

République Algérienne Démocratique et Populaire

**Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri De Tizi-Ouzou**



Faculté De Génie Electrique Et d'Informatique

DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE
Option : Commande des Systèmes.**

Présenté par :

**BENSERAYE Mohamed Adel
BERREHAL Boualem**

Proposé et dirigé par : M^r MESSAR Youcef

Thème

**Approche Floue pour la classification des images multispectrales.
Applications aux images XS de SPOT**

Mémoire soutenu publiquement le 30 Septembre 2015 devant le jury composé de :

| | |
|-----------------------------|---------------------|
| <i>Mr Ahmed DIRAMI</i> | <i>President</i> |
| <i>Mr Youcef MESSAR</i> | <i>Encadreur</i> |
| <i>Mme Malika AÏT AIDER</i> | <i>Examinatrice</i> |
| <i>Mme Zohra HALICHE</i> | <i>Examinatrice</i> |

Remerciements

On remercie tout d'abord le bon dieu pour nous avoir donné le courage et la santé pour accomplir ce travail.

Ce travail n'aurait pas pu aboutir à des résultats sans l'aide et les encouragements de plusieurs personnes, nos parents, amis et camarades que nous remercions.

Nos vifs remerciements accompagnés de toute notre gratitude vont ensuite à notre encadreur Mr MESSAR Youcef, professeur à l'université de Tizi Ouzou, pour ses conseils judicieux, sa grande disponibilité et pour nous avoir suivi et orienté.

On adresse nos sincères remerciements aux membres du jury et à l'intérêt qu'ils ont bien voulu porter à ce travail.

Enfin, que tous ceux qui nous ont aidé et encouragé de près ou de loin dans la concrétisation de ce projet, trouvent ici notre gratitude et nos sincères remerciements.

Nous tenons aussi à exprimer notre reconnaissance à tous nos amis pour leur soutien inconditionnel.

SOMMAIRE

| | |
|------------------------------------|---|
| <i>Introduction générale</i> | 1 |
|------------------------------------|---|

Chapitre I : Généralités sur le Traitements d'Images

| | |
|---------------------------------------------|----|
| I.1. Introduction..... | 4 |
| I.2. Classification automatique..... | 4 |
| I.3. Quelles données ?..... | 4 |
| I.4. Qu'est ce qu'une classe ?..... | 5 |
| I.5. Combien de classe ?..... | 5 |
| I.6. Exemple introductif | 5 |
| I.7. Types d'images | 6 |
| I.7.1. Images matricielles (bitmap)..... | 6 |
| I.7.2. Image vectorielles | 6 |
| I.8. Les images bitmap | 7 |
| I.8.1. Concepts et définitions de base..... | 7 |
| I.8.1.1. Pixel | 7 |
| I.8.1.2. Le codage de pixel..... | 7 |
| I.8.1.3. La résolution d'une image..... | 7 |
| I.8.1.4. La luminance | 8 |
| I.8.1.5. Le contraste | 8 |
| I.8.1.6. Le bruit..... | 9 |
| I.8.1.7. L'histogramme..... | 9 |
| I.8.1.8. La palette de couleurs | 10 |

SOMMAIRE

| | |
|-----------------------------------------------------|----|
| I.9. Le codage des couleurs | 11 |
| I.9.1. Les images binaire (noir ou blanc)..... | 11 |
| I.9.2. Les images en niveaux de gris | 11 |
| I.9.3. Les images couleurs | 11 |
| I.10. Formats de fichiers d'images | 11 |
| I.10.1. Windows (BitMap)..... | 12 |
| I.10.2. Tagget Image File Format (TIFF)..... | 12 |
| I.10.3. Joint Photographic Expert Group (JPEG)..... | 12 |
| I.10.4. Graphics Interchange Format (GIF)..... | 12 |
| I.10.5. Portable Network Graphic (PNG)..... | 12 |
| I.11. Quelques traitements d'images classiques..... | 13 |
| I.11.1. Binarisation | 13 |
| I.11.2. Convolution..... | 14 |
| I.11.3. Filtrage | 14 |
| I.12. Conclusion | 15 |

Chapitre II : La logique Floue en Traitement d'Images

| | |
|--------------------------------------------------------|----|
| II.1. Introduction | 17 |
| II.2. Classification..... | 17 |
| II.3. Méthodes de classification | 17 |
| II.4. Méthodes supervisées | 18 |
| II.4.1. Algorithmes de Bayes..... | 19 |
| II.4.2. Algorithmes K plus proche voisins (K-PPV)..... | 19 |
| II.4.2.1. Exemple introductif | 20 |

SOMMAIRE

| | |
|-----------------------------------------------------------------------|----|
| II.4.2.2. Quelle distance ? | 20 |
| II.4.2.3 Esquisse algorithmique pour $k = 1$ | 20 |
| II.4.3. Méthodes basées sur les réseaux de neurones artificiels | 22 |
| II.5. Méthodes non supervisées | 23 |
| II.6. Algorithmes des C-moyenne | 24 |
| II.7. Algorithme K-Means | 25 |
| II.8. Méthodes des Fuzzy C-Means (FCM)..... | 26 |
| II.9. Démarche pour l'implémentation du FCM..... | 29 |
| II.10. Systèmes neuro-flous..... | 30 |
| II.10.1. Introduction | 30 |
| II.10.2. Notions fondamentales sur les réseaux de neurones | 31 |
| II.10.2.1. Introduction | 31 |
| II.10.2.2. Neurone biologique | 31 |
| II.10.2.3. Neurone artificiel | 32 |
| II.11. L'apprentissage..... | 33 |
| II.12. Types d'apprentissages..... | 33 |
| a. Apprentissage supervisé | 33 |
| b. Apprentissage non supervisé | 34 |
| c. Apprentissage auto-supervisé..... | 34 |
| II.13. Cartes auto-organisatrices de KOHONEN | 34 |
| II.14. Méthode hybride neuro-floue | 36 |
| II.15. Conclusion | 38 |

SOMMAIRE

Chapitre III : Conception et Réalisation

| | |
|-------------------------------------------|----|
| III.1. Introduction | 40 |
| III.2. L'environnement de travail | 40 |
| III.2.1. Histoire | 40 |
| III.2.2. Syntaxe | 41 |
| III.2.3. Variables..... | 41 |
| III.2.4. Vecteurs et matrices..... | 41 |
| III.2.5. Outils et modules associés | 42 |
| III.2.6. Alternatives..... | 42 |
| III.3. Présentation des algorithmes | 43 |
| III.3.1. Algorithme k-means | 43 |
| III.3.2. Algorithme FCM..... | 49 |
| III.4. Conclusion | 52 |

Chapitre IV : Expérimentation

| | |
|------------------------------------------|----|
| IV.1. Introduction | 54 |
| IV.2. Les données utilisées..... | 54 |
| Présentation des données SX de SPOT..... | 54 |
| IV.3. Présentation des résultats | 56 |
| IV.3.1. K-means | 57 |
| IV.3.2. FCM | 58 |
| IV.4. Conclusion | 59 |
| <i>Conclusion générale</i> | 62 |

SOMMAIRE

| | |
|-----------------------------------|----|
| <i>Bibliographie</i> | 65 |
|-----------------------------------|----|

Introduction générale

La classification (clustering) est une technique mathématique d'analyse de données pour faciliter l'étude d'une population d'effectif important, particulièrement dans le domaine d'occupation des sols. Le traitement d'images numériques est actuellement en plein progrès, il est d'utilité incontestable dans de nombreuses disciplines scientifiques et industrielles particulièrement, dans le domaine de l'investigation et de la connaissance du milieu naturel.

L'analyse d'images a pour but la description des objets et scènes d'images, et donc l'extraction automatique des informations qualitatives (quantitative) et la discrimination des divers composants de l'image. Ces techniques employées, dépendent essentiellement de la nature des images à analyser et des objectifs à atteindre.

La classification consiste à attribuer une forme représentée par un vecteur à une classe parmi d'autres, elle est une étape importante dans plusieurs autres tâches accomplies dans le traitement de signal. La classification est un problème central de la reconnaissance de formes. De nombreuses méthodes dites classiques ont été consacrées telles que : arbres de décision, approche Bayésienne, K plus proche voisin, réseaux de neurones artificiels, etc.. Elles ont donné lieu au développement de plusieurs applications. Toutefois, certaines difficultés n'ont toujours pas reçues de solutions satisfaisantes.

En effet, durant les dernières décennies, la production de documents thématiques par l'intermédiaire des images issues des satellites d'observation de la terre a subi d'importantes évolutions en réponse à une demande de plus en plus croissante pour des besoins scientifiques et civils.

Toutes ces méthodes classiques sont utilisées pour faire une classification dure, basée sur le principe : qu'une forme appartient à une classe en d'autres termes, une forme est un membre complet ou n'est pas d'une classe donnée d'informations. Cette logique de traitement ne représente pas la meilleure manière de traiter des données qui sont en général mélangées et imprécises en nature. Ainsi, l'introduction de la logique floue a apporté la classification floue, cette dernière estime la contribution de chaque classe dans la forme. Elle suppose qu'une forme est une unité décomposable et par conséquent, travaille sur un nouveau principe qui suppose qu'une forme appartient à

Introduction Générale

plusieurs classes pour fournir plus d'informations sur la forme, contrairement à la classification dure qui est généralement pauvre en extraction d'informations.

L'objectif de ce mémoire est justement de tester l'approche floue pour la classification des données multispectrales XS de Spot et ce, pour tenter de cartographier les différents thèmes d'occupation des sols.

Le mémoire est organisé en 4 chapitres qui nous permettront de présenter les différents aspects de notre travail.

Dans le premier chapitre introductif, nous avons abordé des notions générales sur le traitement d'images, en s'appuyant sur les caractéristiques des images numérique et leurs supports de stockage.

Le deuxième chapitre sera spécialement consacré aux méthodes de classification en mettant l'accent sur les techniques des approches floues et neuro-floues.

Dans le troisième chapitre, nous avons abordé les techniques de programmation et ce, pour l'implémentation des algorithmes développés.

Le dernier chapitre est réservé d'abord à la présentation de la zone d'étude et des données utilisées puis, nous avons abordé une synthèse des résultats obtenus et ce, en mettant l'accent sur l'évaluation des approches développées.

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

Chapitre I

Généralités sur le

Traitement d'Images.

I.1. Introduction

La classification est une activité mentale qui intervient fréquemment dans la vie courante, dans bon nombre de disciplines. Les articles traitant de classification mettent en jeu soit des nouvelles techniques, soit des versions adaptées à un problème bien spécifique. Il est évident qu'aucune de ces techniques ne peut prétendre à une efficacité universelle. En effet, les objets sont souvent répertoriés par rapport à des catégories ou des classes auxquelles ils sont censés appartenir [1].

Les approches basées sur les ensembles flous constituent des caractéristiques importantes dans le domaine de l'analyse d'images. Dans ce chapitre, nous présenterons quelques principaux concepts de traitement d'images qui sont en relation étroites avec notre sujet d'étude.

I.2. Classification automatique

En quelques mots, la classification automatique est la tâche qui consiste à regrouper, de façon supervisée ou non supervisée (c.-à-d, sans l'aide préalable d'un expert), un ensemble d'entités élémentaires ou plus largement de données, de telle manière que les objets d'un même groupe (appelé cluster) sont plus proches au sens d'un critère de (dis) similarité choisi les uns au autres que celles des autres groupes (clusters). Il s'agit d'une tâche principale dans la fouille exploratoire de données, et une technique d'analyse statistique des données très utilisée dans de nombreux domaines, y compris l'apprentissage automatique, la reconnaissance de formes, le traitement de signal et d'images, la recherche d'information, etc. L'idée est donc de découvrir des groupes au sein des données, de façon automatique.

I.3. Quelles données ?

Les données traitées en classification automatique peuvent être des images, signaux, textes, autres types de mesures, etc.. Dans le cadre de ce projet, les données seront des données multidimensionnelles, par exemple une image couleur. Chaque donnée est donc composée de plusieurs variables (descripteurs). Pour le cas de données multidimensionnelles standard, chaque donnée étant de dimension d (donc un point dans l'espace \mathbb{R}^d) et éventuellement étiquetée (dans le cas où l'on connaîtrait sa classe d'appartenance (le "label")) et peut donc être modélisée, par exemple, par une structure contenant au moins les coordonnées du point et le champ "label". Les n données peuvent donc être modélisées

comme étant un tableau de n éléments, chaque élément du tableau étant une structure "donnée" comme décrite précédemment. Dans le cas d'une image (couleur), l'image contenant n lignes et m colonnes et donc $n \times m$ pixels couleurs, chaque pixel est composé de trois composantes RVB, peut être modélisée par un tableau de $n \times m$ structures. Chaque structure "pixel" est composée au moins des champs, couleur et "label".

I.4. Qu'est ce qu'une classe ?

En gros, une classe (ou groupe) est un ensemble de données formée par des données homogènes qui "se ressemblent" au sens d'un critère de similarité (distance, densité de probabilité, etc.).

Par exemple, si on a une base d'image de chiffres manuscrits, on aura la classe des zéros, la classe des uns, etc. De même pour le cas d'images de lettres manuscrites. Une classe peut être aussi une région dans une image couleur, un événement particulier dans un signal sonore, la classe spam et classes non spam dans le cas de détection de spam dans un mail, etc..

I.5. Combien de classes ?

Le nombre de groupes (qu'on notera K), pour commencer, peut être supposé fixe (donné par l'utilisateur). C'est le cas par exemple si l'on s'intéresse à classer des images de chiffres manuscrits (nombre de classes = 10 : 0; ::; 9) ou de lettres manuscrites dont le nombre de classes = nombres de caractères de l'alphabet.

Néanmoins, il existe des critères, dits de choix de modèle, qui permettent de choisir le nombre optimal de classes dans un jeu de données. On verra par la suite comment on pourrait calculer le nombre de groupe optimal par ce type de critères.

I.6. Exemple introductif

Considérons une image multi spectrale à d bandes spectrales (canaux). Chaque image contient n pixels ($x_1; \dots; x_n$) et chaque pixel x_i contient d valeurs radiométriques. On peut donc représenter le $i_{\text{ème}}$ pixel par un vecteur x_i de dimension d : $x_i = (x_{i1}; x_{i2}; \dots x_{id})^T \in \{0, 1, 2, \dots 255\}^d$.

