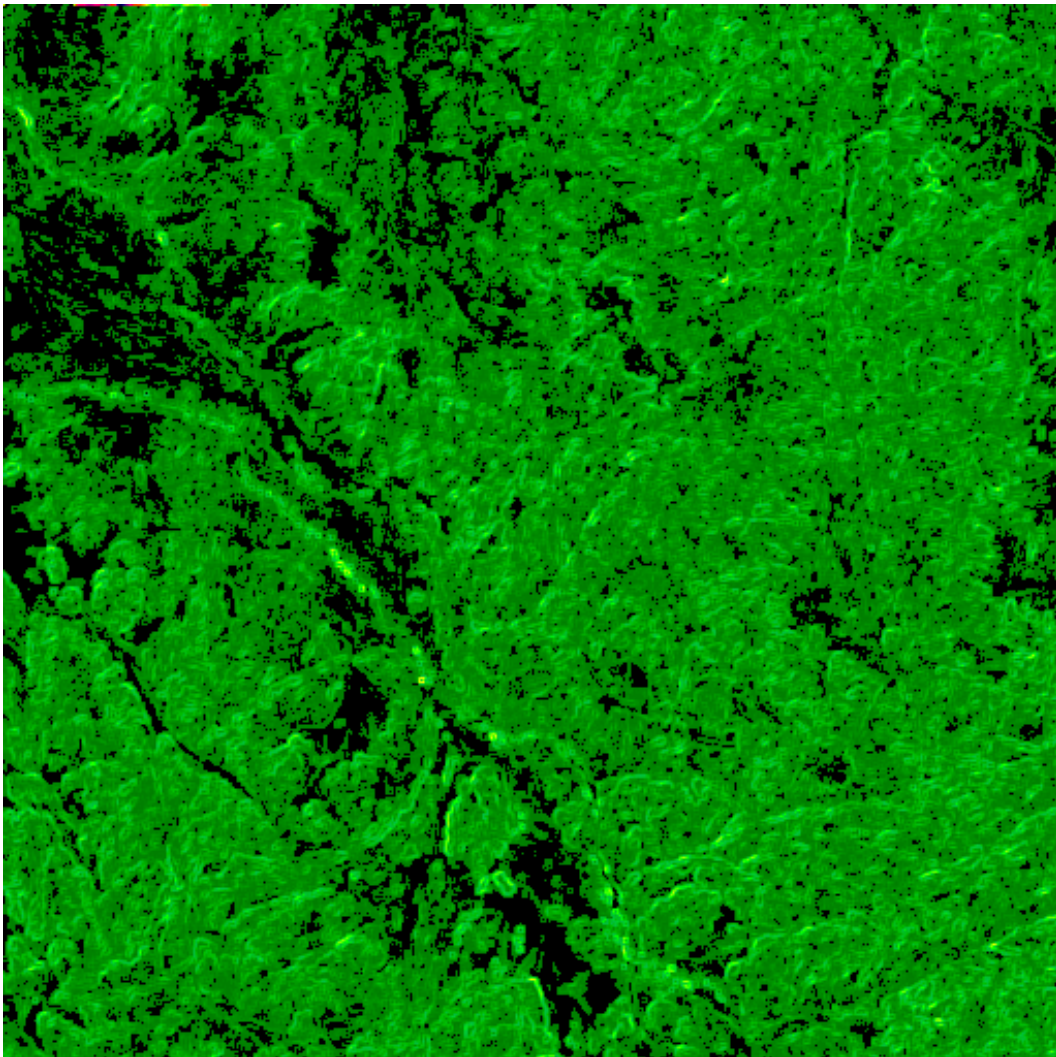


Figure IV.4. : Résultat de VAR_c appliqué à la bande 7 avec $R=2$ et $P=16$.

IV.2.5. LBP multiples

Nous avons appliqué le LBP multiples (MLBP) d'abord sur les 3 bandes spectrales brutes TM3, TM5 et TM7 puis, nous avons testé notre approche sur le modèle qui utilise 5 Bandes. Nous avons sélectionné dans notre cas, les bandes TM2, TM3, TM4, TM5 et TM6. Nous avons fortement constaté une redondance majeure entre les bandes TM1 et TM2. (Les figures IV.5 et IV.6) illustrent les résultats obtenus dans les deux cas. Nous pouvons constater aisément d'autres informations pertinentes permettant une bonne discrimination des différents thèmes d'occupation des sols. Les vallées profondes sont particulièrement mises en évidence, ainsi que les mini-linéaments caractérisant la région. Déjà visuellement nous remarquons une

nette séparabilité des classes recensées. Nous remarquons aussi qu'en augmentant le nombre de variables, la texture devient de plus en plus fine (Figure IV. 6).

