



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider – BISKRA  
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie  
**Département d'informatique**

N° d'ordre : /M2/2019

## Mémoire

Présenté pour obtenir le diplôme de master académique en

# Informatique

Parcours : **Image et Vie Artificielle (IVA)**

---

# Une classification robuste d'images

---

Par

**BENDAHMANE DJAHID**

Soutenu le 7 juillet 2019, devant le jury composant de :

<b>Bougtitiche Amina</b>	<b>MAA</b>	Président
<b>Belaiache Hamza</b>	<b>MAA</b>	Examineur
<b>Chighoub Fouzia</b>	<b>MAA</b>	Encadreur

# DÉDICACES

Je dédie ce modeste travail :

A toute ma famille

A tous mes amis

A tous ceux qui m'ont aidé

# Remerciements

*Je tiens à exprimer toute ma reconnaissance à ma directrice de mémoire, Madame Chighoub Fouzia. Je la remercie de m'avoir encadré, orienté, aidé et conseillé.*

*J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.*

*Je remercie mes très chers parents, Brahim et Houria, qui ont toujours été là pour moi. Je remercie mes sœurs pour tous, et mon frère Hafed et abd el Madjid, pour leurs encouragements.*

*Enfin, je remercie mes amis Illyés Allali, Brahim Serray, Massoud Lallali et Abd el Hakim Berreghise et grand remercie pour Saddam Hocine Mansouri qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.*

*À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude*

## Table des matières

Chapitre I.	La segmentation d'images.....	2
I.1.	Introduction.....	2
I.2.	La définition de la segmentation.....	2
I.3.	Les domaines d'application de la segmentation.....	3
I.3.1.	Segmentation de visage.....	3
I.3.2.	Imagerie médicale IRM.....	4
I.3.3.	Segmentation du mouvement.....	4
I.3.4.	Reconnaissance d'objets.....	5
I.4.	Les approches de segmentation.....	5
I.4.1.	Approche contour.....	6
I.4.2.	Approches régions.....	8
I.4.3.	Segmentation par les approches hybrides (coopérative).....	14
I.5.	Conclusion.....	16
Chapitre II.	La classification spatiale d'images.....	17
II.1.	Introduction.....	17
II.2.	La classification d'image.....	17
II.3.	Éléments de la classification.....	18
II.4.	Application de la classification.....	18
II.5.	Les méthodes de classification.....	19
II.5.1.	Méthode supervisée.....	19
II.5.2.	Méthode non supervisée.....	19
II.6.	La classification spatiale d'image.....	23
II.6.1.	La génération d'une nouvelle image d'entrée.....	23
II.6.2.	La Modification de fonction objective.....	25
II.7.	Conclusion.....	26
Chapitre III.	La conception.....	27
III.1.	Introduction.....	27
III.2.	Conception générale.....	27
III.2.1.	Image d'entrée.....	28
III.2.2.	Les méthodes de classification.....	28
III.2.3.	L'étiquetage.....	29
III.2.4.	Les méthodes de comparaison.....	29
III.3.	La conception détaillée.....	30
III.3.1.	Sauvegarde de l'image d'entrée.....	32

## Table des matières

III.3.2.	La génération d'une nouvelle image .....	32
III.3.3.	L'initialisation des paramètres d'entrées .....	36
III.3.4.	Calcul des centres et des degrés d'appartenance.....	36
III.3.5.	Le critère d'arrêt.....	37
III.3.6.	L'étiquetage.....	37
III.4.	Les algorithmes utilisés .....	37
III.5.	Conclusion.....	41
Chapitre IV.	L'implémentation.....	42
IV.1.	Introduction .....	42
IV.2.	L'environnement de travail .....	42
IV.3.	Le langage de codage .....	42
IV.4.	Logiciel de codage.....	43
IV.5.	Bibliothèques utiliser.....	43
IV.6.	Caractéristiques de la machine .....	44
IV.7.	Présentation de l'interface de l'application .....	44
IV.8.	La comparaison entre les algorithmes .....	49
IV.9.	Conclusion.....	52

## Table des figures

Figure I-1 : Exemple de segmentation d'image .....	3
Figure I-2 : Exemple de segmentation de visages .....	4
Figure I-3: Exemple de imagerie médicale IRM .....	4
Figure I-4: Exemple de segmentation de mouvement .....	5
Figure I-5: Exemple de Reconnaissance d'objets .....	5
Figure I-6: Quelques modèles de contours .....	6
Figure I-7: Modèles de contours .....	6
Figure I-8: Evolution du contour actif.....	7
Figure I-9: Principe de division de région. (a) Blocs de division obtenus , (b) Diagramme hiérarchique de la décomposition, (c) Décomposition de l'image en blocs.....	9
Figure I-10: Hiérarchie de la méthode de segmentation par fusion. ....	9
Figure I-11: Croissance progressive des régions.....	10
Figure I-12 : Segmentation d'image par coopération .....	14
Figure I-13 : Principe de la coopération séquentielle.....	15
Figure I-14 : Principe de la coopération des résultats. ....	15
Figure I-15 : Principe de la coopération mutuelle.....	16
Figure III-1: Architecture Globale du système.....	28
Figure III-2 : L'architecture détaillée du système.....	31
<b>Figure III-3</b> : (a) le point central, (b) les voisins du point central.....	33
Figure III-4 : les cas possibles des positions de pixel central (rouge) et ses voisins (vert) dans une fenêtre de taille 3*3. ....	34
<b>Figure IV-1</b> : l'interface de l'environnement .....	43
Figure IV-2: Fenêtre d'accueil.....	44
Figure IV-3 : Résultat de classification des différents algorithmes.....	45
Figure IV-4 : Résultat de classification des différents algorithmes.....	46
Figure IV-5 : Résultat de classification des différents algorithmes.....	47
Figure IV-6: Résultat de classification des différents algorithmes.....	48
Figure IV-7 : Résultat de classification des différents algorithmes.....	49

## Introduction générale

Au cours de la dernière décennie, le traitement de l'image a été considérablement amélioré et de nombreux travaux ont été réalisés dans divers domaines tels que le domaine médical, la télédétection et la reconnaissance des personnes par le visage et le mouvement, etc.

Dans un système de traitement d'images, la segmentation d'images est l'opération la plus importante car elle conditionne la qualité de l'interprétation d'une image. Un bon résultat de segmentation ne permet pas forcément une bonne interprétation, mais nous ne pouvons pas obtenir une bonne interprétation à partir d'un mauvais résultat de segmentation.

L'algorithme FCM est une méthode de classification floue basé sur la théorie des ensembles flous, il est considérée comme un outil très efficace pour la segmentation d'images, le FCM assigne, non pas à un pixel une étiquette relative à une classe unique, mais son degré d'appartenance à chacune des classes.

Le FCM présente plusieurs inconvénients, d'une part les résultats de classification du FCM lié fortement par l'étape d'initiation (la spécification de nombre de classes, l'initialisation des centres des classes), et ça rend l'algorithme converge vers des solutions d'optimum locaux, et pendant le processus de classification flou le FCM n'utilise aucune information des voisins (l'information spatiale) celle-ci s'influence largement sur la qualité de classification dans le cas de la présence de l'effet du bruit sur l'image d'une autre part.

L'objectif de ce travail est de proposer une méthode robuste de classification floue dans laquelle on intègre l'information spatiale par la génération d'une nouvelle image à partir de l'image d'entrée. Pour cela nous avons organisé ce mémoire comme suit

Dans le premier chapitre, nous parlerons de la segmentation, ses méthodes (contour, région et hybride) et domaines d'utilisation les plus importants.

Dans le deuxième chapitre, nous allons l'allouer pour définir le concept de classification, ses méthodes et en particulier la classification spatiale.

## Introduction générale

Le troisième chapitre est la conception de notre travail, dans lequel nous allons parler sur les composants de notre système et les relations entre eux, et les algorithmes utilisés.

Le dernier chapitre, tout ce qui est concret dans l'application est d'expliquer l'interface avec quelques résultats obtenus.



# **Chapitre I**

## **La segmentation d'image**

## Chapitre I. La segmentation d'images

### I.1. Introduction

L'homme sait naturellement séparer des objets dans une image. Pour séparer un objet d'un fond, l'humain se base sur des connaissances de haut niveau qui lui permettent de détecter dans l'image ce qui l'intéresse.

Fondamentalement, la segmentation est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur. L'union de ces régions doit redonner l'image initiale. La segmentation est une étape importante pour l'extraction des informations qualitatives de l'image. Elle fournit une description de haut niveau.

### I.2. La définition de la segmentation

La segmentation est la partition d'une image en un ensemble de régions qui ne se chevauchent pas et dont l'union est l'image entière (Figure I-1). Quelques règles à suivre pour obtenir une segmentation sont

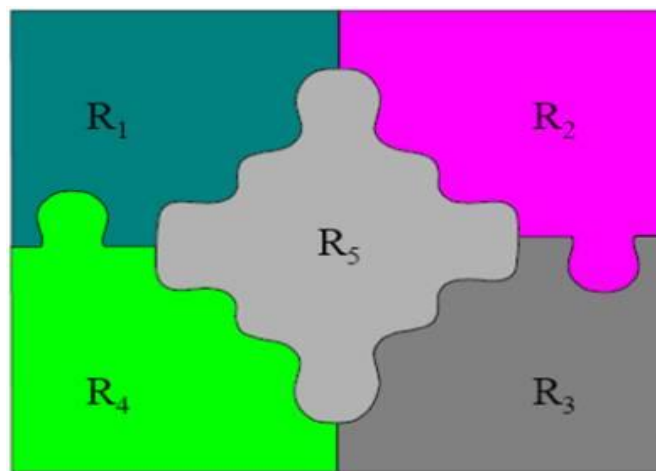
1. Les régions doivent être uniformes et homogènes par rapport à certaines caractéristiques (niveau de gris, écart type, gradient).
2. Leurs intérieurs doivent être simple et sans beaucoup de petits trous (des parties de région non segmentés).
3. Les régions adjacentes doivent avoir des valeurs très différentes par rapport à la caractéristique prise en compte dans la segmentation.
4. Les limites de chaque région doivent être simples et spatialement précises. [1]

Classiquement, on peut définir la segmentation comme étant une partition de l'image  $I$  en un nombre  $N$  de sous-ensembles  $R$  appelés régions est un ensemble connexe de points image (pixels) ayant de même propriétés (texture, intensité,...) qui les différencient des pixels des régions voisines (Figure I-1). La région  $i$  est notée, homogènes dans le sens d'un ou plusieurs critères, comme le niveau de gris, tels que

- La segmentation doit être complète (c'est-à-dire, chaque pixel doit être affecté à une classe).
- Les pixels appartenant à la même région doivent être connectés.
- Les régions doivent être disjointes. [2]

En termes mathématiques

- $U_{i=1}^n R_i = I$  où  $n$  est le nombre de régions et  $I$  est l'image complète.
- $R_i \neq \emptyset \forall i = 1..n$
- $R_i \cap R_j = \emptyset \forall i, j$  avec  $i \neq j$



**Figure I-1** : Exemple de segmentation d'image

### I.3. Les domaines d'application de la segmentation

Il existe de nombreux domaines qui utilisent la segmentation dans leurs études ou leurs recherches ou dans toute utilisation et en termes de segmentation d'image, nous en mentionnons certains.

#### I.3.1. Segmentation de visage

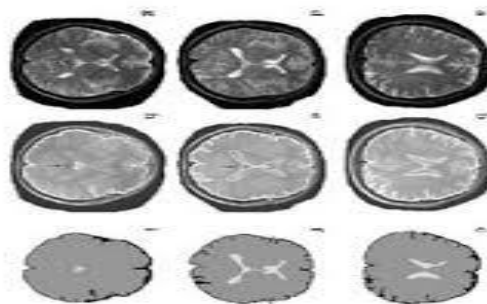
La segmentation de l'image du visage joue un rôle efficace dans l'identification des personnes et de leurs identités. Elle aide également la police à identifier les suspects et même dans les groupes d'identité électroniques (Figure I-2).



**Figure I-2 :** Exemple de segmentation de visages

### I.3.2. Imagerie médicale IRM

L'étape de la segmentation est une étape importante dans le traitement et l'interprétation des images médicales, et dans l'obtention de ces images par IRM. Le but est d'isoler les maladies potentielles (Figure I-3).



**Figure I-3:** Exemple de imagerie médicale IRM

### I.3.3. Segmentation du mouvement

La segmentation du mouvement des objets d'une scène a pour but de détecter les régions d'une image ayant un mouvement homogène et indépendant. Ses applications sont multiples, avec en premier plan la reconnaissance d'objets en mouvement. (Figure I-4)



**Figure I-4:** Exemple de segmentation de mouvement

#### I.3.4. Reconnaissance d'objets

La reconnaissance d'objets est une tâche majeure dans la vision par ordinateur. Un système de vision doit être capable de reconnaître certains objets dans la scène.

La Figure I-5 présente un exemple d'une application de reconnaissance d'objets (Application du LIRIS 2005 Guillaume Ryder [4]).



**Figure I-5:** Exemple de Reconnaissance d'objets

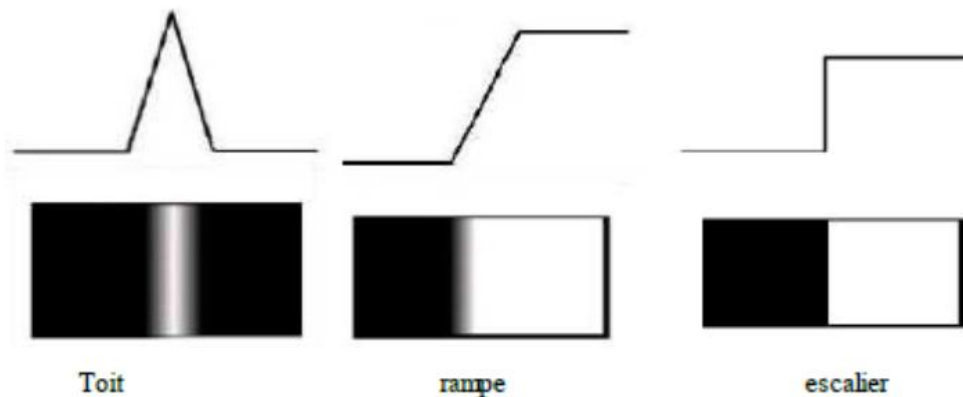
#### I.4. Les approches de segmentation

La segmentation est un vaste sujet d'étude et fait partie des grands thèmes de l'imagerie numérique. Comment préférer l'une ou l'autre est discussion ouverte et continue dans de nombreux laboratoires. Les approches de segmentation peuvent se diviser en trois grandes classes,

- Approche contour (détection de discontinuité ou encore frontières),
- Approche région (détection de similarité),
- Approche de segmentation hybride (coopérative).

### I.4.1. Approche contour

L'approche contour consiste à identifier les changements entre les régions. En général, un élément de contours est un point de l'image appartenant à la frontière de deux ou plusieurs objets ayant des niveaux de gris différents [2]. Les variations d'intensité de lumière et de couleurs sont très bien perçues par le système visuel humain. La Figure I-6 montre quelques modèles de contours.



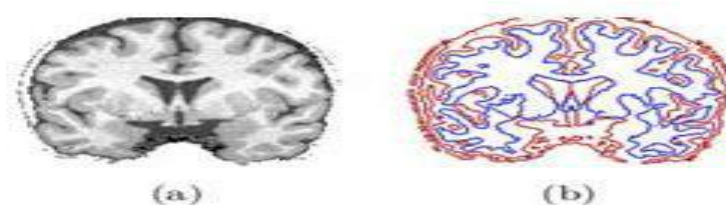
**Figure I-6:** Quelques modèles de contours

Marche d'escalier : le contour est net (contour idéal).