Si l'on connaît les classes de certains pixels, on pourra prédire les classes des autres pixels en choisissant une mesure de (dis) similarité, par exemple une simple distance, ou une mesure de probabilité, etc.. Chaque pixel à classer aura donc la classe de celui qui lui est le plus proche au sens de la mesure de (dis) similarité choisie. Ceci peut être utilisé par exemple en segmentation d'images. De manière générale, on peut représenter les données comme un ensemble de vecteurs $(x_1; \dots; x_n)$, chaque x_i est composée de d composantes réelles : $x_i = (x_{i1}; \dots; x_{ij}; \dots; x_{id})^T \in \mathbb{R}^d$.

I.7. Types d'images

I.7.1. Image matricielle (bitmap)

Une image matricielle (Fig. I.1) est une image en mode point. Le système de codage le plus universel consiste en effet à décomposer la représentation graphique, l'image, en un certain nombre de points élémentaires caractérisés par leurs coordonnées spatiales et leur couleur.

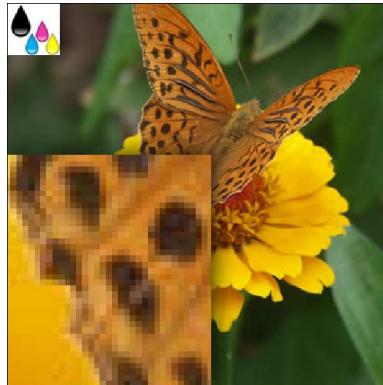


Fig. I.1. : Image matricielle.

I.7.2. Image vectorielle

Dans une image vectorielle, les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique. Il s'agit de représenter les données de l'image par des formules géométriques qui vont pouvoir être décrites d'une façon mathématique. Autrement dit, on stocke la succession d'opérations conduisant au tracé dans le

cas d'une image vectorielle, alors qu'on mémorise une mosaïque de points élémentaires dans le cas d'image matricielle.

Ces images présentent deux avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

I.8. Les images bitmap

I.8.1. Concepts et définitions de base

I.8.1.1. Pixel

Le pixel représente le plus petit élément constitutif d'une image matricielle. Le mot pixel provient d'une abréviation de l'expression britannique « Picture Element ». La valeur numérique d'un pixel représente une intensité lumineuse.

I.8.1.2. Le codage de pixel

Pratiquement toujours, la valeur d'un pixel est un mot binaire de longueur K bits, par conséquent un pixel peut prendre l'une des valeurs de l'intervalle $[0 \dots 2^K - 1]$. La valeur K est appelée profondeur de l'image. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions (une matrice) constituant l'image finalement obtenue.

I.8.1.3. La résolution d'une image

Dans le domaine de l'imagerie numérique, la résolution est une mesure de la finesse de l'affichage ou de la capture d'une image, exprimée en nombre de pixels par unité de surface, c'est-à-dire la « densité » en pixels.

La résolution d'une image numérique s'exprime en PPI (Pixels Per Inch) ou PPP (Pixels Par Pouce). Plus la résolution d'une image est grande, plus sa qualité est meilleure (Fig. I.2) [2].



(a)

(b)

(c)

Fig. I.2. : *La résolution d'une image (a) image acquise à 256 dpi, (b) image acquise à 64 dpi, (c) image acquise à 32 dpi.*

I.8.1.4. La luminance

Le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

La luminance est le degré de luminosité des pixels de l'image. Elle est définie aussi comme l'intensité d'une source étendue dans une direction donnée, divisée par l'aire apparente de cette source dans cette même direction [2].

I.8.1.5. Le contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux régions d'une image.(Fig. I.3)

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux régions voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (I.1)$$



Originale

Contrastée

Fig. I.3. : *Image originale et la même image contrastée.*

I.8.1.6. Le bruit

Un système d'enregistrement d'images ne restitue pas l'image de manière parfaite. En effet des informations parasites s'ajoutent de manière aléatoire aux détails de la scène d'origine, c'est ce qu'on appelle : bruit.

Le bruit n'a pas les mêmes origines, mais provoque des effets assez semblables comme la perte de netteté sur les détails ou l'apparition de grains.

I.8.1.7. L'histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui associe à chaque valeur d'intensité le nombre de pixels de l'image ayant cette valeur.(Fig. I.4)

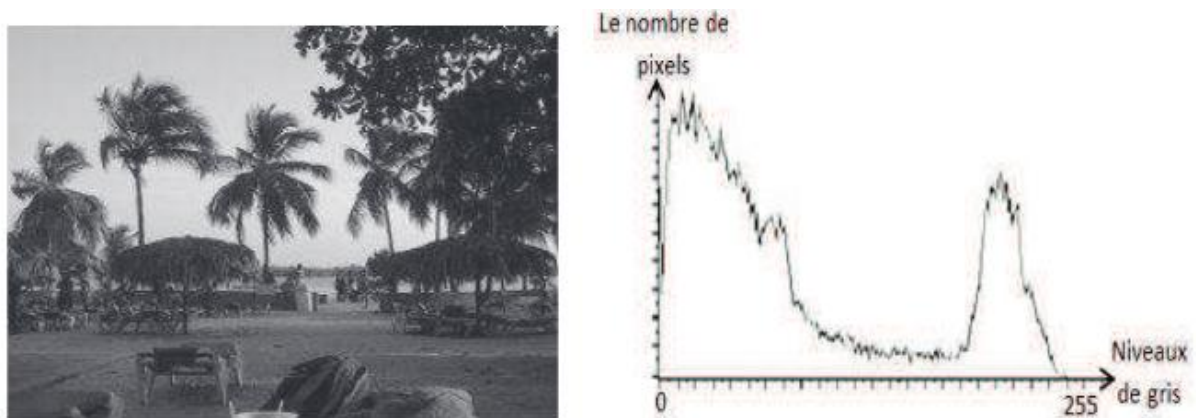


Fig. I.4. : Exemple d'histogramme d'une image.

I.8.1.8. La palette des couleurs

On appelle la palette des couleurs, la liste des couleurs que peut contenir une image.

La valeur de chaque pixel représente le rang d'une couleur dans cette liste.

Il est fréquent de voir des images qui n'utilisent jamais certaines couleurs, il devient dès lors intéressant de limiter la palette de couleur en ne sélectionnant que la ou les couleurs utilisées réellement par l'image (Fig. I.5).

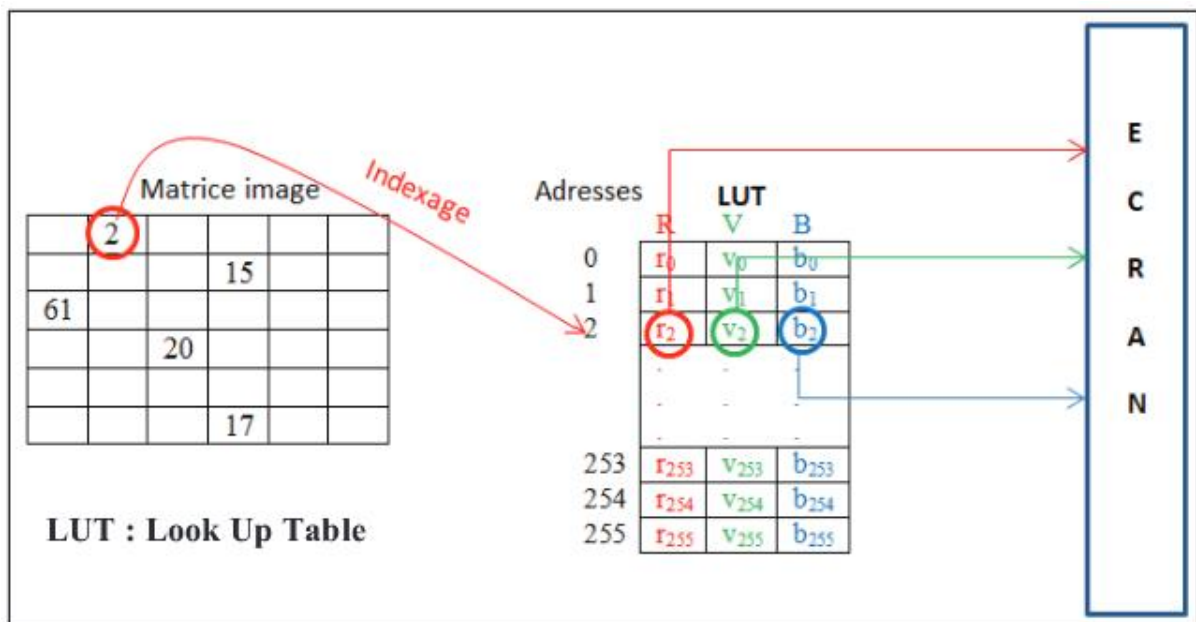


Fig. I.5. : La transformation ponctuelle d'une image numérique.

I.9. Le codage des couleurs

I.9.1. Les images binaires (noir ou blanc)

Les images binaires sont des images de profondeur $K=1$ bit, donc un pixel peut prendre l'une des valeurs : noir ou blanc (0 ou 1).

C'est typiquement le type d'images que l'on utilise pour scanner du texte quand celui-ci est composé d'une seule couleur.

I.9.2. Les images en niveaux de gris

En général, les images en niveaux de gris sont des images de profondeur $k=8$ bits, donc chaque pixel peut prendre l'une des valeurs de l'intervalle $[0 \dots 255]$, où la valeur 0 représente la brillance minimale (le noir) et 255 la brillance maximale (le blanc). Ce type d'image est fréquemment utilisé pour reproduire des photos en noir et blanc ou du texte.

Dans plusieurs applications professionnelles de photographie et d'impression ainsi qu'en médecine et astronomie, 8 bits par pixel n'est pas suffisant, pour cela il existe d'autres types d'images en niveaux de gris de profondeur $K=12$, $K=14$ ou $K=16$ bits.

I.9.3. Les images couleurs

L'espace couleur est basé sur la synthèse additive des couleurs, c'est-à-dire que le mélange entre différentes couleurs (trois, quatre...) donne une couleur.

La plupart des images couleurs sont basées sur trois couleurs primaires : Rouge, Vert et Bleu (RVB) (RGB en anglais), et utilisent typiquement 8 bits pour chaque composante de couleur, donc chaque pixel nécessite $3 \times 8 = 24$ bits pour coder les trois composantes, et chaque composante de couleur peut prendre l'une des valeurs de l'intervalle $[0 \dots 255]$.

I.10. Formats de fichiers d'images

Un format d'image est une représentation informatique de l'image, incluant des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

La plupart des formats sont composés d'un en-tête contenant des attributs (dimension

de l'image, type de codage, LUT, etc.), suivi des données (l'image proprement dite). La structuration des attributs et des données diffère d'un format à un autre. Il existe plusieurs formats d'images, nous allons mentionner quelques-uns :

I.10.1. Windows BitMaP (BMP)

Le format BMP est l'un des formats les plus simples. Il a été développé conjointement par Microsoft et IBM. Cette technologie a pour principal avantage la qualité des images fournies pas de compression (pas de perte de qualité). Cela fait de lui un format d'image très lourd, pas ou peu utilisé sur Internet.

I.10.2. Tagged Image File Format (TIFF)

Ce format est orienté vers les professionnels (imprimeurs, publicitaires...) car il a l'avantage d'être reconnu sur tous types de système d'exploitation : Windows, Mac, Linux, Unix, ... etc.

Il permet d'obtenir une image de très bonne qualité, mais sa taille reste volumineuse, même si elle est inférieure à celle du fichier BMP.

I.10.3. Joint Photographic Expert Group (JPEG)

C'est le format le plus courant, on le rencontre sur Internet. Il occupe peu d'espace disque. C'est le format développé par les photographes pour transmettre des images de qualité photographique professionnelle. Il gère des millions de couleurs, mais il ne possède pas de palette de couleurs associée et donc les couleurs peuvent être différentes sur des machines et des systèmes différents.

I.10.4. Graphics Interchange Format (GIF)

Les fichiers au format GIF sont fortement compressés tout en gardant une qualité très correcte. Ils possèdent une palette de couleurs associée (limité à 256 couleurs) et occupent peu d'espace disque.

I.10.5. Portable Network Graphic (PNG)

Le format PNG reprend le principe de codage du format GIF mais n'est pas limité à 256 couleurs, et offre une compression généralement plus efficace. Il permet donc contrairement à

GIF d'enregistrer des photographies sans perte de qualité, mais avec un gain d'espace de stockage moindre comparativement au format JPEG.

Nous avons utilisés dans notre système des images de format BMP, Pour cela, nous allons décrire la structure d'un fichier BMP.

I.11. Quelques traitements d'images classiques

Il existe une variété de traitements d'images, nous allons présenter quelques-uns :

I.11.1. Binarisation

La binarisation (le seuillage) est la technique de classification la plus simple, où les pixels de l'image sont partagés par un seul seuil S en deux classes : ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et S est retenue et l'autre non, ou vice-versa.