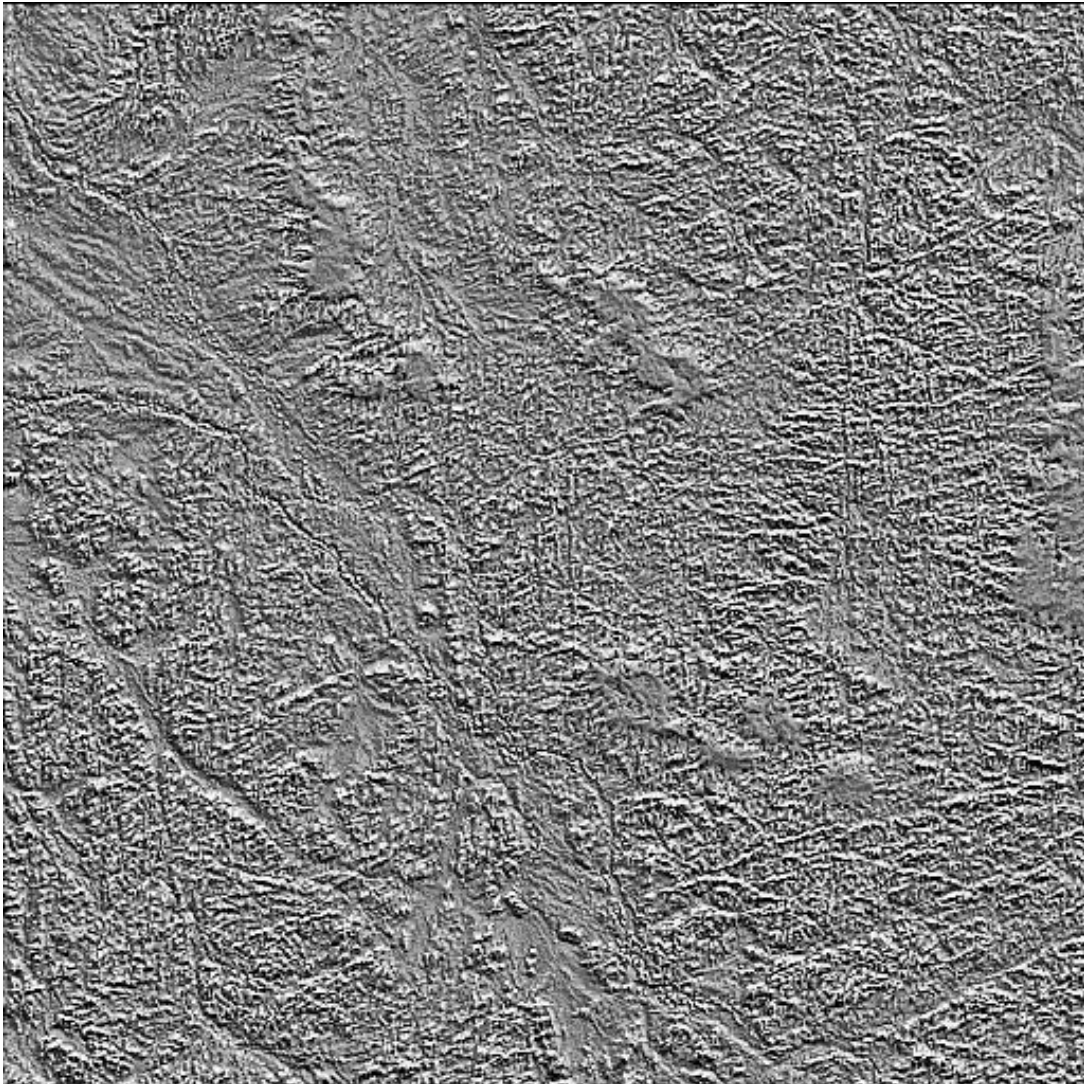


Figure IV.5. 3LBP issue des 3 canaux TM3, TM5 et TM7

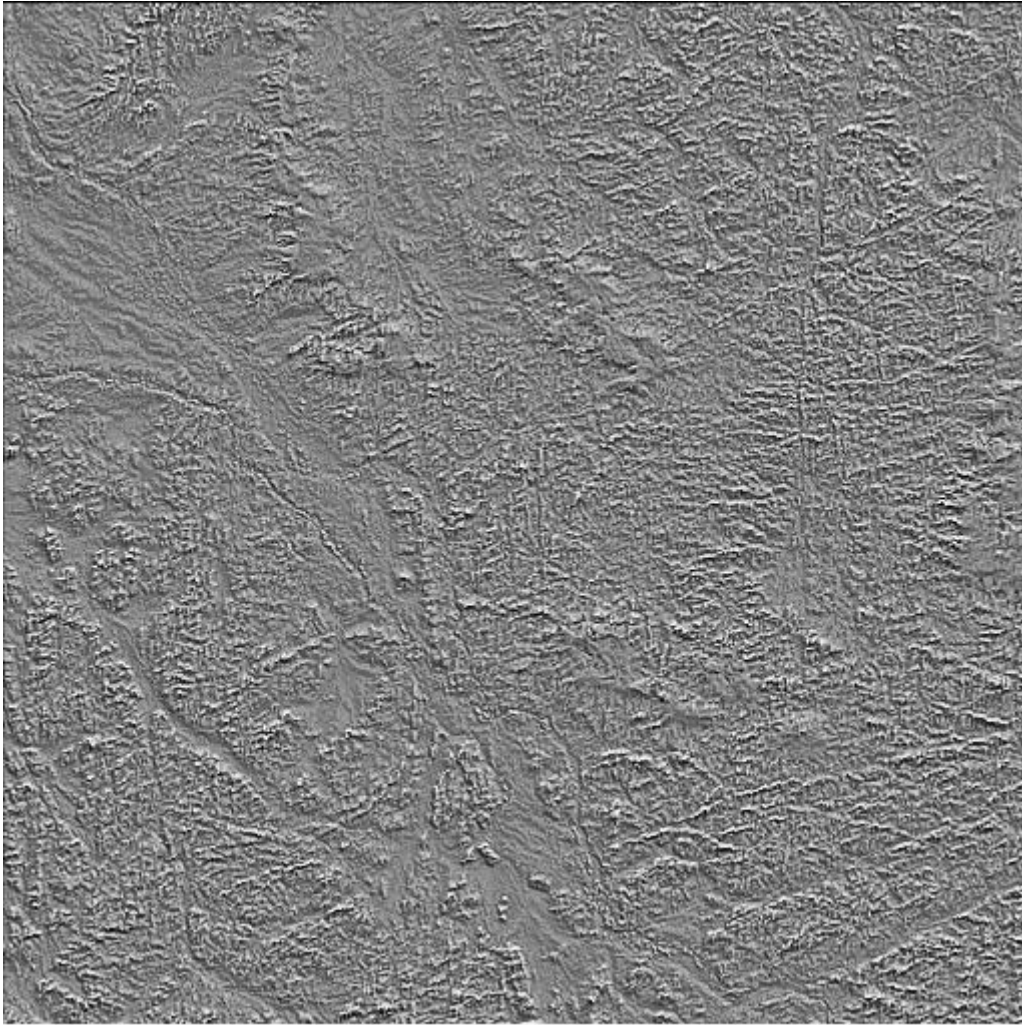


Figure IV. 6.5LBP issue des 5 canaux TM2, TM3, TM4, TM5 et TM7

VI.2.6. Classification multispectrale supervisée

Dans cette partie, nous essayons de quantifier la contribution de nos nouvelles variables, calculées à partir des canaux bruts dans un processus de classification multispectrale. Certains auteurs [22] ont choisi la méthode hiérarchique supervisée qui est justifié par la grande résolution des images utilisées. Ils ont utilisé en effet, les images multispectrales issues du capteur CASI (Compact Airborne Spectral Imager). Ce capteur, embarqué sur un avion, permet d'obtenir des images multibandes dans les domaines visible et proche infra rouge avec une résolution au sol de 2 x 2 m.

La méthode hiérarchique perd considérablement son efficacité quand on passe à une moyenne ou faible résolution. Notre choix s'est porté sur la technique du **maximum de vraisemblance** qui est très adaptative aux images de moyenne résolution, 30 x 30 m dans notre cas.