Rampe : le contour est plus flou.

Toit : il s'agit d'une ligne sur un fond uniforme.

La segmentation contours vise à délimiter les objets selon leurs contours. Ces approches ne se basent généralement pas sur les intensités mais sur les variations d'intensité significatives dans l'image, aux frontières entre régions (Figure I-7). [3]



**Figure I-7:** Modèles de contours

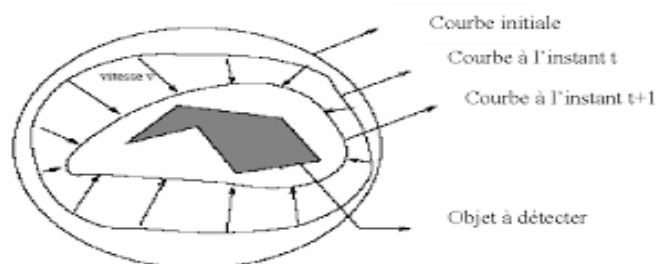
#### I.4.1.1. Contour Actif

Les contours actifs (snakes) sont définis par une courbe paramétrique pouvant être fermée ou non. Un snake consiste à placer aux alentours de la forme à détecter une ligne initiale de contour. Cette ligne va se déformer progressivement selon l'action de plusieurs forces qui vont la tirer ou la pousser vers la forme. Ces forces sont représentées par trois énergies associées au snake [10].

- Une énergie propre : due uniquement à la forme du contour, dite interne.
- Une énergie potentielle : imposée par l'image dite énergie externe. C'est elle qui va attirer la ligne du snake vers les contours réels présents sur l'image
- Une énergie de contexte : qui exprime certaines contraintes supplémentaires qui peuvent être imposées par l'utilisateur vu le snake qu'il veut obtenir.

Donc, le contour actif est formé d'une série de points mobiles et répartis sur une courbe en deux dimensions. La courbe est placée dans la zone d'intérêt de l'image ou autour d'un objet (Figure I-8). Plusieurs équations décrivent son évolution : la courbe se déplace et épouse lentement les contours des objets en fonction de divers paramètres comme l'élasticité, la tolérance au bruit,... etc.

Cette dynamique est basée sur la notion d'énergies interne et externe, le but étant de minimiser l'énergie totale présente le long de la courbe. Des contraintes permettent de conserver une courbe lisse avec des points équidistants tout en laissant un certain champ libre pour les déformations [12].



**Figure I-8:** Evolution du contour actif

Un contour actif (ideal) est défini comme une variation de luminance au point de coordonnées  $(x_0, y_0)$  séparant deux régions de luminance homogènes. On peut définir trois propriétés essentielles permettant de déterminer les caractéristiques d'un détecteur [5]

- **Une bonne détection** Une forte probabilité de détecter le point P quel que soit le bruit dans l'image.

- **Une bonne localisation** très grande précision dans la localisation de la position du point de contour. Les variations  $A_{x0}$  et  $A_{y0}$  représentant l'erreur de position doivent être minimisés.

- **L'unicité de la réponse** la Faculté a détecté un seul point de contour. Éliminer toute possibilité d'avoir une réponse multiple pour un point de contour

#### I.4.2. **Approches régions**

La segmentation d'image par l'approche région consiste à découper l'image en régions. Les pixels adjacents sont regroupés en régions distinctes selon un critère d'homogénéité ou de similarité donnée. Ce critère peut être, par exemple, le niveau de gris, couleur, texture...etc.

Un processus de groupement est répété jusqu'à ce que tous les pixels dans l'image soient inclus dans des régions. Cette approche vise, donc, à segmenter l'image en se basant sur des propriétés intrinsèques des régions. [6]

Dans ceux-ci il y a plusieurs méthodes comme croissance de région, division de région, fusion de région, classification que nous présentons ci-dessous

##### I.4.2.1. **Méthode par division (Split)**

Ce sont des méthodes qui opèrent en découpant l'image principale en régions plus petites suivant un critère d'hétérogénéité. Le découpage s'arrête quand il n'y a plus de régions inhomogènes. La méthode Quadtree est souvent utilisée pour réaliser ce découpage (Figure I-9). Il s'agit d'une décomposition récursive en quatre sous régions carrées des régions hétérogènes en commençant par l'image entière qui est de taille  $N \times N$ , où  $N = n^2$ . [2]



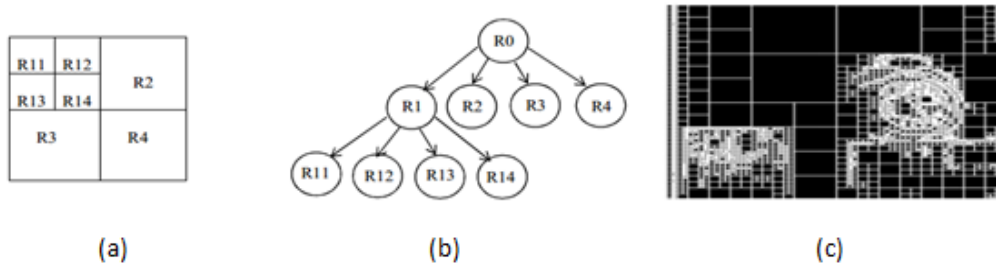


Figure I-9: **Principe de division de région. (a) Blocs de division obtenus , (b) Diagramme hiérarchique de la décomposition, (c) Décomposition de l'image en blocs.**

#### I.4.2.2. Méthode par fusion (Merge)

Ces méthodes opèrent en regroupant des pixels ou des groupes de pixels selon un critère d'homogénéité pour obtenir un ensemble de régions homogènes. Elles suivent une hiérarchie arborescente comme montrée dans la (Figure I-10). Leurs inconvénients majeurs c'est que le résultat final (image segmentée) dépend fortement du choix des régions de départ. [2]

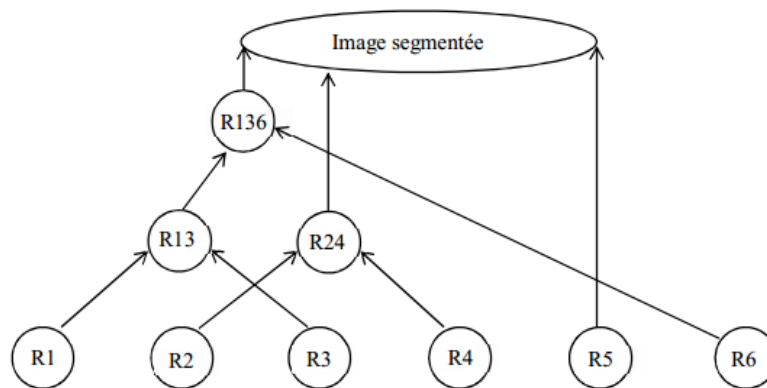
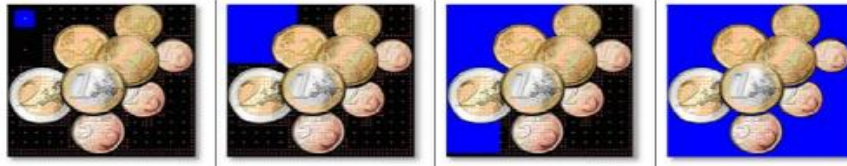


Figure I-10: **Hiérarchie de la méthode de segmentation par fusion.**

#### I.4.2.3. Méthode par division /fusion (Split and Merge)

Proposées par Horowitz [2], elles réunissent l'ensemble des algorithmes utilisés dans les techniques précédemment citées (méthodes par fusion et méthodes par division). Le processus de segmentation se déroule en deux étapes : la division en blocs (régions) suivant un critère d'hétérogénéité est effectuée en premier lieu ; puis, suit l'étape de fusion où toutes les régions adjacentes qui découlent de l'étape de division et vérifiant une heuristique d'homogénéité sont regroupées.



**Figure I-11:** Croissance progressive des régions.

#### I.4.2.4. **La classification**

Pour cette famille de méthodes, la segmentation d'une image est vue comme une subdivision de cette dernière en différentes classes de telle manière que : les éléments d'une même classe soient les plus similaires possibles (minimisation de la variance intra-classe), et les éléments de deux classes distinctes soient les plus différents possibles (maximisation de la variance interclasses). Nous pouvons grouper les méthodes classifications en deux grandes familles, en considérant l'intervention ou non d'un " attribut classe " au fur et à mesure du processus de la classification, ces deux types sont "supervisée" et "non supervisée". [8]

##### **A. Méthodes supervisées**

Les méthodes de classification supervisée nécessitent une base d'images déjà segmentées manuellement. Cette base permet de faire un apprentissage du classifieur. Une fois les paramètres du classifieur déterminés, de nouvelles images peuvent être segmentées. L'exemple le plus simple de ces méthodes est le classifieur du plus proche voisin où chaque pixel est affecté à la même classe que celle du pixel présent dans la base d'apprentissage ayant le plus proche niveau de gris [9]. Dans cette catégorie on peut distinguer les réseaux de neurones et l'algorithme de plus proche voisin.

- **Les réseaux de neurones**

Les réseaux de neurones ont été appliqués avec succès dans plusieurs domaines [25]. Une caractéristique des réseaux de neurones est qu'ils nécessitent de longs temps d'entraînement, mais ils sont des classifieurs relativement rapides [26]. Cependant, la raison principale qui motive l'utilisation des méthodes neuronales pour la classification des données de télédétection est que ces méthodes sont non paramétriques. L'approche neuronale évite aussi le problème qui consiste à spécifier l'influence que chaque source de donnée aura dans la classification, contrairement à l'analyse statistique de donnée multi-source [26].

En conséquence l'utilisation des réseaux de neurones comme outil de traitement des données de télédétection est principalement motivé par les raisons suivantes :

- Ils ont la capacité d'opérer avec une plus grande précision que les autres techniques.
- Ils ont la capacité de traiter plus rapidement de grands ensembles de données.
- Ils ont la capacité d'incorporer différents types de données dans l'analyse y compris des données de différents capteurs.

L'avantage de la classification par les réseaux de neurones est que ces derniers ne nécessitent pas de modélisation analytique au préalable et ils ont une grande capacité d'apprendre par exemples. La classification de données multi-source par les réseaux de neurones ne présente pas les inconvénients rencontrés dans les autres méthodes, à savoir :

- Connaissance de la distribution des données pour l'approche probabiliste,
- Estimation et gestion de l'incertaine dans la théorie des croyances.

Le problème dû à la lenteur des méthodes d'apprentissage est un problème d'actualité. Un grand nombre de chercheurs et d'écoles tentent d'améliorer, du point de vue rapidité de convergence, les méthodes existantes, ou d'en proposer d'autres. [17]

- **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. 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.

L'algorithme de K Plus Proche Voisin se base essentiellement sur les deux éléments principaux suivants

- Le nombre de cas les plus proches (K) à utiliser et une métrique pour mesurer le plus proche voisin.
- La valeur de K est spécifiée à chaque utilisation de l'algorithme puisqu'il détermine le nombre de cas existants qui sont considérés pour prédire un nouveau cas.

Le K Plus Proche Voisin est basé sur le concept de distances. Une métrique est nécessaire pour déterminer les distances, cette dernière est à la fois importante car le choix de métrique influe beaucoup sur la qualité des prédictions et arbitraire du fait qu'il n'existe pas de définition préalable sur ce qui constitue une bonne métrique.

La méthode des K Plus Proches Voisins repose sur le regroupement des pixels en fonction de leur voisinage : chaque point est affecté à la classe la plus représentée parmi ses k plus proches voisins. Cette méthode requiert l'établissement d'une règle de distance et la détermination du nombre de voisins à prendre en considération, ainsi qu'un ensemble d'apprentissage représentant les différentes classes. [17]

## **B. Méthodes non supervisée**

L'intérêt des méthodes non supervisées est qu'elles ne nécessitent aucune base d'apprentissage et par là même aucune tâche préalable d'étiquetage manuel. La seule intervention de l'expert se situe à la fin du processus pour identifier les images.

Les algorithmes non supervisés les plus répandus tendent à découper l'espace des pixels en zones homogènes selon un critère de ressemblance et à minimiser une fonction coût, dépendant de la distance de chaque pixel aux prototypes (ou noyaux) des classes. Selon la certitude de la classification que nous voulons obtenir, et la relation entre les classes, nous pourrions distinguer plusieurs méthodes de classification, dans notre travail nous présentons les méthodes suivantes : K-Means, ensuite la méthode C-Moyennes Flous (FCM) d'où découle la méthode FCM\_S illustrées ci-dessous. [9]

- **L'Algorithme K-moyennes (K-Means)**

C'est l'un des algorithmes le plus connu et le plus utilisé en classification non supervisée du fait de sa simplicité de mise en œuvre. 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. Chaque classe de la partition est définie par ses objets et son centroïde.

L'algorithme K-Means à plusieurs inconvénients, la classification finale dépend du choix de la partition initiale et le choix aléatoire des centres initiaux, et la convergence vers un

minimum local à cause de l'initialisation aléatoire des centres et il est aussi fortement sensible au bruit.

- **L'Algorithme fuzzy c-means (FCM)**

Fuzzy c-means (FCM) est une méthode de regroupement qui permet à un élément de données d'appartenir à deux ou plusieurs clusters. Cette méthode développée par Dunn en 1973[27] et améliorée par Bezdek en 1981[28], il est fréquemment utilisé dans la reconnaissance de formes. Son processus de classification basé sur la minimisation de la fonction objective suivante [13]

$$J_m = \sum_i \sum_k u_{ik}^m * \|x_i - c_k\|^2 \quad (\text{I.1})$$

Où  $m$  est un nombre réel supérieur à 1,  $u_{ij}$  est le degré d'appartenance de  $x_i$  au classe  $j$ ,  $x_i$  est le  $i^{\text{ième}}$  des données mesurées à dimension  $d$ ,  $c_i$  est le centre à dimension  $d$  du groupe.

Le partitionnement flou s'effectue via une optimisation itérative de la fonction objective indiquée ci-dessus, avec la mise à jour de l'appartenance  $u_{ij}$  et des centres de classes  $c_i$  par

$$U_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}}} \quad (\text{I.2})$$

$$C_i = \frac{\sum_{j=1}^N \mu_{ij}^m * x_j}{\sum_{j=1}^N \mu_{ij}^m} \quad (\text{I.3})$$

L'approche de segmentation région a plusieurs avantages cités sur les points suivants

- La simplicité et la rapidité de la méthode,
- La segmentation d'objet à topologie complexe,
- La préservation de la forme de chaque région de l'image.

Mais il reste subit de plusieurs inconvénients comme l'influence du choix des germes initiaux et du critère d'homogénéité sur le résultat de la segmentation. Et une mauvaise sélection des germes ou un choix du critère de similarité mal adapté peuvent entraîner des

phénomènes de sous-segmentation ou de sur-segmentation. Il peut y avoir des pixels qui ne peuvent pas être classés.

#### I.4.3. Segmentation par les approches hybrides (coopérative)

La segmentation coopérative est une méthode permettant d'obtenir ces deux segmentations compatibles dans laquelle on relie l'extraction de plusieurs types de primitives ou d'informations (le gradient, écart type). Elle combine les avantages de chacune prise séparément : la précision et la rapidité d'une segmentation en contours, et la fermeture des frontières et la densité de l'information extraite, d'une segmentation en régions (Figure I-12). [7]

La segmentation par coopération région contour peut être exprimé comme une entre-aide entre ces deux concepts afin d'améliorer le résultat final. C'est une approche qui peut être une solution pour pallier aux inconvénients des deux approches. [34]



**Figure I-12 :** Segmentation d'image par coopération

On peut distinguer trois types de coopération, la coopération séquentielle, la coopération des résultats et la coopération mutuelle.