Soit l'image $I (M \times N)$, supposons que $f(x,y)$ représente le niveau de gris du pixel aux coordonnées (x,y) , $0 \leq x < M$, $0 \leq y < N$ et S est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à S et les autres ayant le niveau de gris supérieur à S sont des pixels du fond. Alors, l'image binarisée G est déterminée par les pixels (x,y) dont la valeur est :

$$g(x,y) = \begin{cases} 0 & \text{si } f(x,y) < S \\ 255 & \text{si } f(x,y) \geq S \end{cases} \quad (I.2)$$

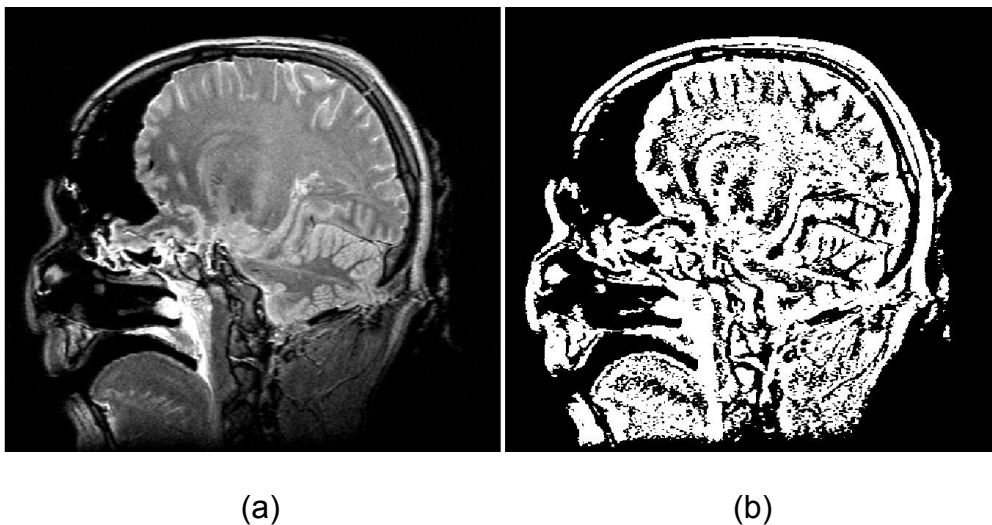


Fig. I.6. : Binarisation d'une image (a) image originale, (b) image binarisée.

I.11.2. Convolution

La convolution est le remplacement de la valeur d'un pixel par une combinaison de ses avoisinants. Elle consiste à faire balayer une fenêtre d'analyse (masque) sur l'ensemble des pixels de l'image.

L'opération de convolution est calculée en tout point de l'image source en 3 étapes :

1. On centre le masque (par exemple un masque carré 3*3) sur le pixel courant.
2. On calcule les 9 produits entre la valeur de l'image et la valeur du masque superposé.
3. Puis on somme les 9 produits pour obtenir la valeur du pixel de l'image filtrée.

On peut résumer l'opération de convolution par la formule suivante :

$$S(i,j) = \sum_{k=-(N-1)/2}^{(N-1)/2} \sum_{l=-(N-1)/2}^{(N-1)/2} E_{i+k,j+l} \cdot h_{k,l} \quad (I.3)$$

Où E est l'image d'entrée, h le masque de convolution avec $h(0,0)$ le centre du masque et N la taille du filtre (dans notre exemple $N=3$).

I.11.3. Filtrage

La notion de filtre est empruntée à la physique et aux techniques de traitement de signal. Si un signal (électrique, radio, image,...etc.) présente des composantes de fréquences très différentes, il peut être avantageux d'en éliminer certaines, en parle dans ce cas de filtrage.

Le filtrage des images est un traitement local utilisé principalement pour réaliser une analyse spatiale d'une image. Son objectif est d'accentuer les variances d'intensité de l'image, ou de détecter les contours et de réduire les bruits existants.

Il existe un grand nombre de filtres, on peut les classer en deux grandes catégories : les filtres linéaires et les filtres non linéaires.

I.12. Conclusion

L'analyse d'images nécessite la compréhension et la maîtrise d'un ensemble d'informations pertinentes liées aux structures de données des images et surtout, les traitements de bases qui en découlent. Dans ce chapitre, nous avons vu quelques notions essentielles permettant la caractérisation d'une image et les supports utilisés. Le chapitre II sera consacré essentiellement aux techniques floues, utilisées dans la classification des images multispectrales.

Chapitre II

La logique Floue en Traitement d'Images.

II.1. Introduction

Les objets sont souvent répertoriés par rapport à des classes ou des catégories auxquelles ils sont censés appartenir. Cette appartenance est, la plupart du temps, vague et/ou graduelle. Des modificateurs linguistiques tels que très, trop, assez, insuffisamment, traduisent l'incertitude que l'on peut rencontrer concernant les caractéristiques ou les propriétés de certaines entités.

II.2. Classification

La classification consiste à attribuer à chaque pixel dans l'image une classe (étiquette). Cette affectation peut être effectuée en se basant sur des régions dont on connaît les classes d'appartenance a priori, alors, on parle de classification supervisée, ou non, dans ce cas on parle de classification non-supervisée (segmentation/clustering).

La classification peut être représentée comme une fonction μ qui associe à chaque pixel p l'indice k de la classe C_k à laquelle il appartient. De même que pour les régions R_k , le contenu des classes C_k doit être homogène (par exemple, les pixels qui composent une classe doivent avoir des valeurs f similaires).

Notant que les problèmes de segmentation et de classification sont très liés et peuvent être utilisés pour signifier la même chose : Un classifieur implicitement segmente une image : l'ensemble des pixels ayant la même classe forme une région de l'image. Une segmentation implique une classification (les régions sont étiquetées selon leurs appartenances)

II.3. Méthodes de classification

D'un point de vue général, les méthodes de classification ont pour but de regrouper les éléments d'un ensemble $X = \{X_1, \dots, X_n, \dots, X_N\}$ en un nombre C optimal de classes selon leurs ressemblances [4].

De manière générale, les problèmes de classification s'attachent à déterminer des procédures permettant d'associer un objet (individu) à une classe. Ces problèmes se déclinent

essentiellement en deux variantes selon Bezdek [5] : la classification dite " supervisée " et la classification dite " non supervisée ".

La classification, supervisée ou non, en tant que discipline scientifique, n'a été automatisée et massivement appliquée que relativement récemment. Comme la plupart des activités scientifiques, l'essor des différentes techniques de classification a largement bénéficié de l'avènement et du perfectionnement des outils informatiques. De nos jours, la classification est une démarche qui est appliquée dans d'innombrables domaines. Un autre nom possible pour cette branche de la recherche est la typologie, et la science qui lui est associée est la taxinomie.

Les méthodes de classification ont pour but de regrouper les éléments d'un ensemble X , de nature quelconque, en un nombre restreint de classes. La qualité de la classification peut être jugée sur la base des deux critères suivants :

- Les classes générées doivent être les plus différentes possibles les unes des autres vis-à-vis de certaines caractéristiques ;
- Chaque classe doit être la plus homogène possible vis-à-vis de ces caractéristiques [6].

Si l'on se place dans un espace métrique, en termes d'inertie, le premier critère peut être traduit par le fait que l'inertie interclasse doit être la plus élevée possible, et le deuxième critère traduit la préoccupation d'obtenir une inertie intra-classe la plus petite possible.

II.4. Méthodes supervisées

Ces sont des méthodes dans lesquelles les classes sont connues *a priori* avant d'effectuer l'opération d'identification des éléments de l'image. Elles demandent une **phase d'apprentissage** sur l'échantillon représentatif dans le but d'apprendre les caractéristiques de chaque classe et une autre phase pour décider l'appartenance d'un individu à une classe spécifique.

Nous pouvons donc citer des algorithmes de classification pixels supervisés comme celui de Bayes [7], Les K plus proches voisins (K-PPV) ou les réseaux de Neurones Multicouches.

II.4.1. Algorithmes de Bayes

Ce sont des méthodes qui reposent sur la théorie de Bayes représentant une référence théorique pour les approches statistiques de résolution des problèmes de classification. Le principe de cette théorie est le suivant :

Soit X_j le vecteur forme dont la classe est inconnue et qu'on veut la déterminer, et soit C_i une hypothèse (X_j appartient à la classe C_i par exemple). On cherche à déterminer $P(C_i/X_j)$ la probabilité de vérification de C_i après l'observation de X_j . $P(C_i/X_j)$ est la probabilité postérieure (après la connaissance de X_j) tandis que $P(C_i)$ est la probabilité à priori représentant la probabilité de vérification de C_i pour n'importe quel vecteur forme.

Le théorème de Bayes propose une méthode de calcul de $P(C_i/X_j)$ en utilisant les probabilités $P(C_i)$, $P(X_j)$ et $P(X_j/C_i)$:

$$P(C_i/X_j) = \frac{P(C_i)P(X_j/C_i)}{\sum_{k=1}^C P(C_k)P(X_j/C_k)} \quad \forall i \in [1 \dots C] \quad (\text{II.1})$$

$P(C_i/X_j)$ est donc la probabilité d'appartenance de X_j à la classe C_i , $P(C_i)$ la probabilité d'apparition de la classe C_i (peut être calculée comme le rapport entre le nombre d'échantillons appartenant à la classe C_i et le nombre total d'échantillons). $P(X_j/C_i)$ peut être considéré comme la probabilité d'apparence de chaque valeur des attributs de X_j dans les attributs des échantillons appartenant à la classe C_i . Alors l'algorithme cherche à déterminer pour chaque élément X_j la classe C_i qui maximise la probabilité de contenir cet élément.

II.4.2. Algorithmes K plus proche voisins (K-PPV)

L'algorithme des k plus proches voisins (*KPPV*) (*k-Nearest-Neighbors (kNN)* en anglais) est une méthode non paramétrique et supervisée de classification introduite dans [8]. Elle est largement utilisée en classification d'une manière générale et en segmentation d'images en particulier. Elle repose sur un principe simple et intuitif de regroupement d'individus en fonction de leur voisinage.[9]

II.4.2.1. Exemple introductif

Considérons par exemple une image couleur, chaque image contient n pixels, chaque pixel contient d valeurs (RGB). On représente le i ème pixel ($i = 1, \dots, n$) par un vecteur x_i de dimension d : $x_i = (x_{i1}, \dots, x_{ij}, \dots, x_{id})^T \in \{0, 1, \dots, 255\}^d$ avec d ici égale à 3. Si l'on connaît les classes de certains pixels, on pourra prédire les classes des autres pixels en choisissant une mesure de (dis) similarité, par exemple une simple distance, ou une mesure de probabilité, etc. Chaque pixel à classes aura donc la classe de celui qui lui est le plus proche au sens de la mesure de similarité choisie.

II.4.2.2. Quelle distance ?

Afin de trouver les K plus proches d'une donnée à classer, on peut choisir la distance euclidienne.

Soient deux données représentées par deux vecteurs x_i et x_j , la distance entre ces deux données est donnée par :

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (\text{II.2})$$

II.4.2.3. Esquisse algorithmique pour $k=1$

Il s'agit de prédire les classes de nouvelles données à partir de données étiquetées (données d'apprentissage représentant les classes avec leurs échantillons). On commencera par le 1-ppv. L'algorithme 1nn est donné par le pseudo-code 1 ci-après.

Algorithm 1 : Algorithme du plus proche voisin (1-ppv)

Données : Données d'apprentissage ; $X^{train} = (x_1^{train}, \dots, x_n^{train})$; classes des données d'apprentissage $Z^{train} = (z_1^{train}, \dots, z_n^{train})$; $X^{test} = (x_1^{test}, \dots, x_m^{test},)$

Algorithme knn :

Pour i = 1 à m **faire**

Pour j =1 à n **faire**

 Calculer la distance euclidienne entre x_i^{test} et x_j^{train} en utilisant l'équation (II.2)

$$d_j = d(x_i^{test}, x_j^{train})$$

Finpour

 Calculer la classe z_i^{test} du ième exemple qui vaut la classe de son ppv :

 Trouver l'indice du ppv de x_i^{test} :

$$ind_ppv_i = \arg \min_{j=1}^n d_j$$

 Trouver la classe du ppv de x_i^{test} (qui est $x_{ind_ppv_i}^{train}$) :

$$z_i^{test} = z_{ind_ppv_i}^{train}$$

Finpour

Result : classes des données de test $Z^{test} = (z_1^{test}, \dots, z_n^{test})$

Ensuite, étendre le code pour le cas des K-ppv pour une valeur de $K \geq 1$.

L'algorithme Knn est donné par le pseudo-code 2 ci-après

Algorithm 2 : Algorithme des k-plus proches voisins

Données : Données d'apprentissage ; $X^{train} = (x_1^{train}, \dots, x_n^{train})$; classes des données d'apprentissage $Z^{train} = (z_1^{train}, \dots, z_n^{train})$; $X^{test} = (x_1^{test}, \dots, x_m^{test})$; nombre des ppv k

Algorithme knn :

Pour i = 1 à m **faire**

Pour j =1 à n **faire**

 Calculer la distance euclidienne d_{ij} entre x_i^{test} et x_j^{train} en utilisant l'équation (II.2)

$$d_j = d_{ij}$$

Finpour

```

Calculer la classe  $z_i^{test}$  du  $i$ ème exemple qui vaut la classe de son ppv :
/* trouver les k-ppv de  $x_i^{test}$  */:
Trier les distances  $d_j$  selon un ordre croissant pour  $j=1, \dots, n$ 
Récupérer les classes en même temps les indices Ind oisins avant le tri des  $d_j$ 
Récupérer les classes des k premiers ppv a partir des indices IndVoisins et en trouver
la classe majoritaire :
 $c_k = 0$  ( $k=1, \dots, k$ )
Pour  $k=1$  a  $k$  faire
    |
    |    $ind\_voisin_k = IndVoisins_k$ 
    |    $h = z_{ind\_voisin_k}^{train}$ 
    |    $c_h = z_h + 1$ 
Finpour
/*trouver la classe du ppv de  $x_i^{test}$  :
(la classe majoritaire de celles de ses k-ppv) */ :
 $z_i^{test} = \arg \max_{k=1}^K c_k$ 

```

Finpour

Result : classes des données de test $Z^{test} = (z_1^{test}, \dots, z_n^{test})$

II.4.3. Méthodes basées sur les réseaux de neurones artificiels

Un réseau de neurones est constitué d'un certain nombre d'unités élémentaires (neurones) fortement interconnectés (neurones) fonctionnant en parallèle et organisée en couches avec un flux d'informations à anticipation. L'architecture du perceptron multicouche est organisée de la manière suivante: les signaux flux successivement dans les différentes couches de l'entrée à la couche de sortie. Les couches intermédiaires sont appelées couches cachées. Pour chaque couche, chaque unité élémentaire calcule un produit scalaire entre un vecteur de poids et le vecteur de sortie donné par la couche précédente. Une fonction de transfert est alors réalisée à la suite de faire une entrée pour la couche suivante. Les poids des connexions sont ajustés au cours du processus d'apprentissage pour obtenir la relation d'entrée/sortie désirée du réseau. De nombreux modèles de réseaux existent (réseaux de Hopfield, perceptrons multicouche, etc.).