Nous rappelons que tous les échantillons sélectionnés répondent plus au moins à la loi normale, une exigence de la convergence du processus du maximum de vraisemblance. Nous avons utilisé le logiciel ENVI pour exécuter le maximum de vraisemblance avec 7 canaux qui sont :

1. **3LBP** (image LBP multivariable issue de TM3, TM5 et TM7) ;
2. **4LBP** (image LBP multivariable issue de TM3, TM4, TM5 et TM7) ;
3. **5LBP** (image LBP multivariable issue de TM2, TM3, TM4, TM5 et TM7) ;
4. **LBP_{c,j}7** appliqué à la bande TM7 (R=2, P=16 et j=2) ;
5. **VAR_c7** appliqué à la bande TM7 ;
6. **LBP_{c,j}4** appliqué à la bande du visible TM4 (R=2, P=16 et j=2) ;
7. **TM6** (canal original qui présente un maximum de séparabilité des classes)

(La figure IV.7) illustre le résultat de la classification issue des 7 bandes précédentes

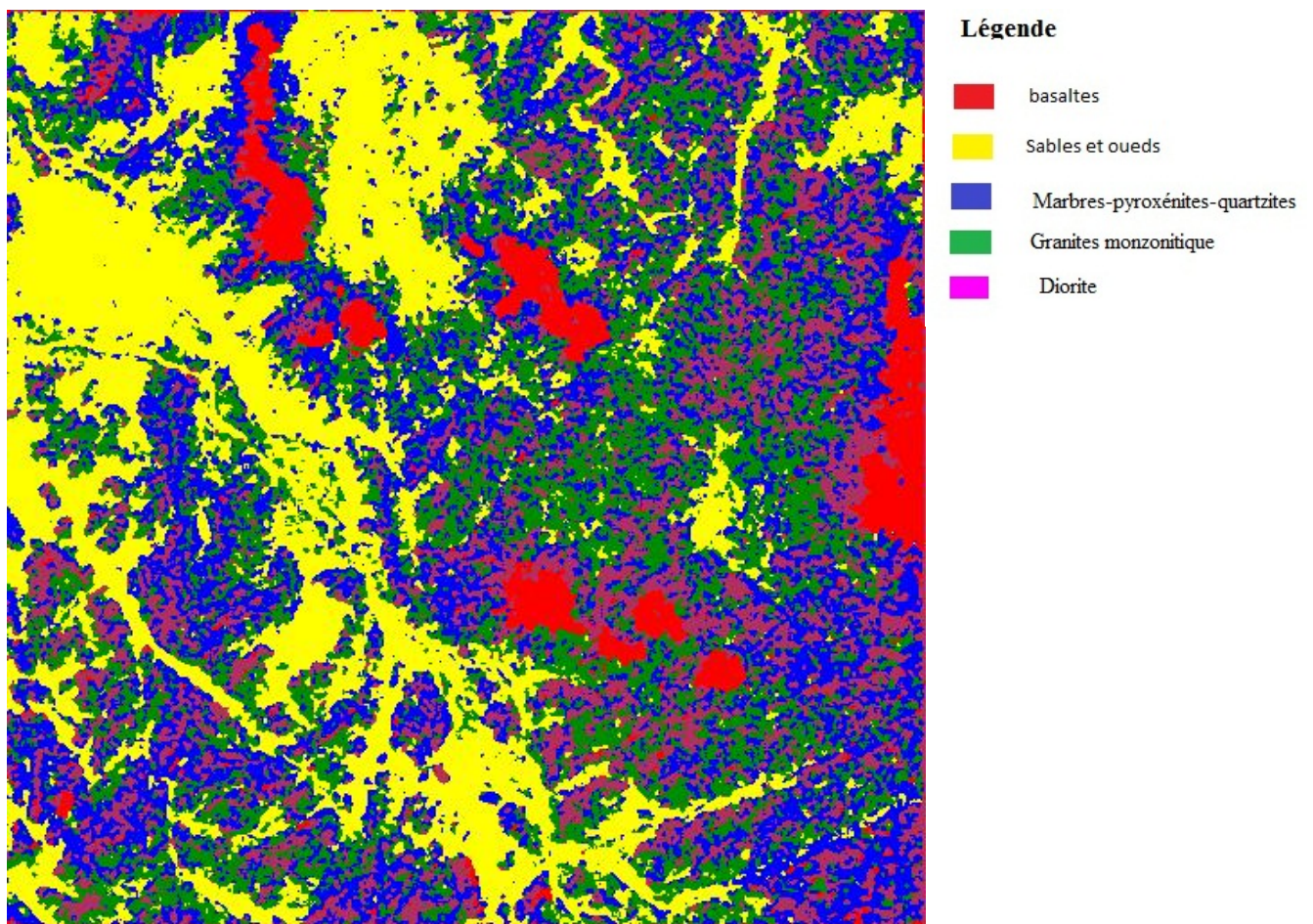


Figure IV.7. Classification multispectrale utilisant les variables LPB

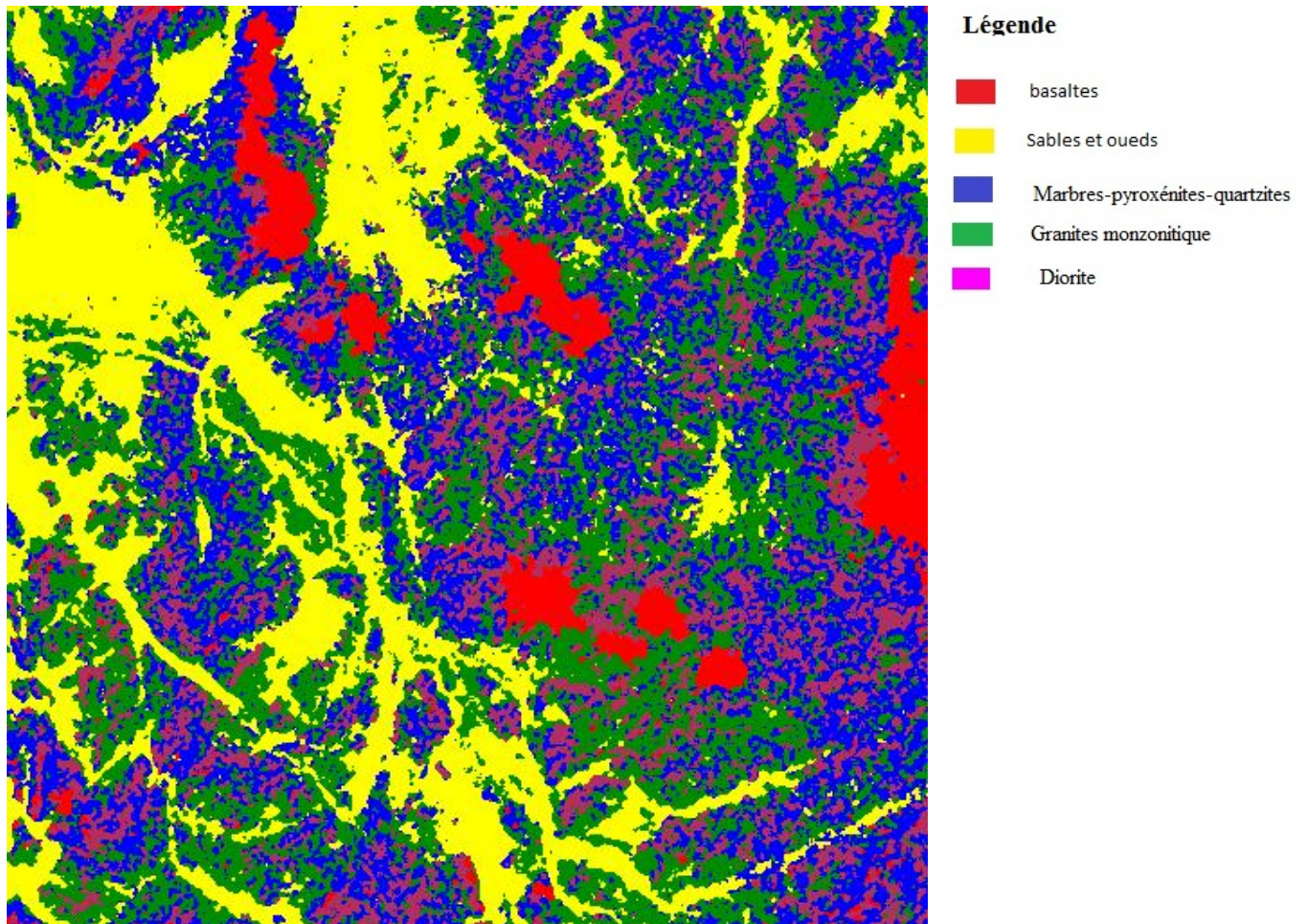


Figure IV.8 : Classification multispectra utilisant les bandes originale (TM2, TM3, TM4, TM5 et TM7)