##### I.4.3.1. La coopération séquentielle

Dans laquelle l'une des techniques de segmentation (région ou contour) est réalisée en premier lieu ; son résultat va être exploité par l'autre technique pour renforcer la définition des critères ou des paramètres de la segmentation (Figure I-13).

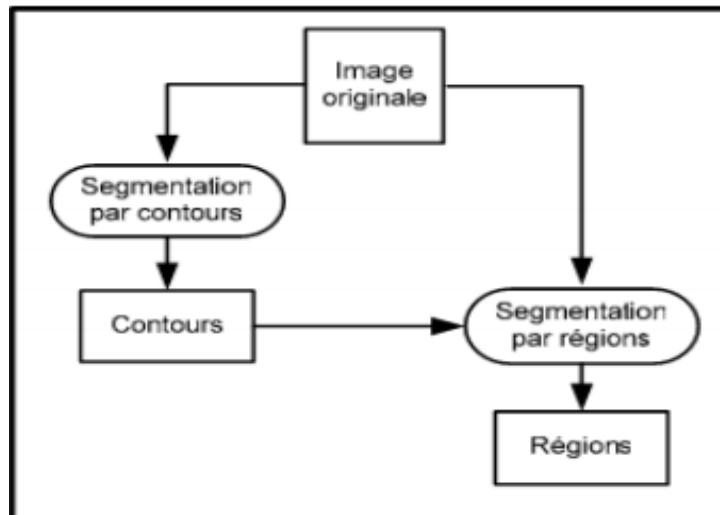


Figure I-13 : Principe de la coopération séquentielle.

#### I.4.3.2. La coopération des résultats

Les deux types de segmentation seront réalisés indépendamment ; la coopération concernera leurs résultats qui seront intégrés afin d'atteindre une meilleure segmentation (Figure I-14).

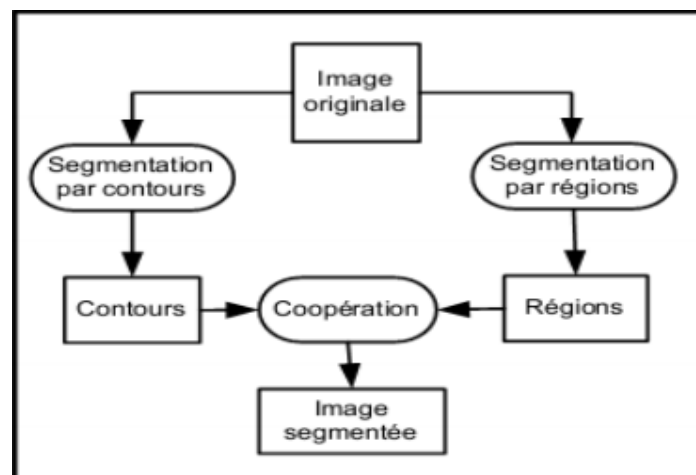


Figure I-14 : Principe de la coopération des résultats.

#### I.4.3.3. La coopération mutuelle

Les deux types de segmentations coopéreront mutuellement au cours de leur processus d'exécution (Figure I-15).

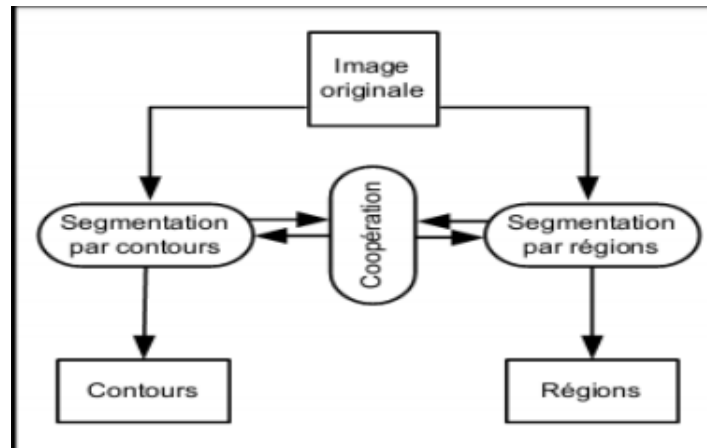


Figure I-15 : Principe de la coopération mutuelle.

### I.5. Conclusion

Dans ce chapitre, nous avons présenté une segmentation de l'image qui concentre sur les méthodes les plus courantes de fragmentation d'image, en général, tout en présentant les trois méthodes principales de (régions, contour et coopération (hybrides)), bien que chacune ait ses propres avantages et inconvénients. Dans le prochain chapitre, nous allons détailler le concept de classification, et en particulier l'algorithme de classification floue FCM et ses variantes.



# **Chapitre II**

## **La classification spatiale d'images**

## Chapitre II. **La classification spatiale d'images**

### II.1. **Introduction**

La classification est un processus très nécessaire pour les individus, c'est-à-dire une activité mentale qui se produit à chaque fois et qui se répète constamment dans la vie quotidienne. En fait, les choses sont souvent énumérées par rapport les classe ou aux catégories qui sont censées leur appartenir, ce qui est souvent vague ou graduelle.

Dans ce chapitre, nous discuterons de quelle classification et quelles sont les méthodes les plus importantes, où nous avons consacré l'essentiel de l'explication de non supervisé et que nous avons séparé dans l'algorithme utilisé (FCM) et dans certaines applications de la classification.

### II.2. **La classification d'image**

La classification est une méthode mathématique d'analyse de données, il est appliqué sur des données numériques (points, tableaux, images, sons, . . .etc. ), pour faciliter l'étude d'une population d'effectif important.

Classifier une image est une tâche ou une série de méthodes qu'une théorie unifiée pour pouvoir utiliser les images pour les analyses complémentaires ou pour la cartographie. Elle consiste à déterminer des procédures permettant de rassembler l'ensemble des pixels de l'image en un nombre limité de classes correspondant aux grands éléments structuraux de l'image. 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 [14].

Selon [15], on peut résumer essentiellement les étapes d'une classification sur les points suivant

- Choix des données.
- Calcul des similarités entre les  $n$  individus à partir des données initiales.

## Chapitre II : la classification spatiale d'images

- Choix d'un algorithme de classification et exécution.
- L'interprétation des résultats.

### II.3. Éléments de la classification

**La classe** Dans le contexte de la télédétection, une classe est un ensemble de pixels possèdent des caractéristiques semblables. On peut distinguer les classes thématiques, qui sont des catégories propres au thème traités : mur, forêt, sol nu etc..., et les classes spectrales qui sont constituées par des pixels de valeurs semblables. Les classes spectrales sont le plus souvent définies à l'aide de paramètres statistiques tels que la valeur moyenne, l'écart type, etc....

**La ressemblance** Pour déterminer la ressemblance d'un pixel avec une classe, on utilise les critères de distance ou de probabilité. Deux pixels se ressemblent au mieux si la distance entre eux est minimale.

**L'appartenance** Les critères d'attribution d'un pixel à une classe donnée sont établis par une fonction discriminante qui décidera de l'appartenance d'un pixel à une classe. On peut distinguer deux types de classes

- Les classes d'informations, c'est ce que l'analyse tente d'identifier sur les images.
- Les classes spectrales, c'est le regroupement de pixel qui ont les mêmes caractéristiques ou presque par exemple la valeur de densité, dans les différentes bandes spectrales des données.

L'objectif de la classification est de faire la correspondance entre les classes spectrales et les classes d'informations. [11]

### II.4. Application de la classification

La classification est été relié à innombrables domaines et applications tel que, le diagnostic médical, le traitement d'images aériennes, le traitement des eaux usées, l'étude du climat, les domaines de l'agriculture, la chimie, la géologie, le traitement d'image, la reconnaissance de la parole, la reconnaissance des caractères écrits, le traitement automatique des documents,...etc.

## II.5. Les méthodes de classification

Les méthodes de classification peuvent être schématiquement regroupées en deux grandes catégories, la classification supervisée (dirigée) et la classification non supervisée (non dirigée).

### II.5.1. Méthode supervisée

Dans la classification supervisée, le nombre de classes est connu et on dispose d'un ensemble de pixels déjà étiquetés, servant d'ensemble d'apprentissage. Il s'agit alors de pouvoir associer chaque nouveau pixel à la classe la plus adaptée en se servant des pixels déjà étiquetés. [16]

### II.5.2. Méthode non supervisée

Dans cette même catégorie de méthodes de classification, ils existent des méthodes non supervisées. Elles ne nécessitent aucune base d'apprentissage et aucune tâche préalable d'étiquetage manuel [16]. La seule intervention de l'expert se situe à la fin du processus pour identifier les classes trouvées. Les méthodes de classification non supervisées cherchent à estimer dans l'image les paramètres de leur modèle. Dans cette section nous décrivons brièvement les approches les plus connus tel que les k-moyennes et c-moyennes floues.

#### II.5.2.1. k-means

Ces méthodes construisent  $k$  classes à partir d'un ensemble de  $n$  individus, tout en minimisant la quantité :

$$\sum_{r=1}^k \sum_{x_i \in C_r} (x_i - g_r)^2 \quad (\text{II.1})$$

Ou :

- $C_r$  est la classe numéro  $r$
- $X_i$  est un individu dans une classe
- $G_r$  est le centre de classe  $C_r$ .

L'algorithme général de ces méthodes :

**Algorithme k-means**

Choisir  $k$  le nombre de classes désiré.

**Début**

- (1) Choisir  $k$  individus au hasard (comme centre des classes initiales)
- (2) Affecter chaque individu au centre le plus proche
- (3) Recalculer le centre de chacune de ces classes
- (4) Répéter l'étape (2) et (3) jusqu'à stabilité des centres
- (5) Editer la partition obtenue

**Fin**

Nous citons ici deux méthodes connues sur le principe de k-means sont :

- Méthode de centre mobile
- Méthode de nuée dynamique

Ces méthodes donnent la plupart de temps une partition localement optimale, il est donc conseiller d'effectuer plusieurs exécutions et comparer les différents résultats obtenues. Une suggestion consiste [29] à appliquer la méthode des k-means dans une première étape à plusieurs sous-ensembles de données extraits de l'ensemble total, et la meilleure partition obtenue fournit les centres à utiliser au départ de l'algorithme appliqué à l'ensemble total des données [30].

#### II.5.2.2. Le FCM

Fuzzy C-Means (FCM) est une méthode de classification non supervisée populaire qui a été largement appliquée pour la segmentation d'images qui s'appuie sur la logique floue, ou une donnée peut appartenir à deux ou plusieurs classes quand le nombre de classes est connu a priori. [18]

Ou en d'autres termes, l'algorithme des c-moyens flous (fuzzy c-means) est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité. L'algorithme nécessite de connaître le nombre de classes au préalable et génère les classes par un processus itératif en minimisant une fonction objective. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une région donnée. [19]

## Chapitre II : la classification spatiale d'images

L'algorithme utilise l'ensemble des pixels  $A = \{x_1, x_2, \dots, x_n\}$  où  $x_i$  est un vecteur de trois composantes (RVB par exemple), et le nombre de région  $c$ . Les valeurs des degrés d'appartenance sont regroupées dans une matrice de partition floue  $U = [u_{ik}]$  pour  $1 \leq i \leq n, 1 \leq k \leq c$  où  $u_{ik}$  désigne le degré d'appartenance du pixel  $i$  à la classe  $k$ . [19]

Pour avoir une bonne partition, on impose aux éléments de  $U$  les contraintes suivantes:

$$u_{ik} \in [0,1]$$

$$\sum_k u_{ik} = 1, \text{ ceci } \forall i \quad (\text{II.2})$$

l'équation (II.2) est une contrainte de normalisation qui assure que la somme des degrés d'appartenance de chaque observation à toutes les classes soit égale à 1.

L'algorithme FCM fait évoluer la matrice de partition floue  $U$  en minimisant la fonction objective (I.1)

Où  $m > 1$  est un paramètre contrôlant le degré de flou (généralement  $m = 2$ );  
 $c_k$  est le centre de la classe  $k$ .

Le partitionnement flou est réalisé par une optimisation itérative de la fonction objectif donnée par l'équation (II.3), avec une mise à jour des degrés d'appartenance  $u_{ik}$  et des centres des classes  $c_k$  comme dans le cas de l'algorithme K-means.

### Algorithme FCM

- Choisir le nombre de classes  $C$
  - Initialiser la matrice de partition floue  $U$ .
1. Mettre à jour les centres  $c_k \{k=1 \dots C\}$  en utilisant la formule (I.3)
  2. Mettre à jour les différents degrés d'appartenance selon la formule (I.2)

où 
$$d_{ij} = \|x_i - c_j\|$$

3. si  $|c_k^{(t+1)} - c_k^{(t)}| < \text{seuil}$ , pour  $k=1 \dots C$ , arrêter sinon aller à l'étape 1

Le FCM reste souffre de plusieurs inconvénients

1. Les degrés d'appartenance sont des degrés relatifs. Autrement dit, l'appartenance d'un individu à une classe dépend de l'appartenance de cet individu aux autres classes. Les fonctions d'appartenance estimations des centres des classes ne correspondent pas aux centres réels ou typiques.
2. Les points aberrants (éloignés) peuvent avoir des valeurs d'appartenance élevées et ils peuvent affecter de façon significative l'estimation des centres des classes.
3. Cet algorithme modélise dans la phase de classement l'hésitation ou l'ambiguïté entre classes à partir de règles de décisions floues fixées à priori. Dans la phase de classification, les points ambigus ou non représentatifs des classes peuvent alors affecter la position des centres. [24]
4. Le FCM nécessite une étape de spécification préalable de nombre de classes et initialisation des centres (initialisation aléatoire), cela permet de converger vers des résultats d'optimum local.
5. L'absence de l'utilisation de l'information des voisins dans le processus de classification rend le FCM faible dans le cas de présence du bruit sur l'image.

## II.6. La classification spatiale d'image

L'absence de l'utilisation de l'information spatiale au sein du processus de classification de FCM rend l'algorithme moins robuste contre l'effet du bruit. Dans ce contexte, plusieurs travaux ([23], [20], [31]) ont été proposés dans la littérature pour améliorer le FCM. On peut décomposer ces travaux en deux grandes catégories, la première cherche à calculer une nouvelle image en fonction de l'image d'entrée, de sorte qu'elle conserve la fonction cible et la fonction d'appartenance, notamment ([21], [22]), tandis que les méthodes de la seconde catégorie ont intégré l'information spatiale par la proposition d'une version modifiée de la fonction objective du FCM traditionnel ([23], [20], [31]).

### II.6.1. La génération d'une nouvelle image d'entrée

Szilágyi et al [21] a présenté une version modifiée du FCM traditionnel pour segmenter les images cérébrales, les auteurs ont proposé de calculer une image filtrée générée à partir des pixels de l'image originale et de la moyenne de leurs voisins locaux.

$$\xi_i = \frac{1}{1+\alpha} \left( x_i + \frac{\alpha}{\text{card}(N_i)} \sum_{r \in N_i} x_r \right), i = 1, \dots, N \quad (\text{II.3})$$

$N_i$  représente l'ensemble des pixels voisins à l'intérieur d'une fenêtre de voisinage centrée sur le  $i^{\text{ème}}$  pixel.

La fonction objective utilisée pour segmenter la nouvelle image générée  $\xi$  est définie par

$$J_s = \sum_{i=1}^c \sum_{l=1}^q \xi_l u_{il}^p (\xi_l - v_l)^2 \quad (\text{II.4})$$

$q$  désigne le nombre de niveaux de gris sur l'image, avec  $q \ll N$ .