D'après des études antérieures, la notion d'apprentissage supervisé (dirigé) dans ce cas, à partir d'exemples, est attrayante et semble bien adaptée au domaine de l'image. L'expérience commune de tout un chacun suggère qu'il est aisé de reconnaître un objet après l'avoir vu une ou deux fois sous plusieurs angles.

Les réseaux de neurones artificiels (RNA) fournissent un cadre algorithmique pour les techniques d'*apprentissage statistique* [10]. Leur application à l'image constitue un domaine de recherche actif depuis une trentaine d'années.

Certaines applications, comme la reconnaissance de caractères, de chiffres manuscrits [11] ou la localisation de visages [12], confirment l'efficacité de la technique. De nombreuses architectures neuronales, classiques ou plus spécifiques, ont été proposées et testées [13]. Ces réseaux restent cependant difficiles à mettre en œuvre et les réalisations pratiques des RNA pour l'image sont encore rares.

De nouvelles approches, basées sur une large connectivité aléatoire, ont été introduites sous les termes d'*Echo State Network* (ESN) [14] en 2001 pour les systèmes dynamiques, et, plus récemment, d'*Extreme Learning*

Cependant, les méthodes de segmentation supervisée offrent l'avantage d'être plus rapides et plus reproductibles que les méthodes non supervisées. Toutefois, leur inconvénient est de rester très dépendant de la qualité de la base d'apprentissage et des connaissances apriori sur les données à traiter. C'est pourquoi il est intéressant de développer des algorithmes entièrement automatiques "*non supervisée*", qui présenteront l'avantage d'offrir un résultat reproductible et indépendant des actions de l'opérateur et des connaissances apriori sur les données sources.

II.5. Méthodes non supervisées

L'intérêt des méthodes *non supervisées* est qu'elles ne nécessitent **aucun apprentissage et aucune connaissance à priori** et par là même aucune tâche préalable d'étiquetage manuel. Elles ont pour but de découper l'espace d'individus (pixels) en zones homogènes selon un critère de ressemblance (critère de proximité de leurs vecteurs d'attributs dans l'espace de représentation entre les individus).

Nous pouvons donc citer des algorithmes de classification pixels non-supervisés comme les C-moyennes proposée par Mac Queen [15], C-moyennes floues [16] [7] [17] ou Algorithmes les C-moyennes possibilistes [18]:

II.6. Algorithme des C-moyennes

L'algorithme des C-moyennes (C-means) (centres mobiles) est l'un des algorithmes de clustering les plus connus, il a été mis au point par Mac Queen à la fin des années soixante [15]. L'algorithme des C-moyennes (C-means) [15] réalise un partitionnement automatique des pixels de l'image en C classes homogènes. Soit $X = \{X_j : j=1, \dots, N\}$ un sous-ensemble d'un espace N-dimensionnel. Une partition de X en C classes peut être représentée par des sousgroupes mutuellement disjoints $C_1 \dots C_k$, tel que : $C_1 \cup \dots \cup C_k = X$.

Chaque vecteur X_j est assigné à une et seulement une classe qui lui est la plus proche. La fonction objective correspondante s'exprime sous la forme

$$J = \sum_{i=1}^C \sum_{j=1}^N U_{ij} d^2(x_j, b_i) \quad (\text{II.3})$$

Où : b_i désigne le prototype (centre) de cette classe.

d: mesure l'inverse de la similarité entre le prototype et le vecteur de données.

U_{ij} : est un indicateur binaire désignant l'appartenance ou non du vecteur x_j à la classe

C_i tel que $U_{ij} = 1$ si $x_j \in C_i$ et $U_{ij} = 0$ sinon pour $i = 1 \dots C$. L'indicateur d'appartenance U_{ij} s'exprime comme suit :

$$U(X_j, b_i) = \begin{cases} 1 & \text{si } d^2(X_j, b_i) = \min\{d^2(X_j, b_k) \mid k = 1 \dots C\} \\ 0 & \text{sinon} \end{cases} \quad (\text{II.4})$$

L'optimisation se fait de manière itérative. Les étapes de cette classification pouvant être résumé comme suit :

1- Initialiser l'ensemble des centres avec des éléments, soit pris au hasard de l'ensemble de données, soit d'après des connaissances a priori ;

2- Assigner chaque point au centre le plus proche ;

- 3- Calculer les nouveaux centres des classes construites. Ils sont calculés par la moyenne des éléments de la classe ;
- 4- Générer une nouvelle partition de l'ensemble de données en assignant chaque point au centre le plus proche ;
- 5- S'il y a changement dans la partition, réitérer depuis l'étape 3.

II.7. Algorithme K-Means

Ce type de méthode considère une région comme un ensemble de pixels connexes appartenant à une même classe. Elle suppose donc que les pixels qui appartiennent à une même région possèdent des caractéristiques similaires et forment un nuage de points dans l'espace des attributs. La classification consiste à retrouver ces nuages de points qui correspondent aux classes des pixels présentes dans l'image.

C'est l'un des algorithmes les plus connus en classification non supervisée. Il vise à produire un partitionnement des pixels de manière à ce que les pixels d'une même classe soient semblables et les pixels issus de deux classes différentes soient dissemblables. L'idée principale est de définir C centres, un pour chaque classe. Chaque classe C_i est ainsi caractérisé par son centre noté b_i et le nombre d'éléments N_i . L'algorithme C-Means dans sa formulation originale cherche à minimiser une fonction de coût global définie par :

$$J = \sum_{i=1}^C \sum_{(x,y), k \in C_i} d^2(f(x,y), b_i) \quad (II.5)$$

Où $f(x,y)$ représente le niveau de gris du pixel de coordonnées (x,y)

Il se déroule selon les étapes suivantes :

1. Initialisation de chaque centre b_i
2. Pour chaque pixel (x,y) , calculer la distance $d(f(x,y), b_i)$ aux différents centres des classes b_i , et affecter à la classe la plus proche C_i avec $d(f(x,y), b_i) = |d(f(x,y) - b_i|$
3. Mise à jour de nombre de pixels et des centres b_i des classes $b_i = \frac{\sum_{(x,y) \in C_i} f(x,y)}{N_i}$

4. Arrêt si N_i satisfait, sinon retour à l'étape 2.

Le principal inconvénient de cette méthode est que la classification finale dépend du choix de la partition initiale. Le minimum global n'est pas obligatoirement atteint, on est seulement certain d'obtenir la meilleure partition à partir de la partition de départ choisie [28]. De nombreuses variantes peuvent être rencontrées. Par exemple, au lieu de calculer le centre des classes, après avoir affecté tout les pixels, les centres de gravité peuvent être calculés immédiatement après chaque affectation. La méthode des K-Means a été généralisée sous l'appellation de la "méthode des nuées dynamiques" [29]. Au lieu de définir une classe par un seul point (son centre de gravité), elle est définie par un groupe de points (noyaux de classe). Un autre algorithme proposé dans la littérature et qui est issu de l'algorithme K-Means est l'algorithme ISODATA [30]. L'avantage de ce dernier est qu'il permet de regrouper les pixels sans connaître a priori le nombre exact de classes présentes dans l'image. Ce nombre pourra être modifié au cours des itérations.

II.8. Méthode des Fuzzy C-Means (FCM)

Nous rappelons que la classification non supervisée (*clustering en anglais*) consiste à segmenter un ensemble de vecteurs non labellisés (la classe n'est pas fournie) en groupes (*clusters*) qui possèdent les propriétés suivantes :

- *Homogénéité* dans les groupes, c'est-à-dire, les données appartenant à un même cluster doivent être les plus similaires possibles ;
- *Hétérogénéité* entre groupe : les données appartenant à différents clusters doivent être les plus dissemblables possibles.

Les notions de similarité / "dissimilarité" seront abordées dans le paragraphe suivant.

Un exemple illustratif d'une classification non supervisée est donné sur la figure suivante. Les vecteurs d'entrée sont décrits sur la figure (a) et les clusters désirés sur la figure (b).

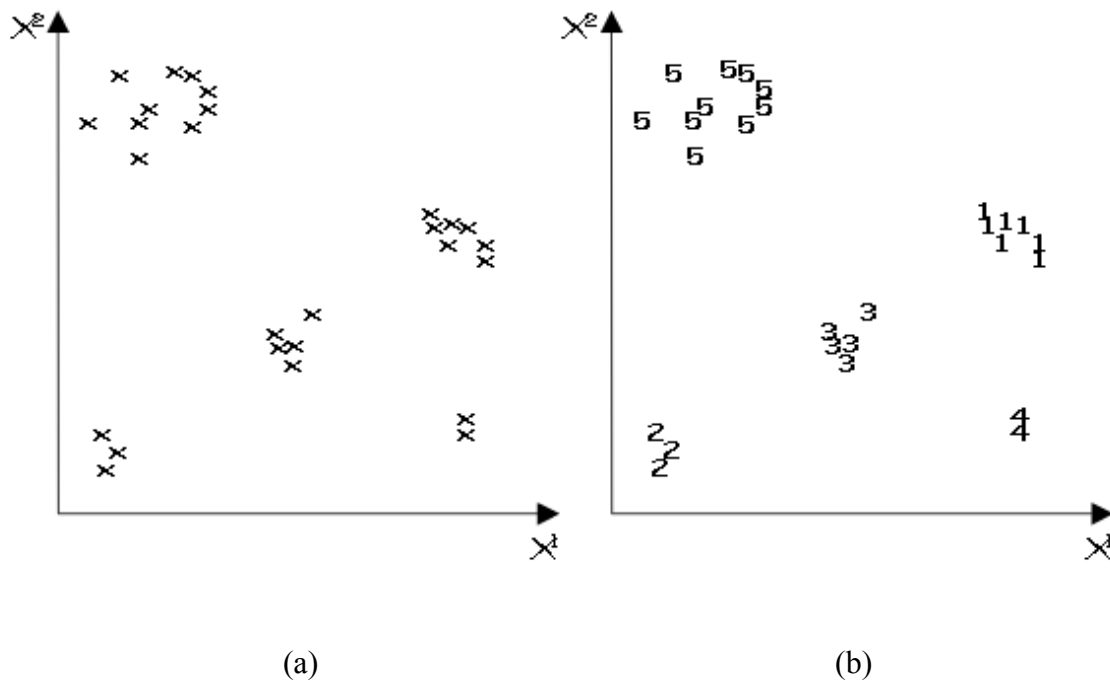


Fig.II.1 *Principe du clustering.*

Cependant, il existe une approche floue du clustering. Dans ce cas, les sous-ensembles flous permettent une représentation simple des incertitudes et imprécisions liées aux informations et aux connaissances. Son principal avantage est d'introduire le concept d'appartenance graduelle à un ensemble alors qu'en logique ensembliste classique, cette appartenance est binaire (appartient ou n'appartient pas à un ensemble).

En 1981, Besdek a proposé une version floue du K-Means : l'algorithme du Fuzzy C-means (FCM) [32]. La modélisation de l'imprécision s'effectue en considérant des frontières graduelles au lieu de frontières nettes entre les classes. L'incertitude s'exprime par le fait qu'un pixel possède aussi bien des attributs qui l'assignent à une classe qu'à une autre. La classification floue assigne donc, non pas à un pixel une étiquette relative à une classe unique, mais son degré d'appartenance à chacune des classes. Ces valeurs expriment l'appartenance incertaine d'un pixel à une région et sont appelées degrés d'appartenance. Le degré d'appartenance se situe dans l'intervalle $[0, 1]$ et les classes obtenues ne sont pas forcément disjointes. Dans ce cas, les données X_j ne sont plus assignées à une classe unique, mais à plusieurs par l'intermédiaire de degrés d'appartenance U_{ij} du vecteur X_j à la classe i . Le but des algorithmes de classification est non seulement de calculer les centres de classe v mais aussi l'ensemble des degrés d'appartenance des vecteurs aux classes.

Soit E un ensemble de vecteurs d'attributs, $\{V_1, V_2, \dots, V_c\}$ les centres des différentes classes où c dénote le nombre de classes. Le degré d'appartenance d'un élément x_k à un groupe V_i est noté $\mu_i(x_k)$. Dans cette approche, un vecteur d'attribut donné peut appartenir à plusieurs groupes.

U est la matrice des degrés d'appartenance (appelée également matrice de C-partition floue) de dimension $c \times n$ où c est le nombre de classes et n le nombre d'éléments à classer.

L'algorithme de classification non supervisé des FCM est une méthode de partitionnement d'ensemble basé sur l'itération de Picard à travers les conditions nécessaires pour optimiser une somme d'erreurs au carré d'une fonction objective (J_m) qui est construite à base de la somme des distances entre le vecteur et le centre des classes pondérée par les fonctions d'appartenances :

$$J_m(U, V) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d^2(x_k, V_i) \quad (\text{II.6})$$

Où : $d = \|x_k - V_i\|$ est la norme Euclidienne et V_i représente le centre de la classe i .

Le paramètre $m > 1$ [32] contrôle le degré de flou.

$$\forall i, j \quad \mu(x_j) \in \{0,1\} \quad U = [\mu_{ik}(x_k)] \quad (\text{II.7})$$

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{c1} & \cdots & \cdots & \mu_{cn} \end{bmatrix} \quad (\text{II.8})$$

Deux contraintes sont définies pour la matrice des degrés d'appartenance U .

$$0 \leq \mu_{ik} \leq 1 \quad 1 \leq i \leq c \quad 1 \leq k \leq n$$

$$\sum_{i=1}^c \mu_{ik} = 1, \quad \forall k \in [1, n]$$

Le FCM est basé sur la mise à jour de la fonction d'appartenance pendant l'itération de l'algorithme donc le FCM fait ainsi évoluer la partition (matrice U) en minimisant la fonction objective J_m .

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad (\text{II.9})$$

Où μ_{ik} est la fonction de mise à jour des degrés d'appartenance (avec $d_{ik} = \|x_i - c_k\|$).

La fonction de mise à jour des centres est :

$$V_i = \frac{\sum_{k=1}^n \mu_{ik}^m x_k}{\sum_{k=1}^n \mu_{ik}^m} \quad (\text{II.10})$$

Le test d'arrêt au temps $t + 1$ est donné par la différence $|J_m^{t+1} - J_m^t| < \varepsilon$ (*Seuil*).

