

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE

## **Mémoire de Fin d'Etudes de MASTER ACADEMIQUE**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Système Informatique**

*Présenté par*

**Lilia NEGAB  
Mebarka ZAHER**

Thème

## **Réseaux de neurones pour l'interprétation des images médicales cérébrales**

*Mémoire soutenu publiquement le 18/07/2016 devant le jury composé de :*

**Président : Mr Mohammed Said HABET**

**Encadreur : M<sup>elle</sup> Yasmine YESLI**

**Examinatrice: M<sup>elle</sup> Chafia CHERIFI**

**Examineur: Mr Rabah REHMANI**

# Remerciements

*Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.*

*En second lieu Nous tenons à exprimer notre profonde gratitude et nous remercions à notre promotrice Mademoiselle YESLLI YASMINE, Enseignante au département informatique, Faculté de Génie électrique, U.M.M.T.O, pour le sujet qu'elle nous a proposé ainsi que pour tous les efforts qu'elle a consentis tous au long de ce travail. Ses encouragements, ses précieux conseils, la confiance qu'elle nous a toujours témoignée nous ont été d'une grande aide et pour tous ce qu'on a appris d'elle.*

*Puis tout notre respect et nos remerciements vont vers les membres du jury qui vont pleinement consacrer leur temps et leur attention afin d'évaluer notre travail, qui espérons le sera à la hauteur de leur attente.*

*Et enfin nous remercions également tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.*

# *Dédicace*

*Je dédie ce modeste travail :*

*A ma chère maman et mon cher papa que  
J'adore pour leur soutien et leur dévouement  
tout au long de mes études.*

*A mes chers adorables grands-parents.*

*A mes chers frères que j'aime énormément;*

*A ma chère adorable sœur;*

*A ma binôme Mebarka;*

*A tous mes amis (es).*

*Lília*

# *Dédicace*

*Je dédie ce modeste travail :*

*A ma chère maman et mon cher papa que  
J'adore pour leur soutien et leur dévouement  
tout au long de mes études.*

*A mes chers adorables beaux parents.*

*A mon cher frère que j'aime énormément;*

*A mes chères adorables sœurs;*

*A mon très cher mari pour son soutien  
Sans faille et ses encouragements.*

*A ma binôme Lília;*

*Mebarka*

# *Résumé :*

L'objectif de ce travail est de réaliser un système de traitement d'images médicales cérébrales qui est basé sur la méthode Fuzzy c Means pour l'étape de segmentation et l'application du réseau de neurone de type Perceptron Multicouches(PMC) pour la détection des anomalies dans ces images.

**Mots clés:** Segmentation, Fuzzy C-Means, images médicales, Réseau de neurone, Perceptron Multicouches.

# *Abstract:*

The objective of this work is to achieve a brain medical image processing system that is based on the Fuzzy c Means method for the segmentation step and applying the type of neural network Multilayer Perceptron (MLP) for the anomaly detection in these images.

**Keys words:** Segmentation, Fuzzy C -Means, medical image, neuron network, Multilayer Perceptron.

# Sommaire

Introduction générale .....	1
Chapitre 1 : Approches de Segmentation d'images.....	3
1. Introduction :.....	3
2. Définition :.....	3
3. Différentes approches de la segmentation : .....	3
3.1. Approche contour :.....	4
3.1.1. Méthodes dérivatives : .....	5
3.1.2. Les méthodes variationnelles (les modèles déformables) :.....	5
3.2. Approche région :.....	6
3.2.1. Seuillage :.....	6
3.2.2. Croissance de région: .....	6
3.2.3. La division (Split) : .....	7
3.2.4. La fusion (Merge) : .....	7
3.2.5. Division-Fusion (Split and Merge): .....	8
3.3. Coopération contour-région: .....	9
3.3.1. Coopération séquentielle :.....	9
3.3.2. Coopération mutuelle :.....	10
3.3.3. Coopération des résultats : .....	10
3.4. Approche Classification :.....	11
3.4.1. Classification dure : .....	11
3.4.2. Classification floue : .....	14
4. Conclusion : .....	16
Chapitre 2 : Réseaux de neurones .....	17
1. Introduction :.....	18
2. Définition des réseaux de neurones : .....	18