$\xi_l$  représente le nombre de pixels ayant le niveau de gris  $l$  avec  $l=1, \dots, q$  et  $\sum_{l=1}^q \xi_l = N$

Sous la contrainte  $\sum_{j=0}^c u_{jl} = 1 (\forall l = 1, \dots, q)$ , on minimise  $J_s$  en annulant ses dérivées premières par rapport à  $u_{il}$  et respectivement pour en déduire :

$$u_{il} = \left[ \sum_{j=1}^c \left( \frac{(\xi_l - v_l)^{-2}}{(\xi_l - v_j)} \right)^{p-1} \right]^{-1} \quad (\text{II.5})$$



$$v_i = \left( \sum_{l=1}^q \gamma_l u_{il}^p \xi_l \right) \left( \sum_{l=1}^q \gamma_l u_{il}^p \right)^{-1} \quad (\text{II.6})$$

Dans cet algorithme, les auteurs notent qu'une valeur optimale de  $\alpha$  comprise entre 0.5 et 1.2, permet à la fois d'éliminer le bruit dans l'image et de préserver son détail.

Cai et al [22] proposent une autre variante de l'algorithme FCM, dans le but de segmenter une image à niveaux de gris. Cette méthode définit un facteur  $S_{ij}$  comme une mesure de similarité locale (spatiale et en niveaux de gris) entre un pixel  $i$  et son voisin  $r$  appartenant à une fenêtre de voisinage centrée sur le pixel  $i$ .

$$S_{ij} = \begin{cases} S_{s_{ij}} \times S_{g_{ij}}, & j \neq i, \\ 0, & j = i, \end{cases} \quad (\text{II.7})$$

Avec

$$S_{s_{ij}} = \exp\left(\frac{-\max(|p_j - p_i|, |q_j - q_i|)}{\lambda_s}\right) \quad (\text{II.8})$$

Où  $(p_i, q_i)$  représente les coordonnées spatiales du pixel  $i$  et  $\lambda_s$  est un facteur d'échelle.  $S_{s_{ij}}$  Introduit l'information spatiale locale

$$S_{g_{ij}} = \exp\left(\frac{-\|x_i - x_j\|^2}{\lambda_g \times \sigma_{g_i}^2}\right) \quad (\text{II.9})$$

$x_i$  est le niveau de gris du pixel central et  $x_j$  désigne le niveau de gris des pixels voisins dans la même fenêtre et  $\sigma_{g_i}$  est défini par

$$\sigma_{g_i} = \sqrt{\frac{\sum_{j \in N_i} \|x_j - x_i\|^2}{\text{card}(N_i)}} \quad (\text{II.10})$$

Sa valeur reflète le degré d'homogénéité en niveau de gris des pixels situés dans la fenêtre de voisinage, c'est-à-dire plus sa valeur est petite, plus la région locale est homogène. Ce paramètre peut être calculé à l'avance et sa valeur diffère d'un pixel à un autre.

A partir des facteurs  $S_{ij}$  une nouvelle image  $\xi$  est générée de la manière suivante

$$\xi_i = \frac{\sum_{j \in N_i} S_{ij} x_j}{\sum_{j \in N_i} S_{ij}} \quad (\text{II.11})$$

La méthode de Cai et al [22] revient à appliquer la version de Szilágyi et al [21] sur l'image pour accélérer le processus de convergence de l'algorithme.

### II.6.2. La Modification de fonction objective

Chuang et al [23] ont proposé un algorithme FCM spatiale, dans laquelle l'information spatiale peut être intégrée dans la fonction d'appartenance floue. La fonction spatiale est définie pour chaque pixel par la somme des fonctions d'appartenance de ses voisins qui sont définie par une fenêtre carrée centrée sur le pixel donné, selon la formule suivante

$$h_{ij} = \sum_{k \in NB(xi)} u_{ik} \quad (\text{II.12})$$

Où  $NB$  (représente une fenêtre carrée centrée sur le pixel dans le domaine spatial. Une fenêtre 5x5 a été utilisée dans ce travail. Tout comme la fonction d'appartenance, la fonction spatiale représente la probabilité que le pixel appartienne à son groupe. La fonction spatiale d'un pixel pour une le cluster est grand si la majorité de son voisinage appartient à la même classe. La fonction spatiale est intégrée à la fonction d'appartenance comme suit

$$u'_{ij} = \frac{u_{ij}^p h_{ij}^q}{\sum_{k=1}^c u_{kj}^p h_{kj}^q} \quad (\text{II.13})$$

où  $p$  et  $q$  sont des paramètres pour contrôler l'importance des deux fonctions ( $u_{ij}$  et  $h_{ij}$ ). Dans une région homogène, les fonctions spatiales renforcent l'appartenance d'origine et le résultat de la mise en cluster reste inchangé. Cependant, pour un pixel bruyant, cette formule réduit la pondération d'un cluster bruyant par les étiquettes de ses pixels voisins. En conséquence, les pixels mal classés des régions bruitées peuvent facilement être corrigés. D

Dans le cadre de la segmentation d'images IRM et en partant de l'idée que l'algorithme FCM est très sensible au bruit, Pham [20] a proposé la fonction objective suivante

$$j_{RFCM} = j_{FCM} + \sum_{j \in \Omega} \sum_{k=1}^c u_{jk}^q \sum_{l \in N_j} \sum_{m \in M_k} u_{lm}^q \quad (\text{II.14})$$

$$j_{FCM} = \frac{\beta}{2} = \sum_{j \in \Omega} \sum_{k=1}^c u_{jk}^q \|y_j - v_k\|^2 \quad (\text{II.15})$$

Où  $\beta$  est un paramètre qui pondère l'influence du deuxième terme de la fonction objective.

## Chapitre II : la classification spatiale d'images

$N_j$  : représente l'ensemble des pixels voisins du pixel  $y_j$

$M_k = \{1 \dots k\} \setminus \{k\}$  représente l'ensemble des classes sauf la classe  $k$ .

Dans cet algorithme, l'appartenance d'un pixel à une classe est conditionnée par l'appartenance de ses voisins aux autres classes. En d'autres termes si l'appartenance des pixels voisins aux autres classes est faible alors cela augmente l'appartenance du pixel  $y_j$  à la classe  $k$ . Cette méthode dépend donc des valeurs des autres classes et non pas uniquement de la classe envisagée. Comme il n'y a pas de contraintes sur les centres des classes, l'équation qui calcule les  $v_k$ , reste identique à celle de FCM. En revanche la fonction de partition floue  $U$  est différente, elle est obtenue par optimisation Lagrangienne

$$u_{jk} = \frac{\left( \|y_j - v_k\|^2 \beta \sum_{l \in N_j} \sum_{m \in M_k} u_{lm}^q \right)^{-1/(q-1)}}{\sum_{i=1}^C \left( \|y_j - v_i\|^2 \beta \sum_{l \in N_j} \sum_{m \in M_k} u_{lm}^q \right)^{-1/(q-1)}} \quad (\text{II.16})$$

### II.7. Conclusion

La classification consiste à grouper les pixels de l'image ayant des caractéristiques assez similaires sauvent au niveau de gris, en des sous ensembles de classes (segmentes, ou cluster). Dans ce chapitre nous avons présenté un aperçu sur le sens de la classification et les différentes approches utilisées dans cette méthode, supervisées et non supervisées. Ensuite, on a présenté quelques variantes de l'algorithme FCM proposé dans la littérature pour améliorer sa robustesse contre l'effet du bruit.

# **Chapitre III**

## **La conception**

## Chapitre III. **La conception**

### III.1. **Introduction**

La conception s'agit d'un processus visant à formaliser les phases initiales du développement logiciel pour répondre à tous les besoins. Le lancement est toujours une déclaration informelle (résultat souhaité) et nécessite une recherche en profondeur d'informations provenant d'utilisateurs expérimentés et fonctionnels (futurs utilisateurs du programme) avec analyse bon et logique.

À travers ce que nous avons examiné dans les deux chapitres précédents, nous avons pu comprendre et définir les bases du démarrage de notre projet, et travailler pour établir et développer notre programme clairement et sans ambiguïté, avec l'introduction de la structure générale et spécifique de notre programme, à partir de là, nous expliquerons et développerons toute cette structure.

### III.2. **Conception générale**

Dans cette section, nous examinons la structure générale du système, qui consiste en des entrées d'une image bruitée et des algorithmes utilisés dans la classification, puis nous passons au processus de l'étiquetage ainsi que la comparaison de transition entre les résultats des algorithmes. (Figure III-1).

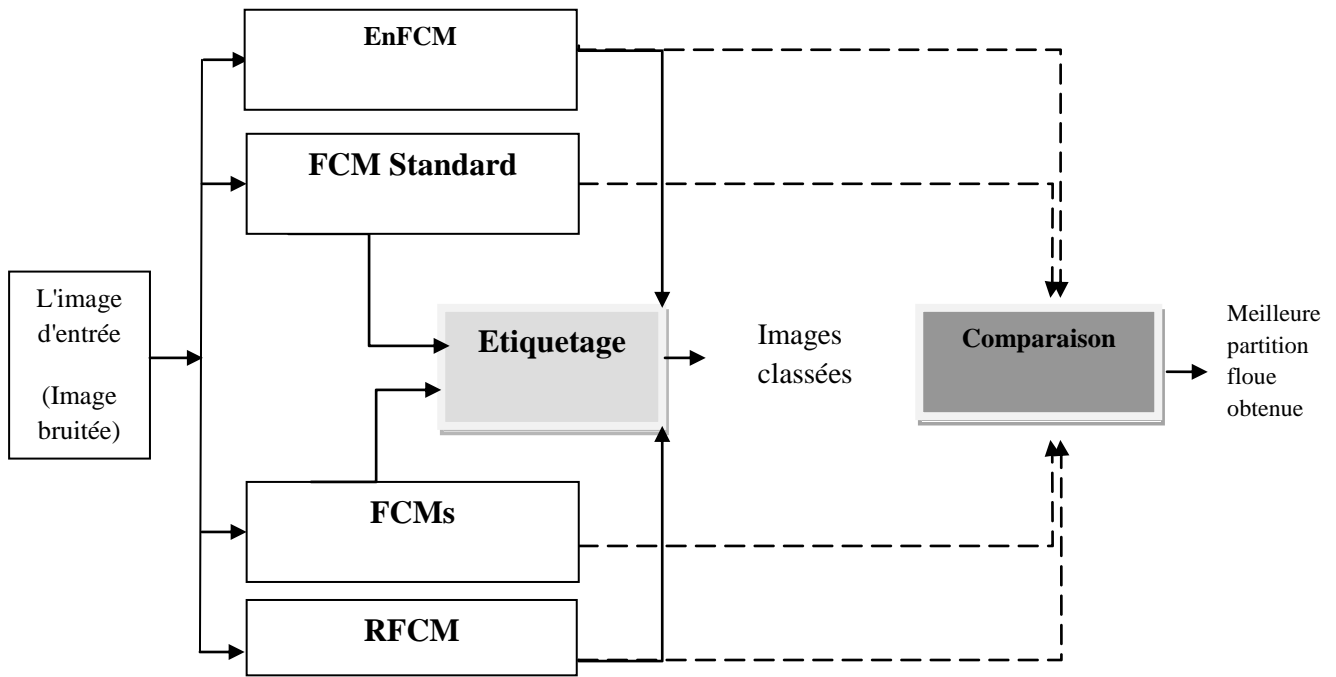


Figure III-1: Architecture Globale du système

Au départ nous choisissons l'image que nous souhaitons classer, ensuite nous appliquons l'un des quatre algorithmes suivants (EnECM, FCM standard, FCMs, RFCM). Après la convergence du processus de classification floue de l'algorithme choisi, une étape d'étiquetage aura un lieu pour obtenir l'image classée. L'étape de comparaison est une étape semi automatique, dans laquelle nous utilisons quelques fonctions d'évaluation floues sur les centres et la matrice des degrés d'appartenance obtenus par chaque algorithme.

### III.2.1. Image d'entrée

L'image d'entrée de notre application est une image bruitée numérique choisie par l'utilisateur sous format JPEG, PNG, JPG, ... cette image peut être en couleur ou bien au niveau de gris.

### III.2.2. Les méthodes de classification

Les méthodes utilisées pour classer l'image dans notre travail ils ont des méthodes de classification floue basée sur les algorithmes (FCM standard, EnFCM, FCMs et notre algorithme RFCM), les trois derniers algorithmes intègrent l'information spatiale sous le processus de classification classique du FCM standard. Le processus de classification de chaque algorithme est un processus itératif basé sur deux étapes essentielles la mise à jour de la matrice des degrés d'appartenance et la mise à jour des centres des classes.

### III.2.3. L'étiquetage

Après la convergence de l'algorithme de classification floue, on applique l'étape d'étiquetage, l'objectif de cette étape est d'affecter la même couleur à l'ensemble des pixels de la même classe.

### III.2.4. Les méthodes de comparaison

Dans cette étape, nous comparons les quatre algorithmes implémentés en termes de meilleure partition floue obtenue. Il existe plusieurs fonctions (indices) d'évaluation des partitions floues [23][31][32][33], les indices utilisés dans cette section sont (Partition entropie ( $V_{PE}$ ) [31] , Partition coefficient ( $V_{PC}$ ) [31], et Xie Beni ( $V_{XB}$ ) [32]).

**Partition Entropie** ( $V_{PE}$ ) cela fonctionne au détriment d'un somme total de produit entre la fonction d'appartenance et son logarithme, selon la formule suivante

$$V_{PE} = -\frac{1}{N} \sum_{i=1}^C \sum_{j=1}^n u_{ij} \log u_{ij} \quad (\text{III.1})$$

La meilleure partition est obtenue lorsque  $V_{PE}$  est minimal.

**Partition Coefficient (Pc), l'indice**  $V_{PC}$  cela fonctionne à la somme du carré total de la fonction d'appartenance, comme montre la formule suivante

$$V_{PC} = \frac{1}{N} \sum_{i=1}^C \sum_{j=1}^n u_{ij}^2 \quad (\text{III.2})$$

La meilleure partition est obtenue lorsque  $V_{PC}$  est maximale.

**L'indice Xie Beni** ( $V_{XB}$ ), utilise deux informations, l'intra-variation et l'inter-variation (la séparation), l'intra-variation mesure la variation à l'intérieur de chaque classe, tandis que l'inter-variation mesure la séparation entre les différents centres, l'indice  $V_{XB}$  est défini par la formule suivante

$$V_{XB} = \frac{\sum_{i=1}^C \sum_{j=2}^n u_{ij}^m \|x_j - v_i\|^2}{N \times \min_{i \neq k} \|x_j - v_i\|^2} \quad (\text{III.3})$$

La meilleure partition est obtenue lorsque l'intra-variation minimale et inter-variation maximale, alors lorsque  $V_{XB}$  tend vers une valeur minimale.

### III.3. La conception détaillée

Dans la conception détaillée, nous obtenons une description des structures de données des algorithmes et de leurs connexions.

Notre application accepte comme une entrée une image numérique, après la lecture, l'utilisateur peut classer son image par l'un des quatre algorithmes de classification (FCM standard, EnFCM, sFCM, RFCM).

- L'algorithme (FCMs), comme nous l'avons appris dans le deuxième chapitre, l'algorithme FCMs [23] intègre l'information spatiale dans le processus de classification floue par la combinaison des degrés d'appartenance globale et locale en utilisant les équations (II.15) et (II.16).
- L'algorithme (RFCM), exprime notre méthode proposée dans laquelle nous avons intégré l'information spatiale, cette dernière est assurée par la génération d'une nouvelle image à partir l'image d'entrée, l'image générée sera l'image d'entrée du processus de classification du FCM standard.
- L'algorithme (EnFCM), comme nous l'avons appris dans le deuxième chapitre intègre l'information spatiale par le calcul d'une nouvelle image en utilisant l'équation (II. 3).
- L'algorithme (FCM standard), n'utilise aucune information spatiale dans le processus de classification.