II.9. Démarche pour l'implémentation du FCM

Etape 1 :

- Choisir le nombre de classe : c
- Fixer m , $1 < m < +\infty$
- Choisir ε pour le test d'arrêt
- Initialiser la matrice de partition à zéro $U^t = 0$ des fonctions d'appartenance à des valeurs aléatoires ou bien en leurs affectant comme valeurs $1/c$
- Choisir la mesure d_{ik} on prend par exemple $d_{ik} = \|x_i - c_k\|$

Etape 2 :

Calculer le vecteur C-Mean V_i des centres en utilisant l'équation (II.9).

Etape 3 :

Mettre à jour U^t en utilisant :

$$U_{ik}^{t+1} = \frac{1}{\sum_{j=1}^C \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \quad (\text{II.11})$$

Comparer J^{t+1} à J^t si $|J_m^{t+1} - J_m^t| < \varepsilon$ alors arrêt sinon $t = t + 1$ et retour à l'étape 2.

II.10. Systèmes Neuro-Flous**II.10.1. Introduction**

Les réseaux de neurones artificiels (Artificial Neural Network) sont appliqués aux différents domaines. Le traitement parallèle et les caractéristiques non linéaires qu'ils offrent font que les ANN sont beaucoup utilisés en classification d'images et en clustering. La carte auto-organisatrice de Kohonen « SOM » est un outil puissant pour le partitionnement des données [19]. SOM possède la propriété de préserver la topologie des données ainsi que la relation de distance entre elles. L'espace des données est projetée sur une grille régulière dont la dimension constitue le nombre de classes pour le FCM.

Nous allons donner une méthode de segmentation hybride, basée sur une information globale fournie par le partitionnement flou dont le nombre de classes sera optimisé par l'algorithme des cartes auto-organisatrices de Kohonen [20] et une information locale fournie par la moyenne et l'écart type calculés sur une fenêtre. En fait, c'est la fonction objective de l'algorithme des C-moyennes floues qui a été modifiée par l'ajout d'un terme de pénalisation pour garantir une distribution uniforme des centres de classes [21].

Après une brève présentation de quelques notions fondamentales sur les réseaux de neurones, nous introduirons les cartes auto-organisatrices de Kohonen pour formaliser la méthode hybride proposée.

II.10.2. Notions fondamentales sur les réseaux de neurones

II.10.2.1. Introduction

Les réseaux de neurones artificiels (RNA) sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Ils infèrent des propriétés émergentes permettant de solutionner les problèmes complexes.

II.10.2.2. Neurone biologique

Le neurone biologique est le composant cellulaire de base du cerveau (voir Fig. II.2), il comprend :

- Le corps cellulaire, qui fait la somme des influx par intermédiaire de l'axone ;
- L'axone qui permet de transmettre les signaux émis par le corps cellulaire aux autres neurones ;
- Les dendrites, qui sont les récepteurs principaux du neurone, captant les signaux qui lui parviennent ;
- Les synapses qui permettent aux neurones de communiquer avec les autres via les axones et les dendrites.

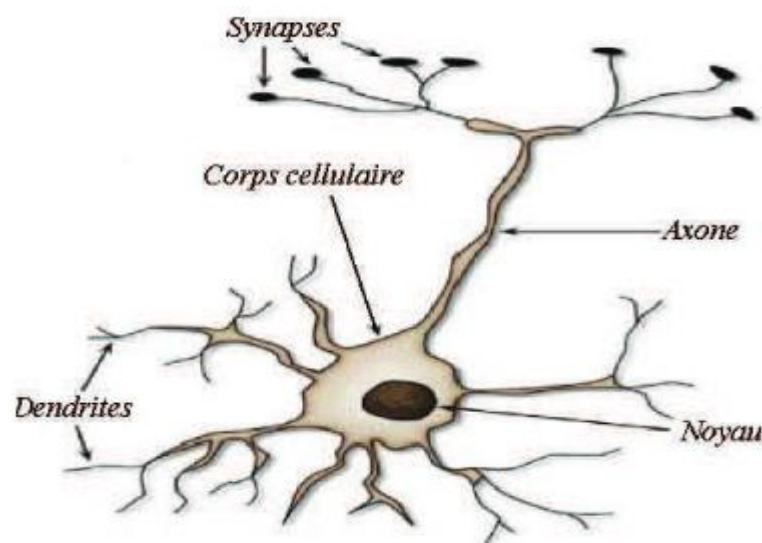


Fig.II.2. : Neurone biologique.

II.10.2.3. Neurone artificiel

Le neurone artificiel est un modèle mathématique simplifié du neurone biologique, il présente un certain nombre d'entrées, un corps traitant les entrées, et un axone véhiculant la réponse du neurone.

La Figure (Fig. II.3) montre la structure d'un neurone artificiel. Chaque neurone reçoit un nombre variable d'entrées en provenance des neurones amont. A chacune de ces entrées est associé un poids W_i abréviation de Weight (poids en Anglais) représentatif de l'entrée X_0 .

Chaque neurone élémentaire est doté d'une fonction de transfert (fonction d'activation) qui donne une sortie unique Y , et se ramifie ensuite pour alimenter un nombre variable de neurones avals [22].

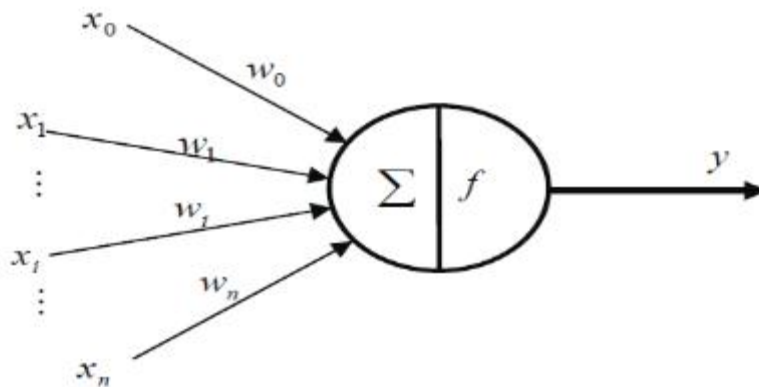


Fig.II.3. : *Neurone artificiel* [23].

$$z = \sum_{i=0}^n w_i x_i \quad \text{et} \quad y = f(z) \quad (\text{II.12})$$

Les fonctions d'activations les plus utilisées sont les suivantes : (voir Fig. II.4)

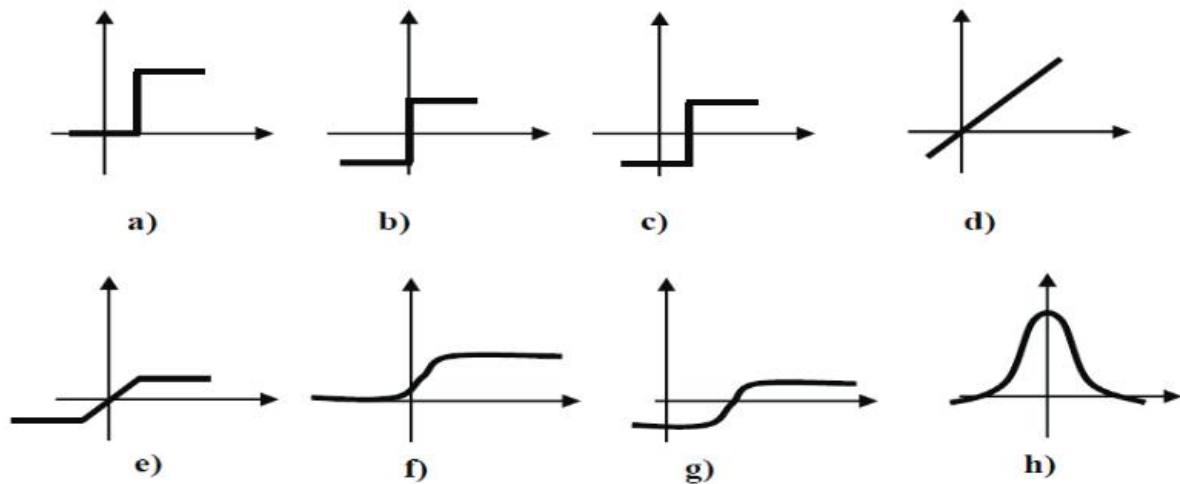


Fig. II.4. : *fonctions d'activations les plus utilisées.* a) tout ou rien ; b) fonction signe ; c) plus ou moins à seuil ; d) fonction affine ; e) saturation ; f) sigmoïde ; g) fonction arc tangente ; h) fonction gaussienne. [23]

II.11. L'apprentissage

L'apprentissage dans le contexte des réseaux de neurones, est le processus de modification des poids de connexions (y compris les biais) ou plus rarement du nombre de couches et de neurones, afin d'adapter le traitement effectué par le réseau à une tâche particulière [24].

II.12. Types d'apprentissages

On distingue trois familles d'apprentissage :

a. Apprentissage supervisé

Un superviseur, ou professeur, fournit au réseau des couples d'entrées-sorties. Il fait apprendre au réseau l'ensemble de ces couples, par une méthode d'apprentissage, comme la rétro-propagation du gradient de l'erreur, en comparant pour chacun d'entre eux la sortie effective du réseau et la sortie désirée. L'apprentissage est terminé lorsque tous les couples entrées-sorties sont reconnus par le réseau.

b. Apprentissage non supervisé

Cet apprentissage consiste à détecter automatiquement des régularités qui figurent dans les exemples présentés et à modifier les poids des connexions pour que les exemples ayant les mêmes caractéristiques de régularité provoquent la même sortie. Les réseaux autoorganiseurs de Kohonen [25] sont les réseaux à apprentissage non supervisé les plus connus.

c. Apprentissage auto-supervisé

Le réseau de neurones évalue lui-même ses performances, sans l'aide d'un « expert », un objet est présenté à l'entrée du réseau de neurones, à qui on a indiqué la classe à laquelle appartient cet objet. Si le réseau ne le classe pas correctement, il mesure lui-même l'erreur, et propage cette erreur vers l'entrée. Le réseau procède à autant d'itérations qu'il est nécessaire jusqu'à obtenir la réponse valide.

II.13. Cartes auto-organisatrices de KOHONEN

Le réseau de Kohonen [26] (SOM) est un réseau de neurones artificiels (ANN) non supervisé. Il est constitué (Fig. II.5) :

- d'une couche d'entrée où tout individu à classer est représenté par un vecteur multidimensionnel ; soit $x \in R^n$. A chaque individu est affecté un neurone qui représente le centre de la classe ;
- d'une couche de sortie ou de compétitions.

Tous les neurones d'entrée sont connectés aux neurones de sortie qui répondent différemment à la forme de l'entrée. Les neurones de cette couche entrent en compétition. A chaque étape, dans la phase d'apprentissage, le neurone qui se rapproche le plus de l'entrée est élu « gagnant » (dans le sens du minimum de la distance euclidienne).

$$\|x(t) - m_c(t)\| = \min_i \{\|x(t) - m_i(t)\|\} \quad (\text{II.13})$$

$m_c(t)$: neurone gagnant

A chaque neurone, on fait correspondre un espace de sortie qui, en général est un espace 2D défini par une grille ou un voisinage rectangulaire ou hexagonale $N_c(t)$.

Le neurone gagnant et son voisinage sont mis à jour de façon que les poids associés soient proches de l'entrée. Le facteur de mise à jour ou facteur de gain n'est pas égale pour tous les neurones mais, sa plus grande valeur se situe près du neurone gagnant et décroît en fonction du voisinage. Avec le temps, le facteur de gain doit décroître ainsi que la taille de chaque voisinage.

$$m_i(t+1) = m_i(t) + \alpha(t)[x_i(t) - m_i(t)] \quad \forall i \in N_c(t) \quad (\text{II.14})$$

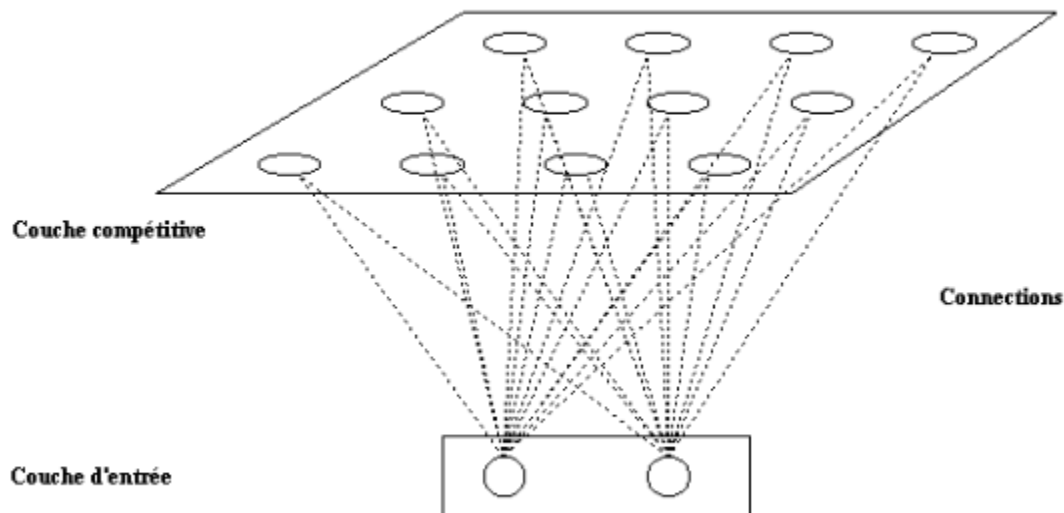


Fig.II.5. : Carte auto_organisatrice de Kohonen.

Durant la phase d'apprentissage, les poids des neurones changent de manière ordonnée de telle façon que les caractéristiques dans une image tendent à s'organiser en fonction de la distribution topologique dans le réseau. Les neurones adjacents répondent de manière similaire alors que les neurones distants ont des réponses divergentes. La convergence des caractéristiques s'effectue en imposant des contraintes au facteur de gain α pendant la mise à jour des poids.