3.	Topologie des réseaux de neurones :	19
3.1.	Réseau multicouche :	19
3.2.	Réseau à connexions locales:	19
3.3.	Réseau à connexions récurrentes :	20
3.4.	Réseau à connexion complète :	20
4.	Types de réseau de neurone:	20
4.1.	Réseau de neurone non bouclé (Feed_forward) :	20
4.1.1.	Le perceptron monocouche (Perceptron simple) :	21
4.1.2.	Le perceptron multicouche (PMC) :	22
4.1.3.	Les réseaux à fonction radiale (RBF):	23
4.2.	Les réseaux de neurones bouclés (récurrents) :	23
4.2.1.	Les cartes auto-organisatrices de Kohonen (SOM) :	24
4.2.2.	Les réseaux de Hopfield :	25
4.2.3.	Les réseaux ART :	25
4.2.4.	Réseaux de neurones à compétition :	25
5.	Apprentissage :	25
5.1.	Apprentissage supervisé :	26
5.2.	Apprentissage non supervisé :	26
5.3.	Le mode Hybride :	26
5.4.	Les règles d'apprentissage:	26
5.4.1.	Règle de Hebb :	26
5.4.2.	Correction d'erreur :	28
5.4.3.	Apprentissage de Boltzmann :	28
5.4.4.	Apprentissage par compétition :	28
6.	Réseaux de neurones en segmentation d'images :	29
7.	Applications :	30
8.	Conclusion :	30

Chapitre 3 : Imagerie Médicale cérébrale .....	31
1. Introduction.....	32
2. Le système nerveux : .....	32
2.1. Définition : .....	32
2.2. Classification du système nerveux : .....	32
2.2.1. Système nerveux périphérique : .....	33
2.2.2. Système nerveux central : .....	33
3. Imagerie médicale (neuroimagerie) : .....	34
3.1. Définition : .....	34
3.2. Classification des méthodes de neuroimagerie : .....	34
3.2.1. Méthodes morphologiques (anatomiques) : .....	34
3.2.2. Méthodes fonctionnelles : .....	37
4. Conclusion : .....	42
Chapitre 4 : PCM pour l'interprétation des images médicales cérébrales .....	43
1. Introduction : .....	44
2. Fuzzy C-Means : .....	44
2.1. Présentation de l'algorithme FCM : .....	44
2.2. Description des Méthodes de L'algorithme FCM : .....	44
2.3. Diagramme de classes FCM : .....	45
2.4. Description de la Classe « FuzzyCMeansImageClustering » .....	47
3. Perceptron Multicouches : .....	51
3.1. Présentation PCM : .....	51
3.2. Fonctionnement : .....	52
3.2.1. Phase d'apprentissage : .....	52
3.2.1.1. La propagation : .....	52
3.2.1.2. Rétro propagation : .....	53
3.2.2. Phase de test : .....	54



4. Conclusion :	54
Chapitre 5 Implémentation et Expérimentation	55
1. Introduction :	56
2. Synoptique de la démarche préconisée :	56
2.1. Base d'image :	57
2.2. Prétraitement :	57
2.2.1. Niveau de gris :	57
2.2.2. Filtre Médian :	57
2.3. Segmentation :	57
2.4. Extraction des caractéristiques de l'image segmentée:	59
2.5. Classification par le réseau MLP (PMC) :	59
3. Implémentation :	60
3.1. Environnement matériel et logiciel de programmation :	60
3.2. Présentation de quelques vues :	60
3.3. Discussion des résultats :	69
4. Conclusion :	69
Conclusion Générale	70

# *Liste des figures*

Figure 1:schéma représentant les approches de segmentation.....	4
Figure 2:la segmentation par contour. ....	5
Figure 3:Structure quadtree.....	7
Figure 4: Image représente exemple de division/fusion. ....	8
Figure 5: Principe de la coopération séquentielle. ....	9
Figure 6: Principe de la coopération mutuelle. ....	10
Figure 7: Principe de la coopération des résultats. ....	11
Figure 8:Réseau multicouche.....	19
Figure 9:Réseau à connexion local. ....	19
Figure 10:Réseau à connexion récurrentes. ....	20
Figure 11: Réseau à connexion complète. ....	20
Figure 12:Topologie de réseau de neurones non bouclés. ....	21
Figure 13:Schéma général de perceptron simple. ....	22
Figure 14:Exemple de réseau de type perceptron Multicouche.....	23
Figure 15:Topologie de réseau de neurones bouclés. ....	24
Figure 16:Architecture d'une carte auto-organisatrice.....	24
Figure 17: Représentation schématique de la classification fonctionnelle du système nerveux. ....	32
Figure 18: Imager représente l'Encéphale.....	33
Figure 19: une image qui représente la radiographie.....	35
Figure 21: une image qui représente l'IRM de diffusion. ....	37
Figure 20: une image qui représente l'IRM classique.....	37
Figure 22:une image qui représente l'IRM fonctionnelle. ....	38
Figure 23:Le fonctionnement de réseau MLP. ....	52
Figure 24: Evaluation de l'erreur.....	53
Figure 25:Les principales étapes pour l'interprétation d'une image médicale .....	56