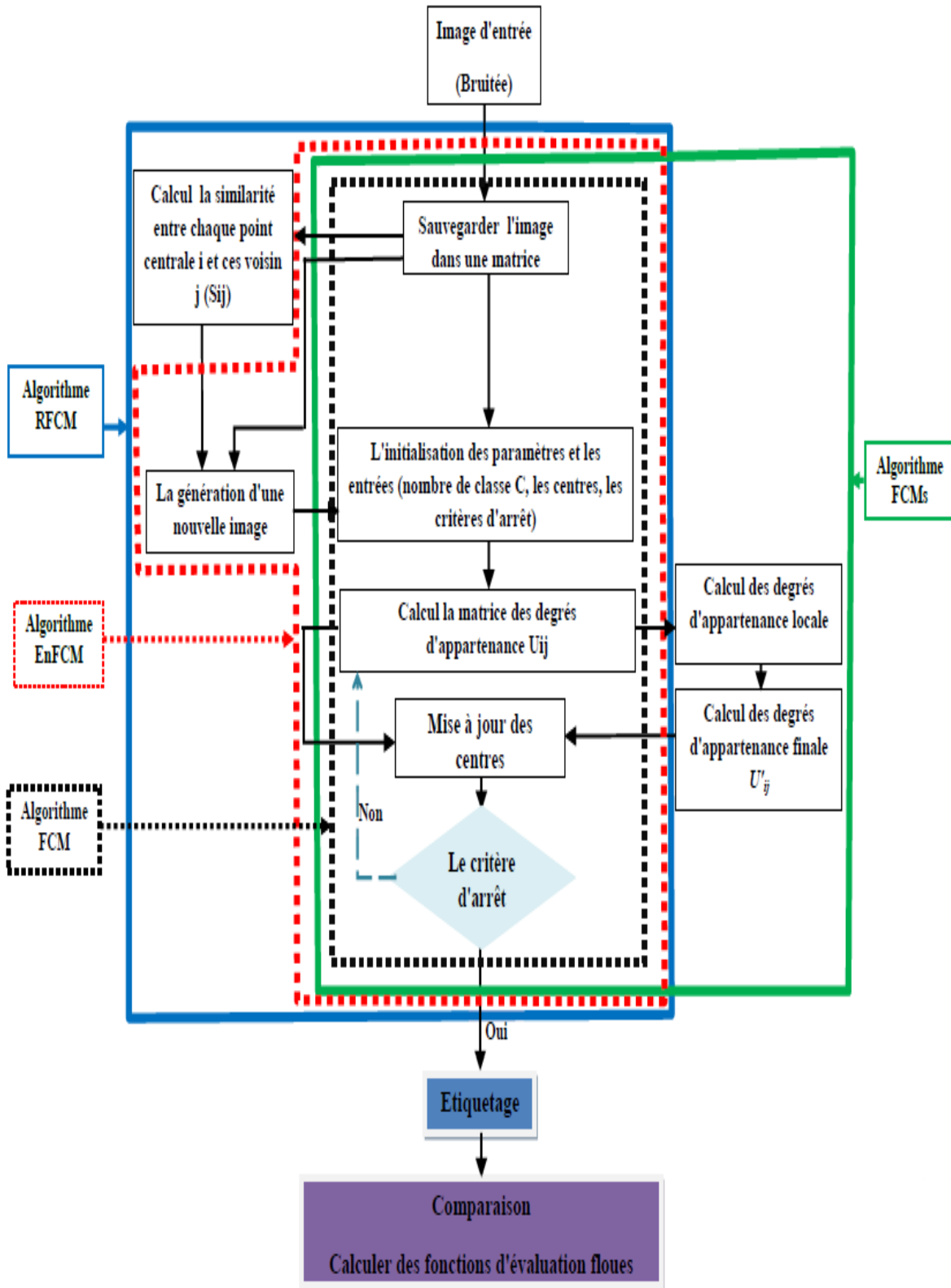


Figure III-2 : L'architecture détaillée du système

### III.3.1. Sauvegarde de l'image d'entrée

Notre application accepte comme une entrée une image bruitée. s'il s'agit d'une image couleur, un pixel est codé par trois composantes (r, g, b) (chacun compris au sens large entre 0 et 255), représentant respectivement les "doses" de rouge, vert et bleu qui caractérisent la couleur du pixel s'il s'agit d'une image au niveau de gris, il est codé par un composante comprise au sens large entre 0 et 255 représentant la luminosité du pixel.

La structure de données utilisée pour stocker l'image est une matrice 2D **mat** [**nb\_ligne**][ **numBands**] tel que **nb\_ligne** est le nombre total des pixels de l'image d'entrée (**nb\_ligne=height \* width**) et **numBands** correspond au nombre de canaux sur l'image d'entrée où il est égal à trois pour les images couleurs et un pour les images à niveau de gris.

### III.3.2. La génération d'une nouvelle image

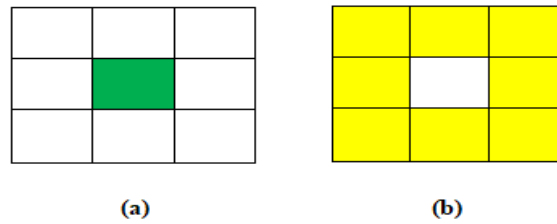
Dans notre algorithme (RFCM), nous avons adopté la stratégie de génération d'une nouvelle image pour exploiter l'information spatiale dans le processus de classification afin d'améliorer la robustesse du FCM standard contre l'effet du bruit.

Le terme de similarité  $S_{ij}$  exprime le degré de contribution de chaque point des voisins  $j$  (**Figure III-3**) sur la définition de l'intensité (couleur) du point central  $i$ , il correspond à la somme des distances des degrés d'appartenances aux différentes classes entre le point  $i$  et ses voisins  $j$ ; la mesure de similarité définie par la formule suivante

$$S_{ij} = \sum_{k=1}^C \|\mu_{ki} - \mu_{kj}\| \quad (\text{III.4})$$

Où  $C$  est le nombre de classes.

$\mu_{ki}, \mu_{kj}$  correspond respectivement aux degrés d'appartenance des pixels  $i$  ( $i$  est le point central) et  $j$  ( $j$  est le pixel voisin) à la classe  $k$ .



**Figure III-3** : (a) le point central, (b) les voisins du point central

Après le calcul de la similarité entre chaque point central et ses voisins on peut maintenant générer la nouvelle image en utilisant la formule ci-dessous

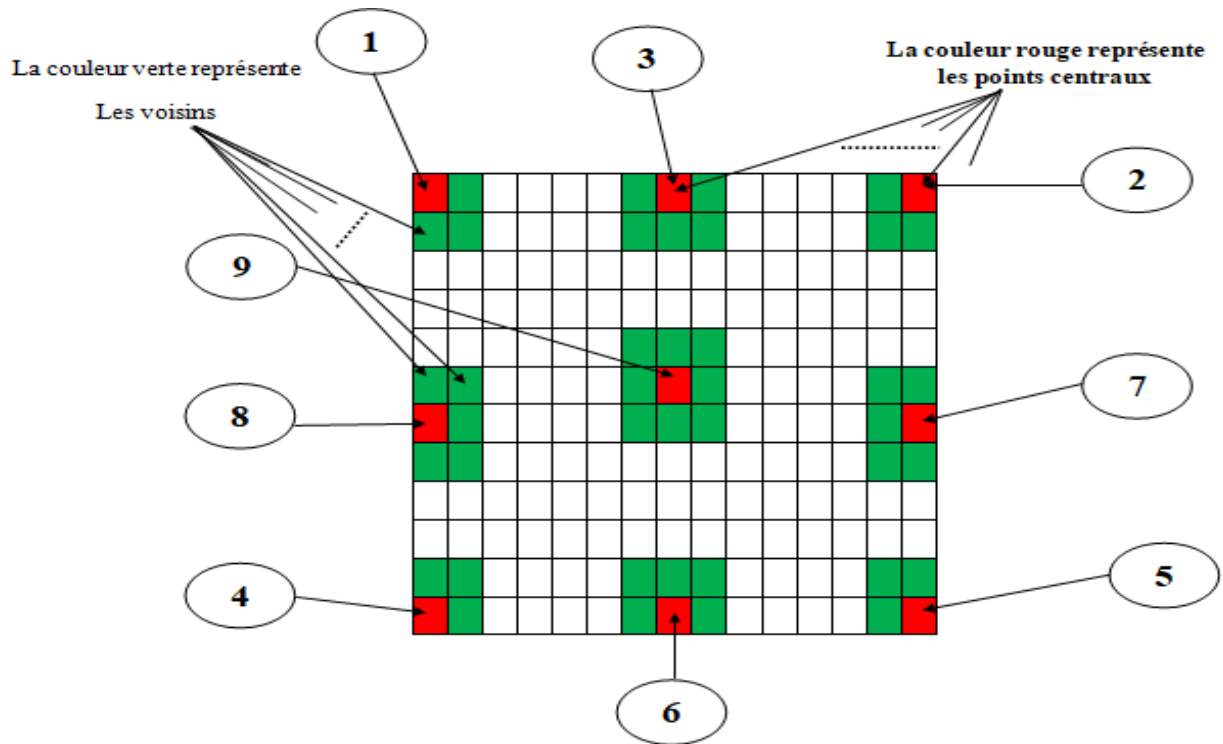
$$y_i = \frac{x_i + \sum_{k=1}^N \exp^{-S_{ik}} \times x_k}{1 + \sum_{k=1}^N \exp^{-S_{ik}}} \quad (\text{III.5})$$

$y_i$  Sont les pixels de la nouvelle image générée.

$x_i$  Sont les pixels de l'image d'entrée.

$S_{ik}$  Est la similarité entre chaque point centrale  $i$  et ses voisins  $k$ .

Pour calculer les niveaux d'influence des voisins pour chaque point central on a utilisé le terme de similarité définie dans la section précédente, et cette partie nécessite l'étude de tous les cas possible de position des voisins, pour notre travail nous définissons les voisins d'un pixel central dans une fenêtre de taille 3\*3 (La Figure III-4).



**Figure III-4 :** les cas possibles des positions de pixel central (rouge) et ses voisins (vert) dans une fenêtre de taille 3\*3.

Pour l’algorithme EnFCM, il n’y a aucun terme de similarité a été utilisé parce que les degrés d’influence des voisins sont similaires, et pour calculer la nouvelle image nous avons utilisé la formule (II.3).

Pour la structure de données utilisée pour stocker la nouvelle image générée pour notre algorithme RFCM et l’algorithme EnFCM, nous avons utilisé une matrice 2D, **CalculeNewImageR** [nb\_ligne][ nb\_col], **CalculeNouvImgEn** [nb\_ligne][ nb\_col], tel que nb\_ligne est le nombre total des pixels de l’image d’entrée (**nb\_ligne=height \* width**) et **numBands** correspond au nombre de canaux sur l’image d’entrée, il est égale à trois pour les images couleurs et un pour les images à niveau de gris.

**Algorithme Génération d'une nouvelle image**

**Les entrées**

- Image  $I$ , nb\_colonne, nb\_ligne

**Les sorties**

- la nouvelle image  $NI$

**pour** chaque pixel  $i$  de l'image  $I$  **faire**

*//cas 1*

**Si** ( $i=1$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 2*

**Si** ( $i=nb\_colonne$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 3*

**Si** ( $i<nb\_colonne$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 4*

**Si** ( $i=nb\_ligne*nb\_colonne$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 5*

**Si** ( $i=nb\_ligne*nb\_colonne$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 6*

**Si** ( $i>(nb\_ligne-1)*nb\_col \ \&\& \ i<nb\_ligne*nb\_col-1$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 7*

**Si** ( $i\%nb\_colonne=nb\_colonne-1$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

**Sinon**

*//cas 8*

**Si** ( $i\%nb\_colonne = 0$ ) **alors**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

*//cas 9*

**Sinon**

Calculer la similarité  $S_{ij}$  entre les points centrale  $i$  et leur voisin  $j$  par l'équation (III.4)

Calculer le nouveau pixel  $y_i$  par l'équation (III.5)

### III.3.3. L'initialisation des paramètres d'entrées

La classification floue exige une étape de spécification de nombre de classes  $C$  ainsi que les valeurs des différents centres et le seuil du critère d'arrêt.

L'étape de spécification de nombre de classes nous la laisse à l'intervention de l'utilisateur. Pour initialiser les centres de classe, nous commençons par la première estimation aléatoire des centres de classes, il s'agit de l'emplacement moyen de chaque classe. L'initialisation aléatoire des centres de classes provoque une convergence vers des solutions des minima locaux.

La structure utilisée pour garder les centres calculés par l'équation (II.2) est une matrice 2D **clusterCenters** [**numClusters**] [**numBands**] dont les lignes sont les nombres des classes et les colonnes sont le nombre des bandes.

L'étape d'initialisation des paramètres d'entrées est une étape commune pour les quatre algorithmes (FCM Standard, EnFCM, FCMs et RFCM).

### III.3.4. Calcul des centres et des degrés d'appartenance

La minimisation de la fonction objective ( $U, C$ ) (II.1) peut être amenée par un processus itératif dans lequel la mise à jour du degré d'appartenance  $U_{ij}$  et les centres de classes sont faits pour chaque itération de la classification.

Dans cette étape on cherche à calculer des degrés d'appartenance de notre algorithme (RFCM) et les autres algorithmes (FCM Standard, FCMs et EnFCM) en utilisant l'équation (I.2), la structure de données que nous avons utilisées pour stocker la matrice des degrés d'appartenance est une matrice 2D **memberships** [**le nombre de pixels**] [**numClusters**] tel que le nombre de pixels est égale à la taille de l'image d'entrée (**width\*height**) et **numClusters** est le nombre de classes.

La structure utilisée pour garder les centres calculés par l'équation (I.3) est une matrice 2D **clusterCenters** [**numClusters**] [**numBands**] dont les lignes sont les nombres des classes et les colonnes sont le nombre des canaux.

### III.3.5. Le critère d'arrêt

Le critère d'arrêt utilisé pour les quatre algorithmes, soit après certain nombre d'itérations spécifié soit lorsque les centres des classes n'évoluent plus entre deux itérations successives (c.-à-d. le changement des centres entre deux itérations successives ne dépasse pas certain seuil).