II.14. Méthode hybride neuro-floue

L'information fournie par l'intensité ou la couleur d'un pixel ne suffit pas pour segmenter correctement une image. A cet attribut, nous avons rajouté deux autres paramètres évalués sur une fenêtre du voisinage du pixel. Ces paramètres sont la moyenne et l'écart type. Ainsi, pour chaque pixel, la couche d'entrée de la carte auto organisatrice de Kohonen sera constituée de trois neurones et la couche de sortie sera constituée de N neurones représentant le nombre de classes.

La procédure de segmentation proposée est inspirée d'une version modifiée du FCM [21], qui consiste à minimiser une fonction objective donnée par :

$$\min_{U,V} \{ \sum_{i=1}^n \sum_{j=1}^c U_{ji}^m \|X_i - V_j\|^2 + \vartheta \operatorname{tr}(VDV^T) \} \quad (\text{II.15})$$

où :

$\vartheta > 0$ représente un paramètre assurant une distribution spatiale régulière des vecteurs de codes distribués sur une grille. Ce paramètre force la valeur d'un vecteur de codes à être très proche de la valeur moyenne de ses plus proches voisins sur la grille cf Fig. II.5

D : L'opérateur Laplacien.

V : Matrice constituée par les vecteurs de codes.

tr : Trace de la matrice.

| | | |
|----------------------|----------------------|----------------------|
| V₁ | V₂ | V₃ |
| V₄ | V₅ | V₆ |
| V₇ | V₈ | V₉ |

Fig. II.6. : Une grille 3x3 contient $c=9$ vecteurs de codes.

Dans ce cas les centres de classes sont calculés par :

$$V_j = \frac{\sum_{i=1}^n U_{ji}^m X_i + \vartheta \bar{V}_j}{\sum_{i=1}^n U_{ij}^m + \vartheta} \quad (\text{II.16})$$

Où \bar{V}_j représente la moyenne des vecteurs de codes les plus proches voisins à V_j dans la grille. Dans ce cas, en faisant référence à la grille de la Fig. II.6 :

$$\begin{aligned}\bar{V}_1 &= (V_1 + V_4) / 2 \\ \bar{V}_2 &= (V_1 + V_3 + V_5) / 3 \\ \bar{V}_5 &= (V_2 + V_4 + V_6 + V_8) / 4\end{aligned}\tag{II.17}$$

La mise à jour des vecteurs de codes reflète la nature du SOM qui sera beaucoup influencé par les valeurs des données et le plus proche code vecteur dans la grille.

L'algorithme suivant trouve la solution qui converge au minimum local de la fonction dans l'équation (II.15)

1. Initialiser aléatoirement V et U satisfaisant les contraintes :

$$\begin{aligned}0 &\leq U_{ji} \leq 1 \\ \text{Et} \\ \sum_{j=1}^c U_{ji} &= 1, \quad \forall i\end{aligned}\tag{II.18}$$

2. mettre $\vartheta > 0$.

3. mettre $m > 1$

4. Pour $i=1 \dots n$, et $j=1 \dots c$, calculer

$$U_{ji} = \frac{1}{\sum_{k=1}^c \frac{\|X_i - V_j\|^{2/(m-1)}}{\|X_i - V_k\|^{2/(m-1)}}}\tag{II.19}$$

5. Pour $j=1 \dots c$, calculer

$$V_j = \frac{\sum_{i=1}^n U_{ji}^m X_i + \vartheta \bar{V}_j}{\sum_{i=1}^n U_{ji}^m + \vartheta}\tag{II.20}$$

6. arrêter quand la différence entre la valeur courante et la valeur précédente de U_{ji} est inférieure à ϵ ; sinon aller à 4.

Les algorithmes basés sur le principe du clustering sont très sensibles aux conditions initiales. Donc, le minimum local dépend des choix initiaux des valeurs des centres de classes. Pour atteindre le minimum global, une stratégie développée dans [27] est utilisée.

Elle consiste à itérer les étapes 5 et 6 de l'algorithme pour une décroissance linéaire du paramètre flou m entre deux valeurs m_2 et m_1 .

II.15. Conclusion

Dans ce chapitre, nous avons mis l'accent sur les différentes approches utilisées pour la classification supervisée et particulièrement non supervisée, ce qui nous a conduits à introduire les approches floues sous leurs différentes formes. Dans le prochain chapitre, nous aborderons la conception de nos algorithmes pour mettre en place les outils permettant de les évaluer.

Chapitre III
Conception et Réalisation.

III.1. Introduction

Après avoir présenté les différentes techniques de classification floue, nous exposerons dans ce chapitre, d'abord, l'ensemble des algorithmes et leurs implémentations sous l'environnement MATLAB en mettant l'accent sur le choix de ce dernier. Afin de mieux évaluer les méthodes programmées et valider les résultats obtenus, nous avons choisi deux types d'images. Une image IRM et une fenêtre multispectrale du capteur SX de SPOT sur la région de Tizi-Ouzou.

III.2. L'environnement de travail

MATLAB « *matrix laboratory* » est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004[34]) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Matlab peut s'utiliser seul ou bien avec des *toolbox* « boîte à outils », ce qui le rend facilement maniable par rapport aux autres langages de programmation.

III.2.1. Histoire

Le langage MATLAB a été conçu par Cleve Moler à la fin des années 1970 à partir des bibliothèques Fortran, LINPACK et EISPACK[35]. Alors professeur de mathématiques à l'université du Nouveau-Mexique, il souhaitait permettre à ses étudiants de pouvoir utiliser ces deux bibliothèques sans connaître le Fortran. Cleve Moler l'utilisa ensuite pour des cours donnés à l'université Stanford où il reçut un accueil mitigé de la part des étudiants en mathématiques (habitués au Fortran). Par contre, les étudiants en technologie, en particulier en traitement du signal, furent beaucoup plus intéressés. Un ingénieur, Jack Little en comprend rapidement les capacités et entreprend avec un collègue, Steve Bangert, de le recoder en langage C. Jack Little, Cleve Moler et Steve Bangert créèrent la société The

MathWorks en 1984 afin de commercialiser la version 1.0 de MATLAB[35],[36].

MATLAB a ensuite évolué, en intégrant par exemple la bibliothèque LAPACK en 2000[37], en se dotant de nombreuses boîtes à outils (Toolbox) et en incluant les possibilités données par d'autres langages de programmation comme C++ ou Java.

III.2.2. Syntaxe

Le logiciel MATLAB est construit autour du langage MATLAB. Une interface en ligne de commande, qui est un des éléments du bureau MATLAB, permet d'exécuter des commandes simples. Des séquences de commandes peuvent être sauvegardés dans un fichier texte, typiquement avec l'éditeur MATLAB, sous la forme d'un « script » ou encapsulé dans une fonction [38].

III.2.3. Variables

Les variables sont définies avec l'opérateur d'affectation =. MATLAB est un langage à typage faible (dans le sens où les conversions de type sont implicites)[39]. Les variables ne sont pas déclarées avec leur type, sauf pour les objets symboliques [40].

III.2.4. Vecteurs et matrices

Comme son nom l'indique, l'intérêt de MATLAB est de pouvoir manipuler facilement des tableaux à une dimension (« vecteur » dans le vocabulaire MATLAB) ou deux dimensions (« matrices image ») ou plus. Par défaut, toutes les variables sont des tableaux, MATLAB permet de faire de la programmation matricielle, ce qui rend la manipulation des images plus aisée et ce, en se servant des multiples outils.

La plupart des fonctions MATLAB acceptent les matrices et s'appliquent à chacun de ses éléments. Par exemple, `mod(2*J,n)` multiplie chaque élément de "J" par 2, puis divise chaque élément modulo "n". Par ailleurs s'il est possible d'utiliser des boucles « For » ou « while », il est possible de se servir des notations vectorielles pour les remplacer. Par exemple, le code suivant, crée un carré magique de dimension n pour, n impair [42].

```
[J,I] = meshgrid(1:n);  
A = mod(I+J-(n+3)/2,n);  
B = mod(I+2*J-2,n);  
M = n*A + B + 1;
```

De même, la fonction **meshgrid** permet de créer des matrices carrées contenant 1:n.

III.2.5. Outils et modules associés

MATLAB est complété par de multiples boîtes à outils.

Parmi les plus importantes, nous retrouvons :

- Communications Toolbox
- Control System Toolbox
- Excel Link
- MATLAB Compiler
- Neural Network Toolbox
- Optimization Toolbox
- Parallel Computing toolbox
- Real-Time Workshop®, renommé commercialement SimulinkCoder[43]
- Robust Control Toolbox
- SimMechanics
- SimPowerSystems
- Simulink
- Statistics and Machine Learning Toolbox
- System Identification Toolbox
- Virtual Reality Toolbox

III.2.6. Alternatives

- AMESim
- FreeMat
- GAUSS
- IDL
- JMathLib
- Maple

- Mathcad
- Mathematica
- ocamlmath [44]
- Octave
- O-Matrix (en)
- OxMetrics (en)
- PyIMSL Studio
- Sage
- SAS/IML
- Scilab
- SciPy
- Sysquake

III.3. Présentation des algorithmes

Pour mettre en place nos algorithmes, nous avons utilisé une structure de données proche de MATLAB et ce, pour pouvoir exploité aisément les fonctionnalités de ce dernier.

III.3.1. Algorithme k-means

Entrée :

- 1- Créer une matrice de lecture à r lignes et c colonnes : **img** [r][c].
- 2- Créer une matrice de sortie (image segmentée) : **b4** [r][c]
- 3- Créer une matrice de partition qui comporte le degré de partition de chaque pixel : **d** [$r*c$][nc] ; nc représente le nombre de classes.
- 4- Créer trois vecteurs de sauvegarde **v** [1][$r*c$] , **x** [1][$r*c$], **x1** [nc]

Paramètres d'initialisation de l'algorithme

- Lire nc #Nombre de classes#

Début :

- Lire l'image à traiter dans la matrice **img** ;

- **w**=1 ;

pour t=1 à c **#Remplissage de r #**

$x(r \cdot (t-1) + 1 : r \cdot t) = \text{img}(w, t); w = w + 1 ;$

fin pour

m1=0 ;

pour j=1 à nc

pour i=1 à r

$x1(j) = x1(j) + u(i, j) ;$

fin pour

$m1 = m1 + x1(j) ;$

fin pour

$m1 = m1 / nc ;$

pour i=1 à r

pour j=1 à c

$\mu(i, j) = \mu(i, j) - m1 ;$

fin pour

fin pour

norme (μ , r*c, nc, N) ;

#Calcul des centres de classes #

nil=0; w=1 ;

```

tant que N>0.0001

  pour i=1 à nc
  |
  |   v(i)=x*(  $\mu$  (w,i).^m)./sum( $\mu$ (w,i).^m);w=w+1;
  |
  fin pour

  #Mise à jour de la matrice de partition U #
  #Calcul des distances euclidiennes#
  w=1 ;

  Pour i=1 à nc
  |
  |   d(w,i)=abs(x'-v(i)); w=w+1 ;
  |
  fin pour

  #calcul de la matrice de partition u#
  #Sauvegarde de la matrice u dans s#
  pour il=1 à r
  |
  |   pour j1=1 à c
  |   |
  |   |   s(il, j1)=  $\mu$  (il ,j1);
  |   |
  |   fin pour
  |
  fin pour

   $\mu$  =zéro( $\mu$ ,r*c,nc) ;

  pour i=1 à r*c
  |
  |   si d(i,1)<d(i,2);
  |   |
  |   |    $\mu$  (i,1)=1;
  |   |
  |   sinon
  |   |
  |   |    $\mu$  (i,2)=1;
  |   |
  |   finsi;
  |
  fin pour;

  N=norm(s- $\mu$ ) ; nil=0;

```

```

pour i=1 à r
|
|   pour j=1 à c
|   |
|   |   s(i,j)= s(i,j) -  $\mu$  (i,j) ;
|   |
|   fin pour
|
fin pour

```

norme (s, r*c, nc, N)

nil=nil+1 ;

fin tant que

#Construction de la matrice de l'image segmentée#

b4=zéro(b4,r,c) ;

```

pour i=1 à r
|
|   pour j=1 à c
|   |
|   |   pour k=1:nc
|   |   |
|   |   |   si  $\mu$  ((j-1)*r+i,k)==1
|   |   |   |
|   |   |   |   b4(i,j)=k ;
|   |   |   |
|   |   |   fin si
|   |   fin pour
|   fin pour
|
fin pour

```

fin pour

```

pour i1=1 à r1
|
|   pour j1=1 à c1
|   |
|   |   b4 (i1,j1)=2 - b4(i1,j1) ;
|   |
|   fin pour
|
fin pour

```

Charger le contenu de b4 sur le pixel de l'image qui est en écriture.