IV.2.7. Interprétation et discussion

En se référant à la carte géologique et aux autres documents thématiques existant, nous constatons que la classe basalte (rouge) est surévaluée dans la classification obtenue avec les canaux brutes (Figure IV.8), de même pour la classe marbre (bleu). Les zones diorites (magenta) sont plus régulières dans la classification qui utilise les variables LBP (figure IV.7). Nous constatons aisément qu'elles sont sous-évaluées dans l'approche classique. Elles sont souvent absorbées par la classe marbre. Ceci est dû à leurs réponses spectrales qui sont relativement proches. L'intégration des variables texturales LBP a permis d'accentuer leur séparabilité.

Les zones sableuses et oueds (Jaune) se retrouvent plus réguliers dans l'approche classique, ce qui ne correspond pas à la réalité du terrain où elles se chevauchent avec les autres classes. Ceci s'explique par le déplacement fréquent des sables par les vents.

Nous constatons largement que les classes granite et marbre (respectivement vert et bleu) sont absorbées par d'autres classes en utilisant l'approche classique.

Il s'agit simplement dans cette étude d'une comparaison préliminaire où, nous avons constaté la pertinence de l'intégration des variables LBP dans le processus d'une analyse supervisée. Une étude plus exhaustive est primordiale pour sérieusement évaluer l'apport de l'approche LBP dans les classifications multispectrales des données satellitaires à moyen résolution.

Nous noterons de même que nous n'avons pas eu le temps pour procéder à l'évaluation par les matrices de confusion. On s'est simplement d'estimer la contribution de ces nouvelles techniques en se référant aux documents thématiques existants.

IV.3.Conclusion

Dans ce chapitre, nous avons fait une description du langage C++ et l'environnement visuel studio dont nous avons implémente nos applications.

Nous avons présenté les différents résultats obtenus en utilisant plusieurs méthodes de LBP (LBP de base, $LBP_{c,j}$, VAR_c , MLBP 3,4 et 5 bandes) a fin d'extraire les textures, dans le but de les utiliser dans le processus de classification des images.

Nous avons démontré la pertinence de l'intégration des variables LBP dans le processus d'une analyse supervisée en faisant appelle a une simple comparaison entre cette dernière et les méthodes classique (classification avec canaux brutes)

Conclusion générale

Conclusion générale

L'objectif du travail exposé dans ce mémoire, a été l'étude des motifs binaires locaux (LBP) appliqués en traitement d'images, pour la caractérisation des textures et la description des régions locales dans les images, afin d'extraire des entités significatives du point de vue thématique et ce, dans le but d'optimiser les phases de segmentation et de classification

Dans une première partie, nous avons rappelé la notion et les types de la texture dans le cadre d'analyse d'image, et nous avons donné un aperçu sur les différentes méthodes permettant l'extraction des attributs texturaux, la texture est définie et étudiée de manières différentes, en fonction de la façon dont on la caractérise

Dans la deuxième partie, nous avons d'abord exposé l'approche LBP, presque dans sa globalité, appliquée dans des domaines pluridisciplinaires. Nous avons remarqué la simplicité de calcul des modèles LBP et les améliorations apportées sur le LBP original, tel que le $LBP_{P,R}$ qui est introduit pour la description des textures dans des voisinages circulaires. Cette approche nous a permis de présenter le $LBP_{P,R}^{r1}$ qui évite l'effet de rotation de ces voisinages et le LBP^{u2} uniforme qui permet d'extraire une grande quantité de texture. Et nous avons présenté par la suite, le modèles LBP multivariable pour une analyse multispectrale, particulièrement les images satellitaires.

La troisième partie s'est portée sur l'étude des méthodes de segmentation et de classifications qui est une étape indispensable pour obtenir des documents thématiquement exploitables.

Dans la dernière partie, nous avons fait une description du langage C++ et l'environnement visuel studio dont nous avons implémenté nos applications et nous avons présenté les différents résultats obtenus en utilisant plusieurs méthodes de LBP (LBP de base, $LBP_{c,j}$, VAR_c , MLBP 3,4 et 5 bandes) et nous avons remarqué que les approches développées, ont données des résultats très pertinents, néanmoins un travail de réflexion reste encore à faire dans le but de faire une étude plus exhaustive pour mieux appréhender l'utilisation du LBP dans de domaine de l'imagerie spatiale.