### III.3.6. L'étiquetage

L'objectif de cette étape est la construction de l'image finale à partir de la matrice des degrés d'appartenance obtenu après la convergence de l'algorithme de classification, et pour réaliser cette étape on affecte chaque couleur (les pixels caractérisent par cette couleur) à sa propre classe (la classe dont le degré d'appartenance est le plus élevé)

## III.4. Les algorithmes utilisés

### Fonction `Fonction_Objective ()`;

Variable `index,w,c, width, height, numcluster`;

début

```
    Si (w < width * height) Alors
        Si (c < numcluster) Alors
            index = Pow (degré d'appareance [w][c], fuzziness) * Pow ( ||Disance entre point w
            et centre de classe k||,2);
            FinSi
        FinSi
    FinSi
Returnner index;
Fin
```

### Fonction `calculateCenters()`;

Variabl,`w,c, width, height, numbands`: entier

`bottom, top` :réel;

début

## Chapitre III : La conception

```
Si (w < width * height * numbands) Alors
|
|   Si (c < width * height * numbands) Alors
|   |
|   |   top = membership [w][c] * le pixel j ;
|   |   bottom = membership [w][c];
|   |   FinSi
|   FinSi
FinSi
```

Returner (top /bottom);

Fin

### **Fonction CalculateMemberShip ();**

Variabl,w,c, width , height, numbands, numcluster: entier

bottom ,top, memberships:réel;

début

```
Si (w < numcluster) Alors
|
|   Si (c < width * height * numbands) Alors
|   |
|   |   top =Distance entre le pixel j et centre de leure classe i ;
|   |   bottom = Distance entre le pixel j et les centres des autres classe k;
|   |   memberships =Pow(|| top|| / Pow (||Botom||,2),2/(m-1))
|   |   FinSi
|   FinSi
FinSi
```

Returner(1/ memberships);

Fin

### **Fonction getImageInMatriceNouv (variable mat[][] : réel);**

Variable sumterms : de réel ; h , i , b , w,ck : d'entier;

début

```
mat [height*width][ numbands];

i ←0;

Pour(h← 0 à height) Faire
|
|   Pour(w ← 0 à width) Faire
|   |
|   |   Index ← (h*width+w)*numBands;
|   |
|   |   Pour(b ← 0 à numbands) Faire
```



## Chapitre III : La conception

```
mat[i][b] ← inputData2[index+b];
```

```
i ← i+1;
```

```
Fin Pour
```

```
Fin Pour
```

```
Returner mat;
```

```
Fin
```

**Fonction Indice\_Xie\_Beni ( ) ;**

Variable w, b, c, c1, c2, index : entier ;

DistancePixelToCluster, minDist, distance, cs : réel;

**Début**

```
Pour(w ← 1 à (width*height))Faire
```

```
Index w*numBands;
```

```
Pour(b ← 1 à numBands)Faire
```

```
a Pixel [b] = InputData [index + b];
```

```
Fin Pour ;
```

```
Pour(c ← 1 à numClusters) Faire
```

```
DistancePixelToCluster = calcSquaredDistance (a Pixel,clusterCenters[c]);
```

```
cs+ = Pow (membership[w][c]m fuzziness ) *
```

```
DistancePixelToCluster*DistancePixelToCluster;
```

```
Fin Pour ;
```

```
Fin Pour;
```

```
cs cs/ (height*width) ;
```

```
minDist = Float.MAX_VALUE;
```

```
Pour(c1 ← 1 à (numClusters-1))Faire
```

```
Pour(c2← c1+1 à numClusters)Faire
```

```
Distance calcSquaredDistance(clusterCenters [c1],clusterCenters[c2]);
```

```
MinDist minimum (minDist, distance);
```

```
Fin Pour ;
```

```
Fin Pour ;
```

## Chapitre III : La conception

cs= cs/(minDist\*minDist);

Retourne (cs);

**Fin.**

**Fonction getPartitionEntropy ( );**

Variable width , height, numcluster, numBands: entier

**Début**

Pe : réel;

Pe =0;

**Pour**(h ←1 à (height\*width))Faire

**Pour**(c ← 1 à numClusters)Faire

Pe← Pe + memberships[h][c] \* log (memberships[h][c]);

**Fin Pour** ;

**Fin Pour** ;

pe ←-pe / (height \* width \* numBands);

Retourne (pe);

**fin**

**Fonction getPartitionCoefficient ( );**

Variable width , height, numcluster, numBands: entier

**Début**

Pc : réel;

Pc =0;

**Pour**(h← 1 à (height\*width))Faire

**Pour**(c← 1 à numClusters)Faire

$P_c \leftarrow P_c + \text{memberships}[h][c] * \text{memberships}[h][c];$

**Fin Pour** ;

**Fin Pour** ;

$pc \leftarrow pc / (\text{height} * \text{width} * \text{numBands});$

Retourne (pc);

**fin**

### III.5. Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes de notre algorithme (RFCM) de classification et les autres algorithmes utilisés (EnFCM, Standard FCM, sFCM), ainsi que la conception générale et détaillée, en expliquant chaque étape de la conception détaillée de chaque algorithme, ainsi que la manière dont les informations spatiales sont utilisées dans notre algorithme proposé. Le chapitre suivant est consacré à l'implémentation de notre application.

# **Chapitre IV**

## **L'implémentation**

## Chapitre IV. **L'implémentation**

### IV.1. **Introduction**

Dans ce dernier chapitre et après l'aperçu théorique des chapitres précédents, nous présenterons le côté pratique de l'application. Notre but est la réalisation d'un système robuste et fiable au bruit et contours flous qui fais la classification des images, nous commençons par la description de la base utilisée, le choix de l'environnement de travail en passons par le langage et logiciel de codage ainsi que des résultats sur différentes images avec deux tables pour les résultats des indices comparatifs entre les quatre algorithmes et pour le temps d'exécution.

### IV.2. **L'environnement de travail**

Pour que notre travail atteigne l'objectif qu'on visait, on a pris l'initiative d'exploiter et d'implémenter notre algorithme sur : Windows 10. Ce choix se traduit par l'efficacité de cet environnement en ce qui concerne la structure d'interaction événementielle qu'elle dispose pour communiquer avec des applications actives, ainsi que les ressources de la machines qu'il offre aux différentes applications , enfin , son système d'allocation de mémoire qui est un des meilleurs présents dans ce domaine.

### IV.3. **Le langage de codage**

Notre application a été codée en sa globalité par le langage JAVA à travers la plateforme NetBeans ce choix repose sur le fait qu'il est :

Populaire : en particulier en entreprise, le langage Java est un investissement pour celui qui l'apprend.

Riche : il existe de nombreuses bibliothèques dans tous les domaines ; celles-ci ont l'avantage considérable d'être standardisées.

International : les programmes Java supportent les normes internationales Unicode pour le codage des caractères de toutes les langues.

Propre : il préserve un typage assez fort des données en présence, les Operations arithmétiques sont très bien spécifiées.

#### IV.4. Logiciel de codage

Les environnements de développement intégrés (EDI), sont des logiciels regroupant un ensemble d'outils nécessaires au développement logiciel dans un (ou plusieurs) langage(s) de programmation (Java, PHP, Ruby, C/C++, ... etc.).

Parmi tous les environnements de développement existant notre choix a porté sur NetBeans (Figure IV-1) à la faveur de sa rapidité à mettre en place une application qui est l'un des points forts de ce dernier

Une boîte à outils qui permet de gagner beaucoup de temps en développement et en effort.

Un ensemble de Standards pour rehausser et renforcer la consistance et l'interopérabilité entre les applications et les systèmes d'exploitation.

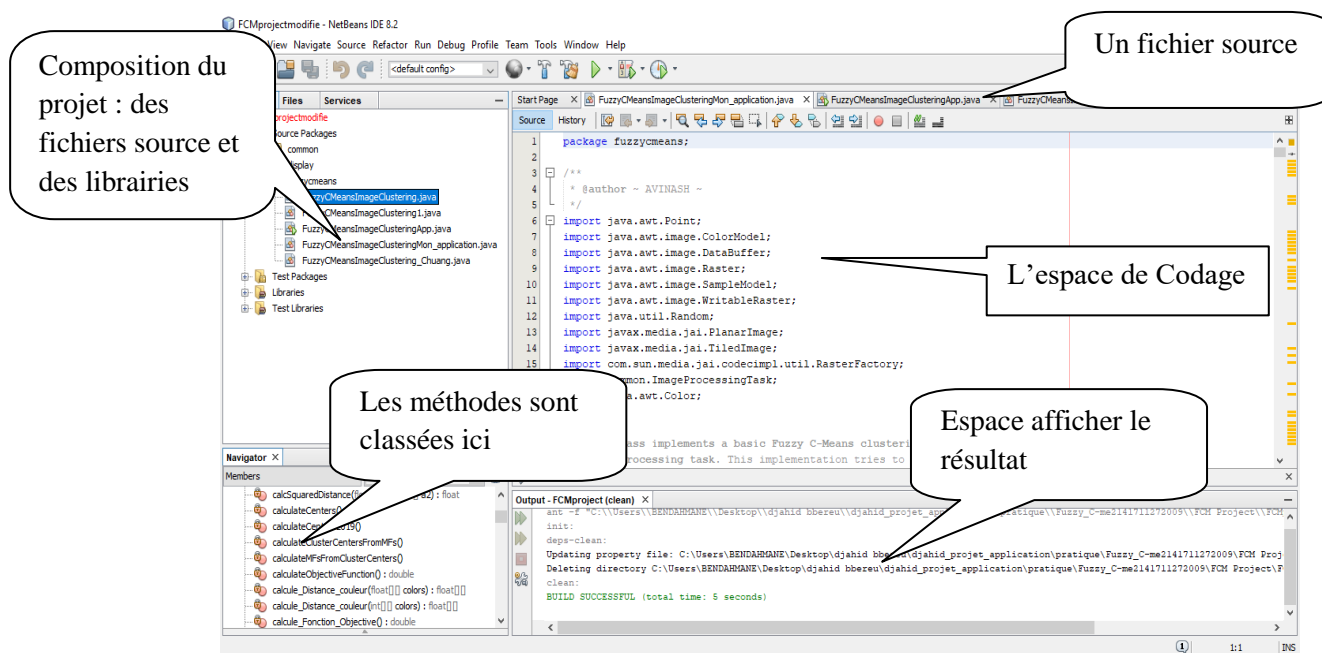


Figure IV-1: l'interface de l'environnement

#### IV.5. Bibliothèques utiliser

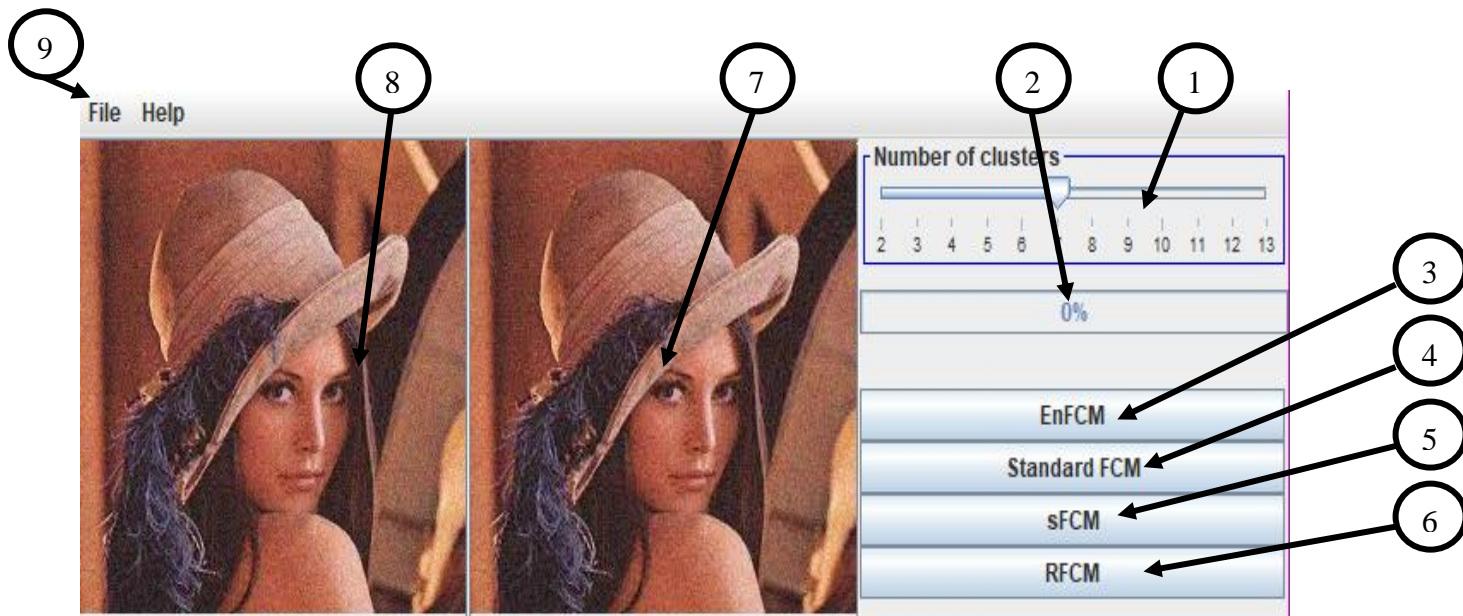
- jackson-core-2.5.3
- jackson-jr-objects-2.5.3
- jai\_core
- java-scanner-access-twain-12.0.9

#### IV.6. Caractéristiques de la machine

L'implémentation du projet est réalisé sur un ordinateur portable qui se caractérise techniquement par :

- Un processeur Intel® Core(TM) i5-3210M CPU @2.50 GHz 2.50 GHz.
- Windows 10 pro ,64 bit.
- Une Mémoire RAM de 8 GB (7.89 GB usable).
- Une carte graphique GeForce GT 630M de version 376.54.

#### IV.7. Présentation de l'interface de l'application



**Figure IV-2:** Fenêtre d'accueil

(1) Le nombre de classe, Afin de sélectionner le nombre de catégories utilisées dans le processus de classification par scrolle Barre

(2) La barre de chargement pour présenté le temps restant à exécuter au moyen d'un composant de type Progrès Barre

(3) Pour lancer l'algorithme EnFCM, au moyen d'un composant de type bouton.

(4) Pour lancer l'algorithme FCM standard, au moyen d'un composant de type bouton.

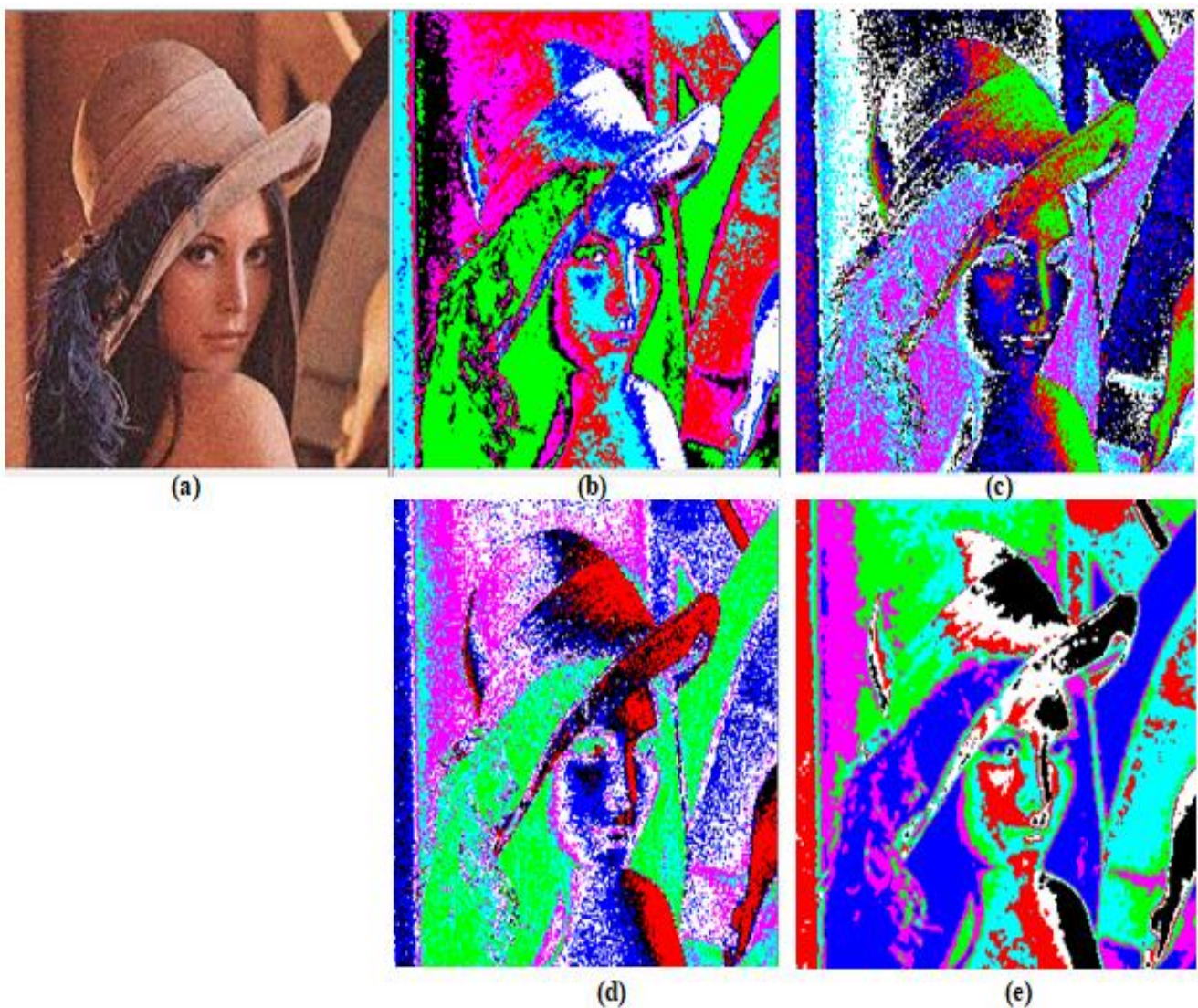
(5) Pour lancer le programme sFCM, au moyen d'un composant de type bouton.