Fin

#Fonction de remise à zéro d'une matrice **u**#

Fonction **zero** ($\mu 1$, r1,c1)

```
Pour i1=1 à r1
|
|   Pour j1=1 à c1
|   |
|   |    $\mu 1(i1, j1)=0$  ;
|   |
|   fin pour
fin pour
```

#Fonction abs qui calcule la valeur absolue #

Fonction **abs** (g)

Si $g < 0$ alors $g = -g$;

Return(g) ;

#Fonction randn qui calcule les valeurs d'une matrice aléatoire #

Fonction **randn** ($\mu 1$,r1 ,c1)

```
Pour i1=1 à r1
|
|   Pour j1 = 1 à c
|   |
|   |    $\mu(i1, j1) = (i1*j1/(r1*c1))$  ;
|   |
|   fin pour
fin
```

#Fonction mean qui calcule les moyennes d'une matrice (colonne ou ligne)

Fonction mean($\mu 1$)

```
Pour il=1 à r1
|
|   Pour jl=1 à c1
|   |    $\mu = \mu - \text{mean2}(\mu)$  ;
|   |   fin pour
|   fin pour
fin pour
```

Calcul de la norme de la matrice $\mu 1$ #Procédure norme ($\mu 1$, $r1*c1$, $nc1$, $w1$) $w1 = 0$;

```
Pour il=1 à r1
|
|   s1=0 ;
|
|   Pour jl=1 à c1
|   |    $s1 = s1 + \text{abs}(\mu 1(i,j))$  ;
|   |   fin pour
|
|   si  $w1 < s1$  alors  $w1 = s1$ 
|   |   fin si
|   fin pour
fin pour
```


III.3.2 Algorithme FCM

Entrée :

- 1- Créer une matrice de lecture à NL lignes et NC colonnes : **img**[NL][NC].
- 2- Créer une matrice double (réel) de NL lignes et NC colonnes **imgd**[NL][NC].
- 3- Créer une matrice de sortie (image segmentée) : **imgs**[NL][NC].
- 4- Créer une matrice de partition qui comporte le degré de partition de chaque pixel : $\mu [c][n]$; n représente le nombre de pixel.
- 5- Créer un vecteurs **v** qui transfert l'image **v** [1][(NL*NC)] ;

Paramètres d'initialisation de l'algorithme

Lire n #Nombre de pixel#

Début :

- Lire l'image à traiter dans la matrice **img** ;

Pour i=1 à NL

Pour j=1 à NC

imgd[i,j] = (réel) img[i][j]

fin pour

fin pour

p=1 ;

Pour i=1 à NC

Pour j=1 à NL

v[p] = img[i][j] ; p=p+1 ;

fin pour

fin pour

```
 $\mu_1 = \mu^2$  ; j1=5 ; j2=1 ;
```

```
tant que (abs(j1-j2))>=0.001
```

```
    #calcul des centre des classes#
```

```
    Pour i=1 à c
```

```
        b(i)=sum(( $\mu_1$ (i,1:n))*v(1:n))/sum( $\mu_1$ (i,1:n)).
```

```
    fin pour
```

```
    #calcul des distance pour chaque classe et pixel #
```

```
    Pour i=1:3
```

```
        Pour j=1 à p
```

```
            d(i,:)=abs((v(:))-b(i)) ;
```

```
        fin pour
```

```
    fin pour
```

```
    d1=d^2 ;
```

```
    # calcul du critère j #
```

```
    Pour k=1 à 3
```

```
        jj=sum(sum( $\mu_1$ (k,:)*d1(k,:)')) ;
```

```
        j1=sum(jj) ;
```

```
    fin pour
```

```
    # mise a jour de uij #
```

```
    Pour i=1 à 3
```

```
        Pour j=1 à p
```

```
            Pour k=1 à 3
```

```
                 $\mu(i,:)=(((d1(i,:))/(d1(1,:))).^2)+(((d1(i,:))/(d1(2,:))).^2)+(((d1(i,:))/(d1(3,:))).^2)).^(-1);$ 
```

```
            fin pour
```

```
        fin pour
```

```
    fin pour
```

```
     $\mu_1=aa.^2$  ;
```

```

    # recalcul du critère avec le nouveau  $\mu_{ij}$  #
    pour i=1 à 3
        jj2=sum(sum( $\mu_1(i,:)*d_1(i,:)$ ')) ;
        j2=sum(jj2) ;
    fin pour
    Charger le contenu de  $\mu$  sur le pixel de l'image qui est en écriture ;
fin tant que
fin

```

#Fonction abs qui calcule la valeur absolue #

Fonction **abs** (y)

Si $g < 0$ alors $y = -y$;

Return(g)

#Fonction double qui transfère l'image en double #

Fonction **double** (μ_1, c_1, n_1)

```

Pour i1=1 à c1
    Pour j1=1 à n1
         $\mu_1(c_1, n_1) = aa(c_1, n_1)^2$  ;
    fin pour
fin pour

```

vecti(m, NI, NC, i1, z)

```

pour j1=1 à NC
    z(j1)=m[i1][j1] ;
fin pour

```

```
pour i=1 à 3
|
|   z[i]=vecti(d1,NL,NC,i,z)
|
|   pour j= 1 à NC
|   |
|   |    $\mu[i][j]=[z[j]/(d1[1][j])^2]+ [z[j]/(d1[2][j])^2]+ [z[j]/(d1[3][j])^2]$  ;
|   |
|   fin pour
|
fin pour
```

IV. Conclusion

Dans ce chapitre, nous avons fait une description du langage de programmation MATLAB qui très adapté à l'implémentation des techniques évoquées. Par la suite, nous avons présenté d'une manière exhaustive les algorithmes déjà évoqués et ce, en respectant les structures de données que MATLAB utilisent. Nous pouvons affirmer aisément que le choix de MATLAB est très adapté à notre problématique. Dans le prochain chapitre, nous tenterons de tester nos méthodes sur des images particulièrement multispectrales pour pouvoir tirer une évaluation.

Chapitre IV
Expérimentation.

IV.1. Introduction

Afin de tester et d'évaluer les techniques développées, nous présenterons dans ce chapitre, les données qui ont fait l'objet de notre travail, ainsi que les résultats obtenus.

IV.2. Les données utilisées

Afin de tester et d'évaluer les techniques développées, nous avons fait recours à deux types de données. La première image est une fenêtre mustispectrale de 256 x 256, extraite des données XS de SPOT de la région de Oued-Aissi.

Présentation des données SX de SPOT

Notre région d'étude se situe dans la partie Est de la daïra de Tizi-Ouzou. Elle couvre toute la surface allant du village Chemlel jusqu'à l'extrémité Est de la zone industrielle de Oued-Aissi. Avec une résolution au sol de 30 x 30 m, la région « test » couvre une étendue de 58 982400 m². Elle est délimitée :

- . Au Nord, par l'oued Sébaou au pied du versant sud du Djebel Aissa Mimoun ;
- . Au Sud, par la ligne de crêt ;
- . A l'Est, par la zone industrielle de Oued-Aissi ;
- . A l'Ouest par la plaine de Chamlal.

Cette région traversée par la route nationale N° 12 est constituée principalement d'un couvert végétal varié allant des champs d'agrumes jusqu'à la végétation éparse tout au long des vallées autour des deux oueds (Sébaou et Aissi). Les deux versants au sud de Oued Aissi sont recouverts d'oliviers ainsi qu'un dense peuplement de brousses et de forêts plutôt claires. La zone est surtout caractérisée par les hangars ainsi que les surfaces goudronnées de la zone Industrielle. Cette région diversifiée par des thèmes d'occupation des sols variés, présente un cas idéal pour tester les techniques développées.

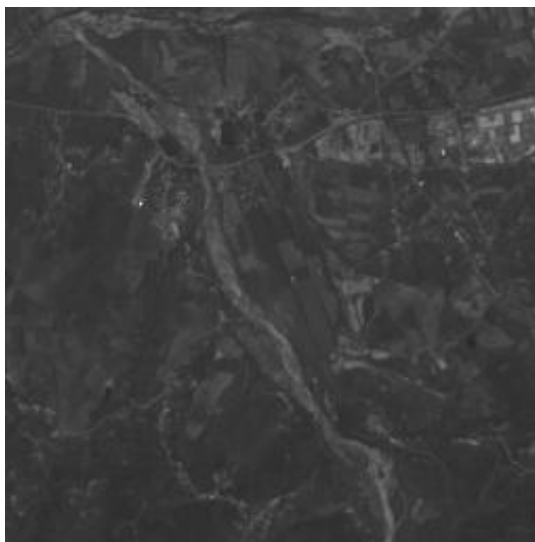


Fig. IV.1. : *Canal de la bande verte SX1.*

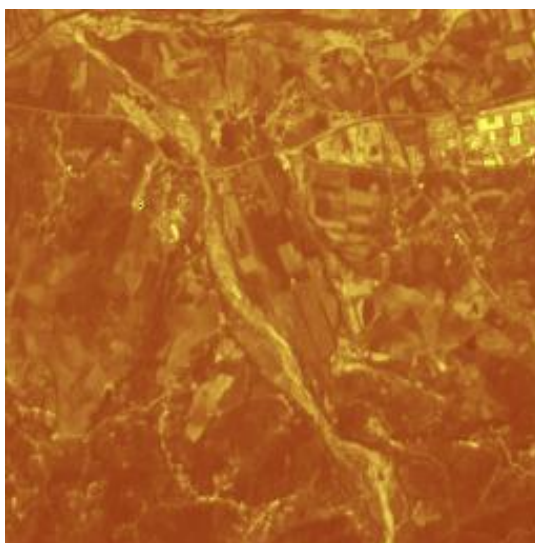


Fig. IV.2. : *Canal de la bande rouge XS2.*

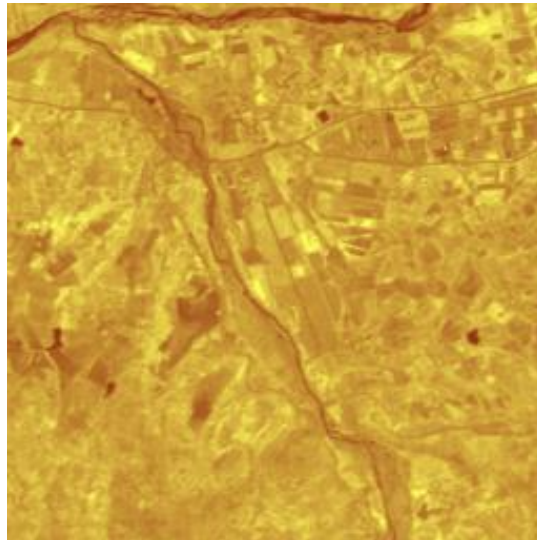


Fig. IV.3. : *Canal de la bande Infrarouge XS3.*

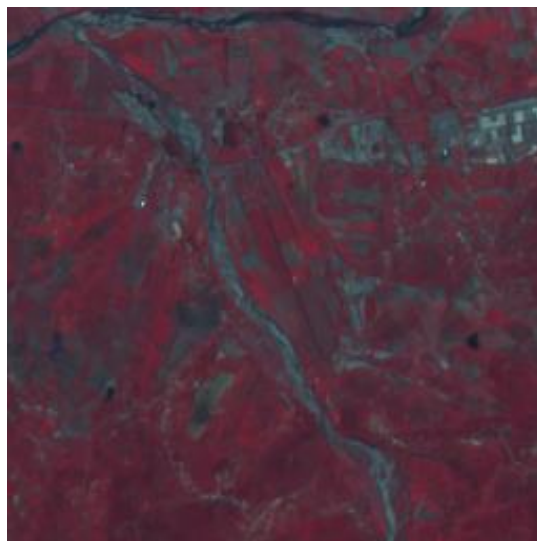


Fig. IV.4. : *Composition colorée usuelle (RVB) des 3 bandes précédentes.*

IV.3. Présentation des résultats

Afin d'évaluer notre application thématique sur des données réelles, nous avons procédé à 2 analyses non dirigées (non supervisées) allant de 3 à 4 classes et ce, pour estimer l'affectation des différents thèmes d'occupation des sols. Nous rappelons que nous avons utilisé la carte topographique de la région à l'échelle du 1 : 50 000 bien que l'évolution de l'occupation des sols est très dynamique ces dernières années, phénomène dû, particulièrement à l'urbanisation intensive et à la dégradation des différents couverts

végétaux. Cependant, **la connaissance parfaite du terrain** reste notre référence principale pour tenter de valider ou du moins, d'évaluer nos résultats.

IV.3.1. K-means

Les figures suivantes IV.5. et IV.6 représentent les résultats obtenus.

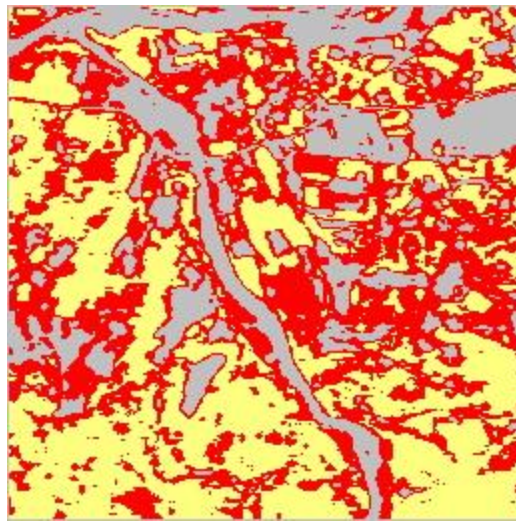


Fig. IV.5. : *K-Means avec 3 classes.*

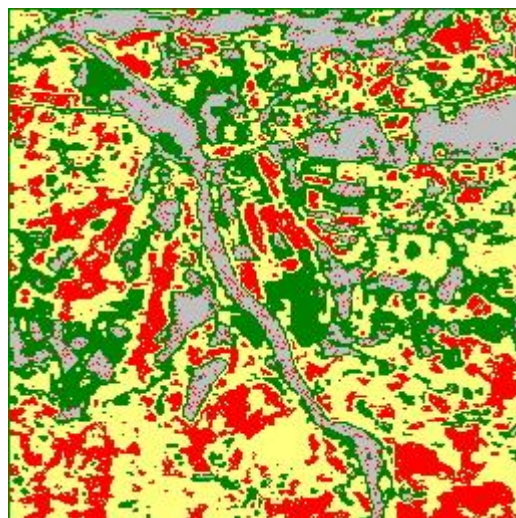


Fig. IV.6. : *K-Means avec 4 classes.*

Nous remarquons que la classification à 3 classes (Fig. IV.5.) a fourni un partage trop grossier. La végétation éparses et les agrumes sont regroupés dans une même classe, représentée par la couleur rouge. Ceci s'explique par des réponses spectrales proches, particulièrement dans la bande spectrale XS3 (ProcheInfrarouge). La couleur jaune regroupe les terrains plus ou moins nus, parsemés de brousses et de quelques arbres de faible densité. Les sables des oueds, le bâti et les terrains totalement nus sont affectés à une même classe, représentée par la couleur grise.

En passant à 4 classes (Fig. IV.6.), nous remarquons que les agrumes et d'autres arbres fruitiers (couleur verte) se démarquent du premier regroupement. Nous remarquons aussi que la végétation éparses devient plus réaliste avec une bonne répartition, mais les cultures maraichères sont sous estimées et affectées à d'autres classes.

Dans les deux classifications, l'occupation des sols à l'intérieur de la zone industrielle de Oued-Aissi est détectée d'une manière très irrégulière, chose qui ne reflète la réalité du terrain.