Enfin, nous espérons que nous avons apporté un appui considérable pour les étudiants des promotions futurs afin d'optimiser les processus d'analyse des images à base de motif binaire locale et ses extensions, nous souhaitons que d'autres études seront faites sur ce sujet, en prenant en considération les différents domaines d'applications des LBP.

Bibliographique

References Bibliographiques

- [1] R. M. Haralick et Gagallowicz [HARA79] [HARA79] R.M.Haralick, "Statistical and structural approaches to texture", Proceedings of the IEEE, mai 1979, number 5, vol.67.
- [2] Linda G. Shapiro , George C. Stockman, *Computer Vision*, Upper Saddle River: Prentice–Hall, 2001.
- [3]http://www.tsi.telecomparistech.fr/pages/enseignement/ressources/beti/textures_Fourier/texture.html
- [4] S. Mavromatis ,Analyse de texture et Visualisation scientifique, thèse de Doctorat, Université de la Méditerranée, 2001.
- [5] Youness , Saporta. Une méthodologie pour la comparaison de partitions. revue de statistique appliquée, 52:97-120, 2004.
- [6] A.Taleb-Ahmed, P.Dubois,E.Duquenoy,Analysis methods of CT-scan images for the characterization of bone texture: First results”, Pattern Recognition Letters 24 (2003) 1971-1982.
- [7] D. Jeulin. Random texture models for material structures. Statistics and Computing, 10:121–132, 2000. doi:10.1023/A:1008942325749,110.
- [8] A.Bekkhoucha, Traitement d'images et analyse de textures par les longueurs de plages, these de doctorat, Université Mohammed Premier - Oujda ,septembre 2001
- [9] Lehamel Malha, «Segmentation d'Images Texturees a Partir des Attributs Fractals», Memoire de Magister en Automatique Option Traitement d'Images et Reconnaissance de Formes, Departement d' Automatique, Universite Mouloud Mammeri de Tizi Ouzou, 2011.
- [10] T.Ojala, M.Pietikainen and D.Harwood, « A Comparative Study of Texture Measures with Classification Based on Feature Distributions », Pattern Recognition 29(1):51-59, 1996.
- [11] AYHMNE mémoire fin d'étude « traitement et segmentation des image par l'operetur locale binaire » master II université de mouloud Mammeri 2013
- [12] Jonathan Bouchard, « Methodes de Vision et d'Intelligence Artificielles pour la reconnaissance de SJ.Jecimens Coralliens », A l' obtention de la Maitrise en Genie de la Production Automatisee, Ecole de Technologie Superieure, Universite du Quebec, 2011.
- [13] OJALA T., PIETIKÄINEN M., AND MÄENPÄÄ, T., 2002, Multiresolution gray-scale androtation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 971-987

- [14] Thomas Penne, « Développement D'un Système De Tracking Video Sur Camera Robotisée », These de Doctorat en Informatique, Université Blaise Pascal - ClermontFerrand II, 2011.
- [15] Topi Maenpaa, « The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications », InfoTech Oulu and Department of Electrical and Information Engineering, University of Oulu, Finland, 2003.
- [16] Marko Heikkila, Matti Pietikainen, Cordelia Schmid, « Description Of Interest Regions with Local Binary Pattern », June 2008.
- [17] K.Meena, A.Suruliandi, « Local Binary Patterns and Its Variants for Face Recognition », IEEE-International Conference on Recent Trends in Information Technology, Anna University, Chennai, 2011
- [18]Chao Zhu Charles-E Image region description using orthogonal combination of local binary patterns enhanced with color information Université de Lyon,CNRS,EcoleCentraledeLyon,LIRIS,UMR5205,F-69134,France
- [19] S.Hussain, B.Triggs, « Feature Sets and Dimensionality Reduction for Visual Object Detection», In: Proc. British Machine Vision Conference, PP. 112.1-112.102010.
- [20] X. Tan And B. Triggs, « Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions », in IEEE Transactions on Image Processing, 19(6):1635-1650, 2010
- [21] H. Jin, Q. Liu, H. Lu and X. Tong, « Face Detection Using Improved LBP Under Bayesian Framework », International Conference On Image And Graphics, Hong Kong, China. 306-309, 2004.
- [22] Arko Lucieer, Alfred Stein and Peter Fisher, « Multivariate Texture Based Segmentation of Remotely Sensed Imagery for Extraction of Objects and Their Uncertainty ».
- [23] Arnaud CAPRI docteur a l'université d'Orléans « Caractérisation des objets dans une image en vue d'une aide à l'interprétation et d'une compression adaptée au contenu application aux images échographiques »2007
- [24] Cocquerez, J.P., Philipp, S., 1995. " Analyse d'images: filtrage et segmentation," Masson
- [25] [Nikolaidis et Pitas, 2001] N. Nikolaidis and I. Pitas, 3-D Image Processing Algorithms. JohnWiley and Sons Inc., 2001
- [26] [Trémeau et al., 2004] A. Trémeau, C. Fernandez-Maloigne, and P. Bonton. Image numérique couleur. Dunod, Paris, 2004