(6) Pour lancer le programme RFCM (Notre algorithme), au moyen d'un composant de type bouton.

(7) L'image après résultat. L'espace pour afficher l'image classée dans un Panel

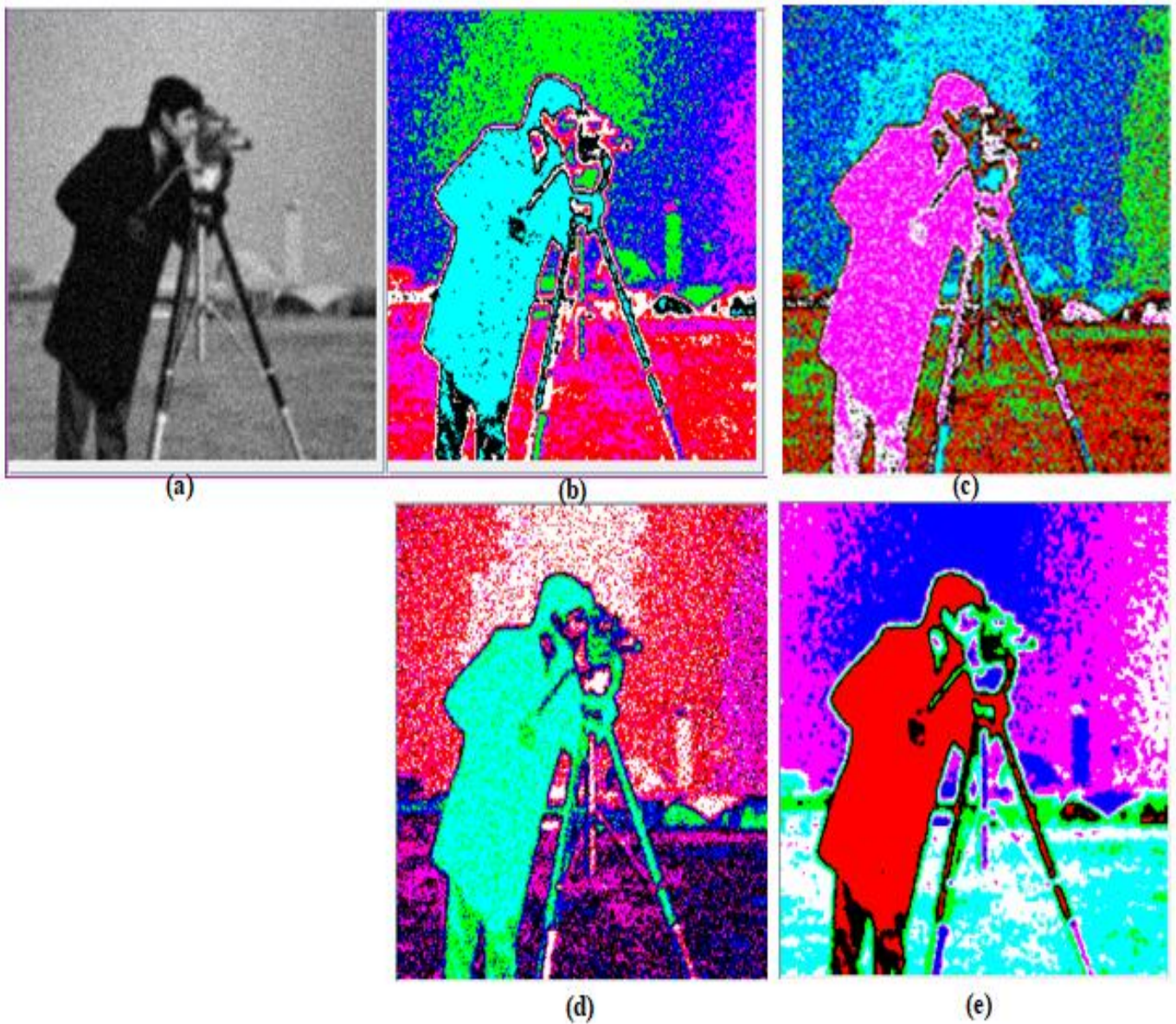
(8) L'image initiale. L'espace pour afficher l'image d'entrée qui a été sélectionné par l'utilisateur dans un Panel.

(9) File: (Open file) pour sélectionner l'image d'entrée, et (Exit) pour fermer le programme, au moyen d'un composant de type Menu Barre

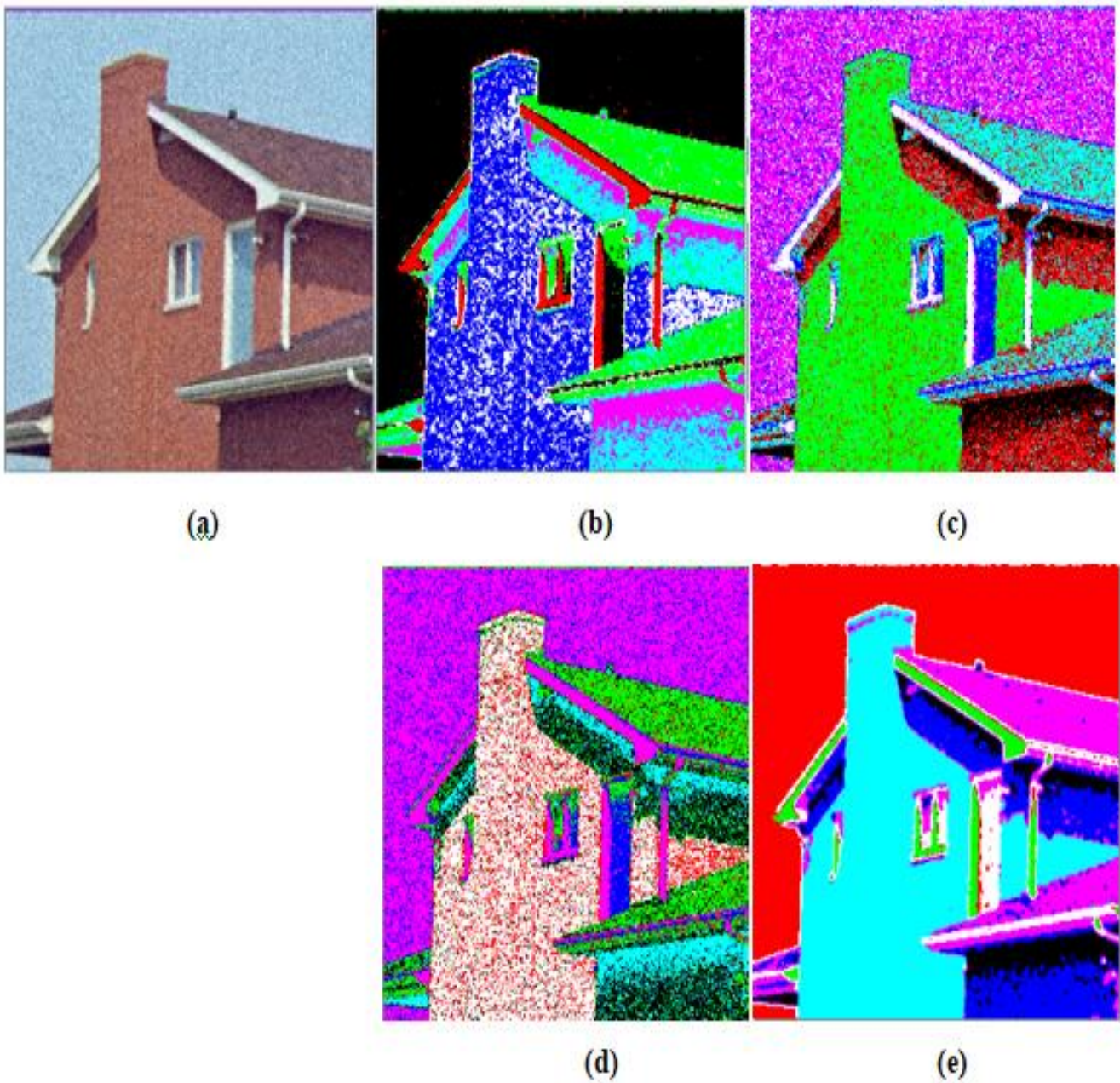


**Figure IV-3** : Résultat de classification des différents algorithmes  
(a) image d'entrée, (b) EnFCM, (c) FCM standard, (d) sFCM, (e) RFCM.

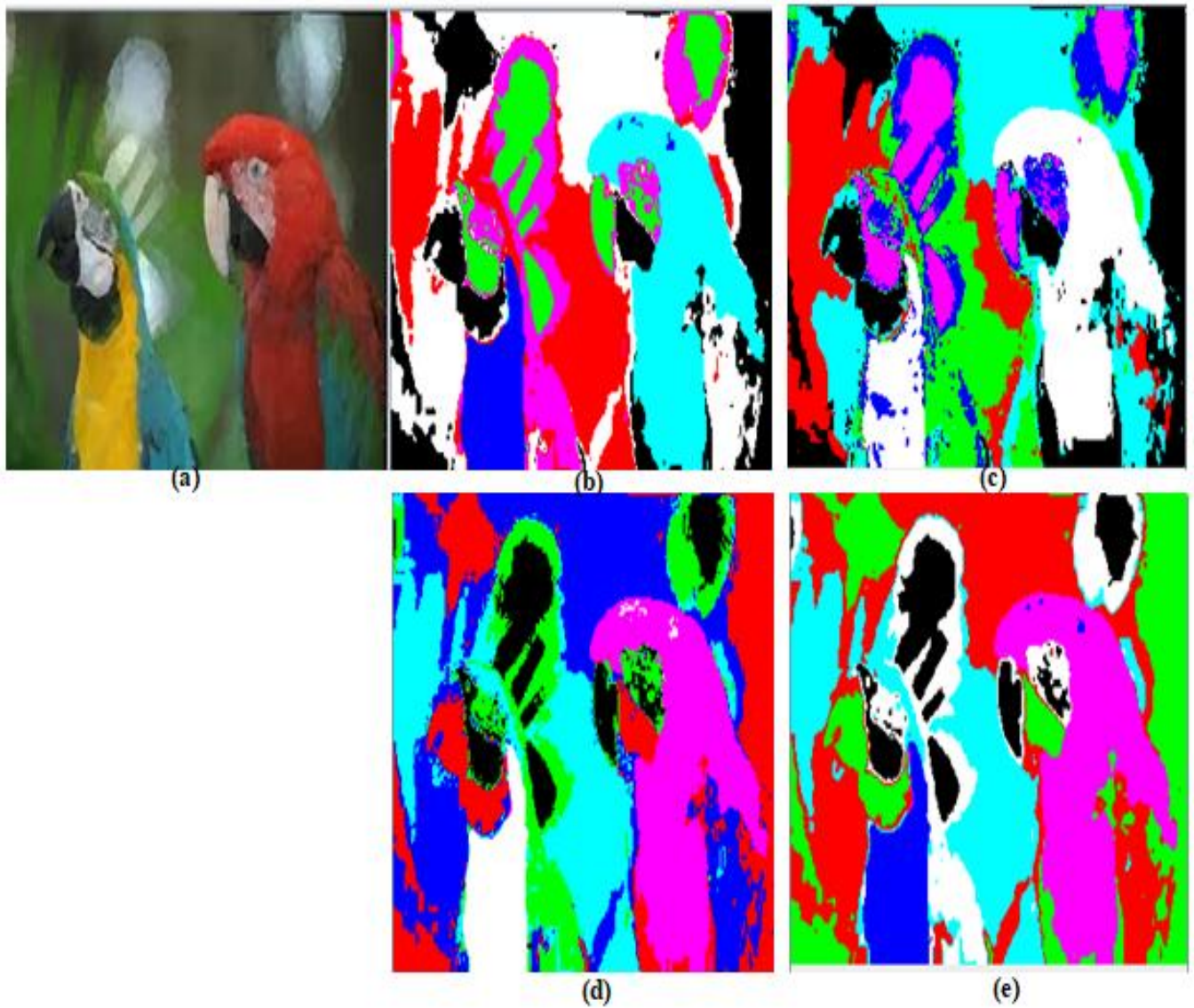




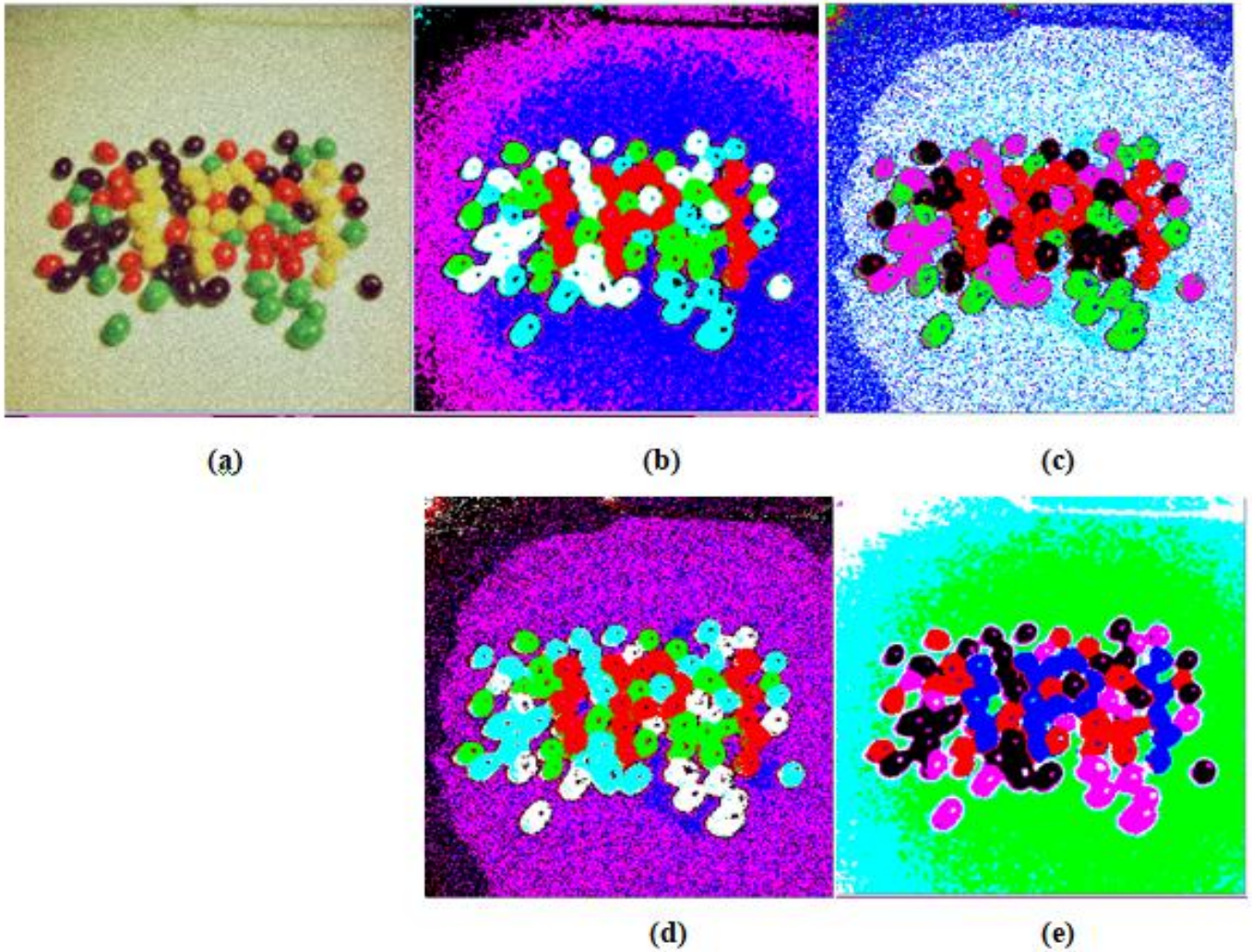
**Figure IV-4 :** Résultat de classification des différents algorithmes  
(a) image d'entrée, (b) EnFCM, (c) FCM standard, (d) sFCM, (e) RFCM.