IV.3.2. FCM

Les résultats obtenus pour 3 et 4 classes préfixées sont représentés successivement dans les figures IV.7 et IV.8 ci-dessous.

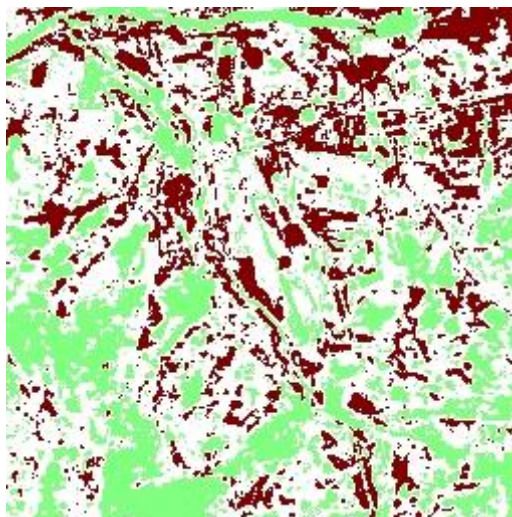


Fig.IV.7. : *FCM avec 3 classes.*

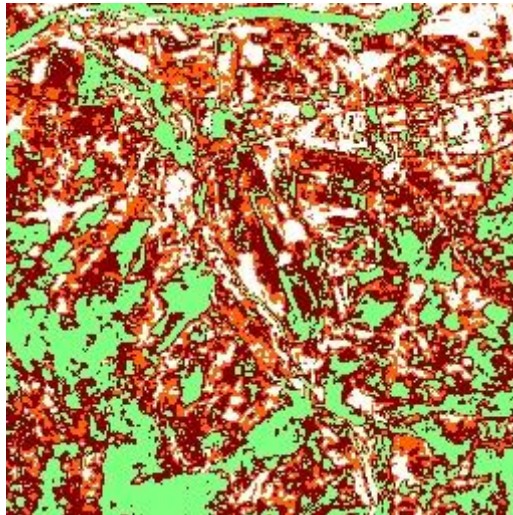


Fig.IV.8. : *FCM avec 4 classes.*

La classification à 3 classes (Fig. IV.7.) nous a fourni un découpage peu intéressant du point de vue thématique à l'exception de la classe « brousses et forêt claire », caractérisée par la couleur verte. Les autres thèmes d'occupation des sols sont regroupés seulement en deux classes et ce, selon le seul critère de ressemblance des réponses spectrales dans les 3 bandes. En passant à 4 classes (figure IV.8.), la classification devient plus réaliste. En effet, les terrains très clairs et le bâti (couleur blanche) est parfaitement détecté avec son recouvrement exact en tenant compte bien sûr des vérités terraines, chose qui a été loupé par les K-Means. De même, la classe des agrumes et des arbres de grande taille (couleur marron) sont parfaitement détectées le long des oueds, sur les plaines alluviales ainsi que le long des deux bassins versants qui déversent sur l'Oued-Aissi. De même, les petites surfaces chlorophylliennes à l'intérieur de la zone industrielle sont mises en évidence avec un taux de recouvrement qui reflète la réalité. La classe végétation éparse et brousses maigres de faible densité (couleur orange) est détectée avec une bonne répartition.

IV.4. Conclusion

La tâche de classification des données satellitaires sur l'occupation des sols est assez difficile en raison de la diversité et de la complexité du milieu. Nous avons remarqué des insuffisances dans la première méthode. En effet, dans la classification FCM, chaque point a une probabilité d'appartenance à chaque groupe, plutôt que complètement appartenant à un seul cluster comme il est le cas dans les k-Means. Ce critère d'appartenance a amélioré

considérablement le processus de classification comme on vient de le voir dans l'analyse thématique précédente. Cependant, il nous semble que la classification Neuro-floue améliorera encore les résultats à cause de l'utilisation d'autres informations liées plutôt une analyse plus intelligente. Appliquée aux images monospectrales (type IRM), cette technique a amélioré considérablement la classification non supervisée. Nous n'avons pas pu l'appliquer pour une image à plusieurs bandes spectrales et ce, à cause de la complexité de son développement.

Conclusion générale

L'objectif de ce travail a été l'étude de la contribution de la logique floue à la classification des données multispectrales XS de SPOT et ce, pour l'élaboration de documents thématiques caractérisant l'occupation des sols dans un contexte d'une région algérienne.

Dans ce mémoire qui s'articule en quatre parties, nous avons d'abord abordé dans le premier chapitre quelques notions générales sur les images numériques, ce qui nous a permis de bien comprendre l'image et les différents supports de stockage. De même, nous avons donné un aperçu sur les différents traitements préliminaires pour pouvoir extraire ou améliorer les données de base, afin d'obtenir déjà à priori, certaines informations sur la structure des différentes entités significatives formant l'image.

Dans la deuxième partie, nous avons exposé des approches de classification dites « classiques » pour mieux cerner les processus d'affectation des entités de l'image aux différentes classes et décrire brièvement, les imperfections de chacune des méthodes. Le mode supervisé qui utilise des connaissances à priori (zones d'entraînement) a été plutôt écarté et ce, pour ne s'intéresser qu'au mode automatique. Mais, rien n'empêche que nous avons exposé exhaustivement l'algorithme de Bayes et les K plus proches voisins (K-PPV) pour mieux appréhender les approches qui ont fait l'objet de notre travail. Par la suite, nous avons mis l'accent sur les approches non supervisées (*non dirigées*) qui ne nécessitent **aucun apprentissage et aucune connaissance à priori** et par là même, aucune tâche préalable d'étiquetage manuel. Elles ont pour but de découper l'espace d'individus (pixels) en zones homogènes selon un critère de ressemblance (critère de proximité de leurs vecteurs d'attributs dans l'espace de représentation entre les individus). L'algorithme K-Means qui ne fait pas appel à la logique floue est particulièrement abordé pour pouvoir faire une comparaison et une évaluation des approches floues.

Par la suite, nous avons introduit la méthode *des Fuzzy C-Means (FCM)* en mettant l'accent sur les critères d'affectation des individus de l'image aux différentes classes dénombrées à l'avance. Enfin, nous avons introduit les réseaux de neurones pour les combiner avec la logique floue et ce, pour proposer une méthode hybride appelée Neuro-Floue.

Conclusion Générale

Le troisième chapitre nous a permis, dans un premier temps de se familiariser avec la programmation sous MATLAB ensuite, nous avons développé les algorithmes en respectant les modalités de notre environnement de travail. L'approche modulaire a été adoptée et ce, pour bénéficier des richesses du logiciel MATLAB.

Dans la quatrième partie, nous avons privilégié le côté applicatif des outils développés. Après la présentation de notre zone d'étude et des données associées SX du satellite Spot, nous avons testé nos images d'abord avec la méthode des K-Means et puis, l'approche FCM. Nous avons constaté que cette dernière (FCM) a donnée des résultats plus concluants et plus représentatifs des réalités du terrain. Cette performance est due au fait au caractère de la multi affectation au préalable d'un individu à plusieurs classes.

Cependant, la classification Neuro-floue semble apporter des améliorations et ce, en utilisant les richesses des réseaux de neurones qui offrent des données exogènes aux données multispectrales. Nous avons seulement réussi à la programmer pour seulement un seul canal et ce, à cause de son caractère plutôt complexe. L'étude exhaustive de cette dernière approche restera néanmoins, la perspective essentielle de ce modeste travail.

Bibliographie

Bibliographie

- [1] O. Assas, These Doctorat en Sciences, Theme Classification Floue des Images, Faculté de Technologie, Département d'électronique, année 2012-2013, Université de Batna.
- [2] http://fr.wikipedia.org/wiki/R%C3%A9solution_spatiale_des_images_matricielles
- [3] L.A. Zadeh. Fuzzy sets. Inf.control, 8, 1965.
- [4] C. LURETTE, " Développement d'une technique neuronale auto adaptative pour la classification dynamique de données évolutives. Application à la supervision d'une presse hydraulique ", Thèse de Doctorat, Université des sciences et technologies de LILLE 2003.
- [5] J. Bezdek, L. Hall, L. Clarke, " Review of MR Image Segmentation Techniques using Pattern Recognition ", Medical Physics, 20:1033-1048, 1993.
- [6] L. KHODJA, " Contribution à la Classification Floue non Supervisée ". Thèse de Doctorat, université de Savoie, France, 1997.
- [7] J. P. COCQUEREZ, S. PHILIPP, Analyse d'images et segmentation. Masson, 1995.
- [8] R. DUDA, P. HART, " Pattern Classification and Scene Analysis " Wiley, New York, 1973. 3796-4/1997. IEEE.
- [9] F. CHAMROUKHI, " Classification supervisée : Les K-plus proches voisins ". mémoire de fin d'étude, Université du Sud Toulon – Var. 2013.
- [10] G. Dreyfus, J.M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, *Apprentissage statistique*, Eyrolles, 2008.
- [11] Simard, P.Y., Steinkraus, D., Platt, J.C., Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis. *Int. Conf. on Document Analysis and Recognition*. IEEE Computer Society, 2003.

Bibliographie

- [12] Yang, F., Paindavoine, M., Implementation of an RBF Neural Network on Embedded Systems: Real-Time Face Tracking and Identity Verification. *IEEE Trans. on Neural Networks*, vol. 14(5), pp. 1162-1175, 2003.
- [13] Egmont-Peterson, M., de Ridder D., Handels, H., Image Processing with Neural Networks - A Review. *Pattern Recognition*, **35**(10), pp. 2279-2301, 2002.
- [14] Jaeger, H., The Echo State Approach to Analysing and Training Recurrent Neural. *Technical Report* (GMD148), German National Research Center for Information Technology, 2001.
- [15] M. SEMCHEDINE, L. TOUMI, A. MOUSSAOUI, " Système Coopératif de Classification floue Possibiliste avec Rejet d'Ambigüité Application à la Segmentation d'Images IRM ", 4th International Conference on Computer Integrated Manufacturing CIP'2007.Sétif, Algérie.
- [16] Y. LIM, S.U. LEE, " On the color image segmentation algorithm based on the threshoding and the fuzzy c-means techniques". *Pattern Recognition*, 23(9), pages 935-88 952, 1990.
- [17] J. MACQUEEN, " Some methods for classification and analysis of multivariate observations ". *Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pages 281-297, 1967.
- [18] R. Krishnapuram, J. Keller, "A possibilistic Approach to Clustering", *IEEE Trans. on Fuzzy Systems*, vol. 1, n. 2, May 1993.
- [19] R. C.GONZALES, P. WINTZ, *Digital Image Processing*, Addison Wesley, 1977.
- [19] J. Vesanto, E. Alhoneimi, " Clustering of the Self Organizing Map ", *IEEE Trans. Neural Networks* 11 (3) (2000) 586-600.
- [20] E. TISSERAND, J. PAUTTEX, P. SCHWEITZER, *Analyse et traitement des signaux*,

Bibliographie

- Méthodes et applications au son et à l'image, Dunod, Paris, pp 281-285, 2008.
- [21] R. D. Pascual-Marqui et al., “ Smoothly distributed Fuzzy C-means : A New Self-Organizing Map”, Pattern recognition 34 (2001) 2395-2402.
- [22] S. Ameer, Z. Ameer, " Revue des approches de segmentation d'images textures : exemple des images météorologiques", 3rd International Conférence : Science of Electronic, Technologies of Information and télécommunication, Tunisia, 2005.
- [23] P. BORNE, J. HAGGE, Les réseaux de neurones présentation et applications, Paris, Edition Technip, 2007.
- [24] O. GUENOUNOU, " Méthodologie de conception de contrôleurs intelligents par l'approche génétique- application à un bioprocédé ",Thèse de Doctorat, Systèmes Automatiques, Université Toulouse III – Paul Sabatier, 2009.
- [25] R. JEAN-PHILLIPE, Réseaux neuronaux, Une introduction accompagnée d'un modèle Java. Vuibert, Paris, pp 56-57, 2010.
- [26] T. Kohonen, “ The Self-Organizing Maps ”, Neurocomputing 21 (1998) 1-6.
- [27] T. Graepel, M. Burger, K. Obermayer, “ Self Organizing-Maps : Generalisation and new Optimisation techniques ”, Neurocomputing 21 (1998) 173-190.
- [28] A. Nakib, " Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application à des images biomédicales ". Thèse de Doctorat. Université de paris 12-val de marne France. 2007.
- [29] N. Monmarché, " Algorithmes des fourmis artificielles: applications à la classification et à l'optimisation ". Thèse de Doctorat. Université de Tours, France. Décembre 2000.
- [30] K. Takahashi, H. Nakatani, K. Abe. " Color Image Segmentation using ISODATA clustering method ". 2nd Asian Conf. On computer Vision, vol 1, pp 523-527, 1995.

Bibliographie

- [31] J.C. Bezdek. "Pattern Recognition with fuzzy Objective Function Algorithm". New York, Plenum Press, 1981.
- [32] J.C. Bezdek, J. Hathaway, M.J. Sabin, and W. T. Tucker, "Convergence Theory for Fuzzy C-Means : Counterexamples and Repairs", IEEE Transactions, 17, pp. 873-877, September/October 1987.
- [33] Richard Goering, "Matlab edges closer to electronic design automation world ", *EE Times*, 10/04/2004.
- [34] Cleve B. Moler, "*Design of an interactive matrix calculator* ", AFIPS '80 Proceedings of the May 19-22, 1980, national computer conference.
- [35] Cleve Moler, " *The Origins of MATLAB* ".
- [36] Code source de la première version de MATLAB sur le site FTP du FUNET.
- [37] Cleve Moler, " *MATLAB Incorporates LAPACK* ".
- [38] " *MATLAB technical documentation* ", Mathworks.com.
- [39] Documentation on MATLAB in relation to other languages.
- [40] sym function Documentation for the MATLAB Symbolic Toolbox.
- [41] " *MATLAB Online Documentation* ", Mathworks.com.
- [42] Experiments with MATLAB, Cleve Moler (chap. 10 - Magic Squares).
- [43] <http://www.mathworks.com/products/simulink-coder> Simulink Coder.
- [44] page web d'ocamlmath.