- [27] Yang, 1988 H.S. Yang, Range Image Segmentation and Classification Via Split and Merge Based on Surface Curvature, Proceedings of 4th International Conference on Pattern Recognition, pp. 58-67, 1988
- [28] Maitre, 2003 Henri maitre, "le traitement d'images", Hermès, 2003
- [39] Zhu et Yuille, Zhu, S. et Yuille, A. (1996). Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Trans. On Pattern Analysis and Machine Intelligence, 18(9):884–900.
- [30] Huu Giao NGUYEN Thèse de Doctorat « Caractérisation et modélisation de la distribution spatiale de signatures locales dans les images : application à la classification d'image sonar de fonds marins ».
- [31] S.Ait Ali et F.Aiboud, «Segmentation D'images Meteosat Par La Methode Du Maximum De Vraisemblance », Mémoire d'ingénieur d'Etat en Electronique Option Contrôle, Département d'Electronique, Université Mouloud Mammeri de Tizi-Ouzou, 2004.
- [32] Ludovic PAULHACTHÈSE doctorat de l'université François – Rabelais « Outils et méthodes d'analyse d'images 3D texturées : Application à la segmentation des images échographiques
- [33] Macqueen, J. B. « Some Methods for Classification and Analysis of Multivariate Observations. Berkeley Symposium on Mathematical Statistics and Probability », University Of California Press, Vol.1, PP 281-297, 1967.
- [34] Laurent Candillier these « Contextualisation, visualisation et évaluation en apprentissage non supervisé (Université de Lille 3), 2006
- [35] Gabriel DOS SANTOS rapport de stage « carte d'occuoation des soles dans la vallée de biriatau » avril 2001
- [36] LACENE, K., OTMANI, D., 2013. Apport de la télédétection à l'étude géologique de la region de l'Amsinassène (Tefedest, Hoggar central). Mémoire de Master. USTHB.

Résumé

Dans l'analyse des textures, on trouve une méthode théoriquement simple mais très puissante avec un opérateur appelé motif binaire locale (LBP). L'opérateur de texture LBP est devenu une approche populaire dans diverses applications. Grâce à ses récentes extensions, il a été transformé en une très puissante mesure de la texture de l'image.

Ce travail concerne l'étude de différentes méthodes de traitements d'images et d'analyse de texture et essentiellement l'application de l'opérateur Local Binary Pattern(LBP).

Le présent travail est structuré en quatre chapitres.

Le premier chapitre présentera les notions et les types de texture, ainsi que les paramètres qui permettent de les quantifier. L'accent est mis sur les méthodes statistiques qui contribueront d'avantage à mieux comprendre l'objet des approches proposées.

Le deuxième chapitre sera spécialement consacré pour la présentation du descripteur LBP, en précisant la manière de calcul de cet opérateur dans les images en niveaux de gris, et comment il joue le rôle dans l'extraction des textures. Nous allons voir aussi quelques extensions du LBP les plus récentes.

Dans le troisième chapitre, nous présenterons les principales approches de la segmentation des images, en se basant essentiellement sur l'approche classification.

Le dernier chapitre est réservé aux tests et à la présentation des résultats obtenus sur quelques images par les approches proposées.

Enfin, nous terminerons notre travail par une conclusion générale, clôturant ce mémoire et quelques perspectives.

Mot clé : vision-analyse de la texture- motif locale binaire – segmentation-classification-multispectrale.