**Figure IV-5** : Résultat de classification des différents algorithmes  
(a) image d'entrée, (b) EnFCM, (c) FCM standard, (d) sFCM, (e) RFCM.



**Figure IV-6:** Résultat de classification des différents algorithmes  
(a) image d'entrée, (b) EnFCM, (c) FCM standard, (d) sFCM, (e) RFCM.



**Figure IV-7 :** Résultat de classification des différents algorithmes  
(a) image d'entrée, (b) EnFCM, (c) FCM standard, (d) sFCM, (e) RFCM.

#### IV.8. La comparaison entre les algorithmes

Une fois que l'un des boutons de l'algorithme est appliqué à l'interface, l'algorithme choisi s'applique à l'image d'entrée, après la convergence de chaque algorithme on peut passer à l'étape de comparaison, dans cette étape on cherche d'une part à évaluer les partitions floues obtenues par les quatre algorithmes, l'évaluation se focalise sur trois fonctions d'évaluation ( $V_{Pe}$ ,  $V_{Pc}$ ,  $V_{XB}$ ), et nous comparons les algorithmes en terme de temps d'exécution d'une autre part. Les tables IV-1 et IV-2 montrent successivement les résultats des fonctions d'évaluation et le temps de convergence des quatre algorithmes.

Les images	Algorithme utilisée	Les fonctions d'évolution		
		V <sub>PC</sub>	V <sub>PE</sub>	V <sub>XB</sub>
Image Lena	<i>EnFCM</i>	0.291028	0.194263	0.136137
	<i>Standard FCM</i>	0.304141	0.186562	0.560448
	<i>sFCM</i>	0.304142	0.186561	0.186878
	<i>RFCM</i>	<b>0.286024</b>	<b>0.197279</b>	<b>0.131088</b>
Image perroquet	<i>EnFCM</i>	0.321708	0.183405	<b>0.763870</b>
	<i>FCM Standard</i>	0.358027	0.167598	7.311276
	<i>sFCM</i>	0.358023	0.167600	2.436308
	<i>RFCM</i>	<b>0.320563</b>	<b>0.184026</b>	0.849920
Image formes	<i>EnFCM</i>	0.105461	0.278038	0.083284
	<i>FCM Standard</i>	0.127995	0.266964	0.087476
	<i>sFCM</i>	0.127994	0.266965	<b>0.029157</b>
	<i>RFCM</i>	<b>0.049339</b>	<b>0.311733</b>	0.489580
Image boules	<i>EnFCM</i>	0.176600	0.165570	0.447109
	<i>FCM Standard</i>	0.192212	0.158959	1.198710
	<i>sFCM</i>	0.192210	0.158960	0.609697
	<i>RFCM</i>	<b>0.175257</b>	<b>0.159008</b>	<b>0.299081</b>
Image boules 2	<i>EnFCM</i>	0.243611	0.015767	<b>0.001819</b>
	<i>FCM Standard</i>	0.234108	0.031061	5.064416
	<i>sFCM</i>	0.234108	0.031066	1.267373
	<i>RFCM</i>	<b>0.244083</b>	<b>0.014109</b>	0.002634
Image Meky	<i>EnFCM</i>	0.188861	0.140351	<b>0.058213</b>
	<i>FCM Standard</i>	0.183367	0.148471	1.813458
	<i>sFCM</i>	0.182593	0.149902	0.299583
	<i>RFCM</i>	<b>0.192429</b>	<b>0.132616</b>	0.142040
Image Cameraman	<i>EnFCM</i>	0.184500	0.137230	0.071274
	<i>FCM Standard</i>	0.183841	0.140950	0.252468
	<i>sFCM</i>	0.140955	0.183839	<b>0.063101</b>
	<i>RFCM</i>	<b>0.130923</b>	<b>0.187880</b>	0.733067
Image House	<i>EnFCM</i>	0.136279	0.270692	0.084045
	<i>FCM Standard</i>	0.134993	0.271426	0.214322
	<i>sFCM</i>	0.134993	0.271426	0.071461

	<i>RFCM</i>	<b>0.134326</b>	<b>0.272855</b>	<b>0.048663</b>
--	-------------	-----------------	-----------------	-----------------

**Table IV-1** : Tableau des fonctions d'évaluation ( $V_{PE}$ ,  $V_{PC}$ ,  $V_{XB}$ ) des partitions floues des quatre algorithmes

Les images	Algorithme utilisée	Temps d'exécution (ms)
<b>Image Lena</b>	<i>EnFCM</i>	<b>21894</b>
	<i>FCM Standard</i>	30706
	<i>sFCM</i>	288111
	<i>RFCM</i>	140568
<b>Image perroquet</b>	<i>EnFCM</i>	<b>20515</b>
	<i>FCM Standard</i>	41261
	<i>sFCM</i>	286084
	<i>RFCM</i>	141089
<b>Image boules</b>	<i>EnFCM</i>	74638
	<i>FCM Standard</i>	69933
	<i>sFCM</i>	346442
	<i>RFCM</i>	<b>56213</b>
<b>Image formes</b>	<i>EnFCM</i>	<b>60089</b>
	<i>FCM Standard</i>	60241
	<i>sFCM</i>	236670
	<i>RFCM</i>	170913
<b>Image Cameraman</b>	<i>EnFCM</i>	91351
	<i>FCM Standard</i>	<b>86635</b>
	<i>sFCM</i>	376895
	<i>RFCM</i>	239297
<b>Image Boules 2</b>	<i>EnFCM</i>	104038
	<i>FCM Standard</i>	102292
	<i>sFCM</i>	319552
	<i>RFCM</i>	<b>20500</b>

<b>Image Meky</b>	<i>EnFCM</i>	236954
	<i>FCM Standard</i>	<b>235964</b>
	<i>sFCM</i>	611117
	<i>RFCM</i>	387158
<b>Image House</b>	<i>EnFCM</i>	62042
	<i>FCM Standard</i>	<b>11200</b>
	<i>sFCM</i>	201234
	<i>RFCM</i>	115860

**Table IV-2 :** Le temps de convergence en mille seconde (ms) pour les quatre algorithmes

Grâce aux résultats obtenus pour les fonctions d'évaluation et le temps de convergence de quatre algorithmes (FCM standard, FCMs, EnFCM et RFCM), notre méthode montre une bonne robustesse au bruit surtoût pour les fonctions d'évaluations  $V_{PE}$  et  $V_{PC}$ , et pour le temps de convergence notre algorithme prend un temps supplémentaire pour calculer la nouvelle image, mais il converge plus rapide que l'algorithme FCMs.

#### IV.9. Conclusion

Dans ce dernier chapitre, nous avons parlé de l'environnement utilisé, ainsi que du langage de programmation Java et de l'utilisation de ce dernier uniquement, les noms des bibliothèques utilisées dans notre application avec quelques explications sur l'interface. Ensuite nous présentons les résultats obtenus pour les quatre algorithmes implémentés, et nous avons fait également une étude comparative sur les résultats obtenus en termes de fonctions d'évaluation et temps de convergen

## **Conclusion générale**

Dans ce mémoire, nous avons abordé la problématique de la classification des images avec méthode non supervisée, parmi la multitude de méthode non supervisée pouvant être utilisées, nous avons supposé pour l'algorithme de classification floue FCM spatial (RFCM), cet algorithme augmente largement la robustesse de FCM classique grâce à l'intégration de l'information spatiale dans le processus de classification floue.

L'intégration de l'information spatiale est assurée par l'algorithme RFCM à travers la génération d'une nouvelle image à partir de l'image d'entrée, et il a également défini un nouveau terme de similarité entre chaque point centré dans une fenêtre et l'un de ses voisins, ce terme exprime le degré d'influence de chaque voisin sur la nouvelle valeur (intensité ou couleur) de pixel central.

Les résultats expérimentaux montrent clairement en termes des fonctions d'évaluation utilisées la supériorité de notre algorithme par rapport le FCM standard, FCMs et EnFCM, mais pour le temps de convergence notre algorithme a pris un temps supplémentaire pour générer la nouvelle image, mais il reste plus rapide que FCMs.

Parmi les perspectives de ce travail, la proposition d'une nouvelle méthode pour assurer une initialisation automatique des paramètres d'entrée de l'algorithme RFCM (le nombre de classes, les centres), l'exploitation de l'information non locale peut influencer largement sur les résultats de classification, la dernière perspective de ce travail c'est l'utilisation des autres fonctions pour évaluer les résultats obtenus (accuracy segmentation, ...).



## Références

- [1] MrMELIANI Maamar, Mémoire Master Professionnel. "Segmentation d'image par Coopération région-contours". 03/07/2012 l'université
- [2] B. GOSSELIN, " Application des réseaux de neurones artificielles aux reconnaissances automatique de caractères manuscrits ", Thèse de Doctorat, Faculté Polytechnique de Mons, 1996
- [3] C.Houassine, "segmentation d'images par une approche biomimétique hybride". université université m'Hamed bougera- boumerdes. 2012.
- [4] Guillaume Ryder . Stage de MIM 1 encadré par N. Zlatoff et B. Tellez 30 mai – 27 juillet 2005 Version du 5 septembre 2005, " Reconnaissance d'objets Approche structurale ", [http://utilfr42.free.fr/host/research/Ryder\\_stage\\_LIRIS\\_2005.pdf](http://utilfr42.free.fr/host/research/Ryder_stage_LIRIS_2005.pdf) '
- [5] J.F.Canny. "Finding edges and lines in images" M.I.T, Artif. Intell. Lab, Cambridge, MA, TR-720 -1983.
- [6] Boufenara Amina,"Segmentation d'image IRM cérébrale par Systèmes Multi-Agents" ,2011/2012.
- [7] Sarra BENFRIHA et Asma HAMEL, "Segmentation d'image par Coopération région-contours", 2015/2016.incomplet
- [8] Ayoub Tamim. "Segmentation et classification des images satellitaires : application à la détection des zones d'upwelling côtier marocain et mise en place d'un logiciel de suivi spatiotemporel " . Traitement du signal et de l'image. UNIVERSITE MOHAMMED V FACULTE DES SCIENCES Rabat 2015. Français. <tel-01242495>
- [9] (ACHAT Thinhinane + BENHAMAOUICHE Fatima) ,"Segmentation d'image par classifieurs flous: Application à l'Imagerie par Résonance Magnétique (IRM)" 2015.
- [10] J. Rousselle, " Les contours actifs, une méthode de segmentation. Application à l'imagerie médicale". Thèse de doctorat, Université de François Rabelais de Tours, Soutenue le 9 juillet 2003.
- [11] "<https://stackoverflow.com/questions/30807543/difference-between-segmentation-and-classification>"
- [12] Cocquerez, J. P. and Philipp, S., Analyse d'images et segmentation. Masson, 1995. (Coello, et al., 1995) Coello, C. A. C. et al., Multiobjective design optimization of counterweight balancing of a robot arm using genetic algorithm. Seventh Int. Conf. on Tools with Arti. Intelli. (TAI'95), pp. 20-23, Novembre 5-8, 1995, Herndon (USA).
- [13] "Fuzzy C-Means Clustering" ([https://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/cmeans.html](https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html) )
- [14] <http://dspace.univ-tlemcen.dz/bitstream/112/1045/4/Memoire.pdf>
- [15] gharbi fatima zahra - ben namia halima, "La logique floue pour Classification Des Feuilles de vigne"
- [16] Mr HOUASSINE. 2012 "Segmentation d'images par une approche biomimétiquehybride". Thèse de magister, Université M'hamed Bougaraboumerdes.
- [17] R. DUDA, P. HART, " Pattern Classification and Scene Analysis ". Wiley, New York, 1973.3796-4/1997. IEEE.
- [18] KHELIFA MOHAMED AMINE "Optimisation de la Classification Floue d'image".Mimoire master 2018
- [19] <http://www.tsi.enst.fr/pages/enseignement/ressources/mti/gric/schedule.htm>

## Références

- [20] D. L. Pham, "Fuzzy clustering with spatial constraints", in Proceedings International Conference on Image Processing : Vol. II, 2002, pp. 65–68.
- [21] L. Szilágyi, Z. Benyo, S. M. Szilágyi, and H. S. Adam, "MR brain image segmentation using an enhanced fuzzy C-means algorithm", Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2003, pp. 17-21.
- [22] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation", Pattern Recognition, 40(3), 2007, pp. 825–838.
- [23] K.S. Chuang, H.L. Tzeng, S.W. Chen, J. Wu, T.J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation", Computerized medical imaging and graphics, 30(1) , 2006, pp. 9–15.
- [24] Ouarda ASSAS Classification floue des images 14/12/2013 (<http://digitallibrary.univ-batna.dz:8080/jspui/bitstream/123456789/408/1/Classification-floue-des-images.pdf>)
- [25] <http://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatisees/reseaux-de-neurones-automatisees.htm#.XRAMiui6bIU>
- [26] Moulay Abderrahim Ennaji. " analyse et conception d'un reseau de neurones formels pour le filtrage d'un signal dynamique", septembre 1992
- [27] J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57
- [28] J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York
- [29] S . P . Bradley ,U.M. Fayyad ,and C. Reina . Scaling clustering algorithms to large databases. In knowledge Discovery and Data Mining ,pages 9-15,1998.
- [30] Mounzer BOUBOU : "contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégations d'opinion", thèse de doctorat, université Claud Bernard – Lyon1, 2007.
- [31] S.K. Adhikari, J. K. Sing, D. K. Basu, M. Nasipuri, "Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images ", Applied Soft Computing, (34), pp.758-769, 2015.
- [32] X. Xie, G. Beni, "A validity measure for fuzzy clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(8), 1991, pp. 841–847.
- [33] C. Qiu, J. Xiao, L. Yu, L. Han and M. N. Iqbal, "A modified interval type-2 fuzzy C-means algorithm with application in MR image segmentation", Pattern Recognition Letters, vol.34, pp. 1329-1338, 2013.
- [34] M. Melliani, segmentation d'image par cooperation regions-contours, magistère dddd en informatique, ecole national supérieur d'informatique, 2012.