Figure 26: le modèle du réseau de neurone proposé.....	60
Figure 27:Fenêtre principale de l'application. ....	61
Figure 28:Information sur le logiciel. ....	61
Figure 29:fenêtre de traitement du logiciel.....	62
Figure 30:Interface de Fuzzy c means. ....	62
Figure 31:Interface de l'extraction de caractéristiques avant la sélection d'une image.....	63
Figure 32:Interface de l'extraction de caractéristiques après la sélection d'une image. ....	63
Figure 33: Interface de régression du réseau de neurone.....	68

# *Liste des tableaux*

Tableau 1:La loi de Hebb.....	27
Tableau 2:Les différents types d'apprentissages.....	29
Tableau 3:les résultats obtenus par FCM.....	58

# Introduction générale

---

Le traitement d'image est né de l'idée de la nécessité de remplacer l'observateur humain par la machine. Ce remplacement était sans fondement expérimental, car nul ne pouvait espérer qu'une machine serait un jour dotée de facultés intellectuelles égale ou même supérieure à celle de l'homme, le champ des applications du traitement d'images est très vaste : la médecine est un des domaines qui n'a pu échapper à ce souffle numérique que ce soit dans la chirurgie, le diagnostic, la pratique thérapeutique etc. le but du traitement des images consiste à saisir une image, extraire les informations intéressantes (relativement aux objets) et enfin interpréter ces informations pour décider d'une action. Pour cela le traitement fait appel à des outils, des algorithmes, qui permettent d'agir sur l'image numérisée ; L'un des processus fondamentaux dans la chaîne de traitement d'image est la segmentation.

La segmentation est un processus qui permet d'engendrer une description compacte de l'image et donc symbolique plus exploitable que l'ensemble des pixels. Les techniques de segmentation d'images est un problème très difficile qui a attiré l'attention de plusieurs chercheurs dans les dernières années. Pour cette raison plusieurs approches de classification et différentes méthodes d'extraction des caractéristiques ont été proposées, mais aucune parmi ces approches n'est considérée comme complète. Parmi les méthodes proposées nous nous intéressons particulièrement aux réseaux de neurones.

Un réseau de neurones est un réseau composé généralement d'un grand nombre d'unités de calcul très simples appelés neurones formels. Ces neurones formels sont interconnectés entre eux, de sorte que les signaux sortant des neurones deviennent des signaux entrant dans d'autres neurones

Dans ce mémoire, nous allons étudier la problématique de la détection de la présence ou non d'anomalie dans une image médicale cérébrale en utilisant un modèle de calcul à base de réseau de neurones.

Dans ce qui suit, nous allons présenter cinq chapitres qui nous permettront de présenter les différents aspects de notre travail.

- **Le premier chapitre** «Approches de Segmentation d'images» nous présentons les différentes méthodes ou les approche de segmentation d'images. Pour chaque technique, nous expliquons le principe théorique.
- **Le deuxième chapitre** « Réseau de neurone » nous définissons d'abord les réseaux de neurones, Après nous donnons les topologies, les types de réseaux existants et les modes d'apprentissage. Ensuite nous l'avons consacré une partie une partie montrant la contribution de réseaux de neurones en segmentation d'images. et enfin nous citons les domaines d'applications des réseaux de neurones.

- **Le troisième chapitre** «Imagerie Médicale cérébrale » nous avons définis le système nerveux ainsi que les images médicales et ces différentes méthodes.
- **Le quatrième chapitre** « PCM pour l'interprétation des images médicales cérébrales» est constitué de deux parties :

La première partie : La reconnaissance des images basée sur une segmentation par une approche classification de type « Fuzzy C Means » pour l'extraction des informations pertinentes nécessaires, La deuxième partie est consacré a la détection d'anomalies dans ces images se fait grâce à un classifieur neuronal de type Perceptron Multicouches.

- **Le cinquième chapitre** « Implémentation et Résultats » ce chapitre nous permettra d'évaluer notre travail à travers une série de tests et établir le bilan de notre projet.

# *Chapitre 1 : Approches de Segmentation d'images*

---

## **1. Introduction :**

La segmentation d'images constitue une étape essentielle en traitement d'image, il existe de très nombreuses approches permettant ce découpage, dont l'efficacité dépend avant tout de l'image.

Dans ce premier chapitre, nous présenterons la notion de segmentation et les différentes approches de segmentation d'images,

## **2. Définition :**

La segmentation est un processus de synthèse, visant à extraire les caractéristiques géométriques des images en faisant abstraction des nuances de couleur, des reflets, des transparences, des ombres, des textures et autres éléments qui constituent l'incroyable richesse de l'information visuelle [1].

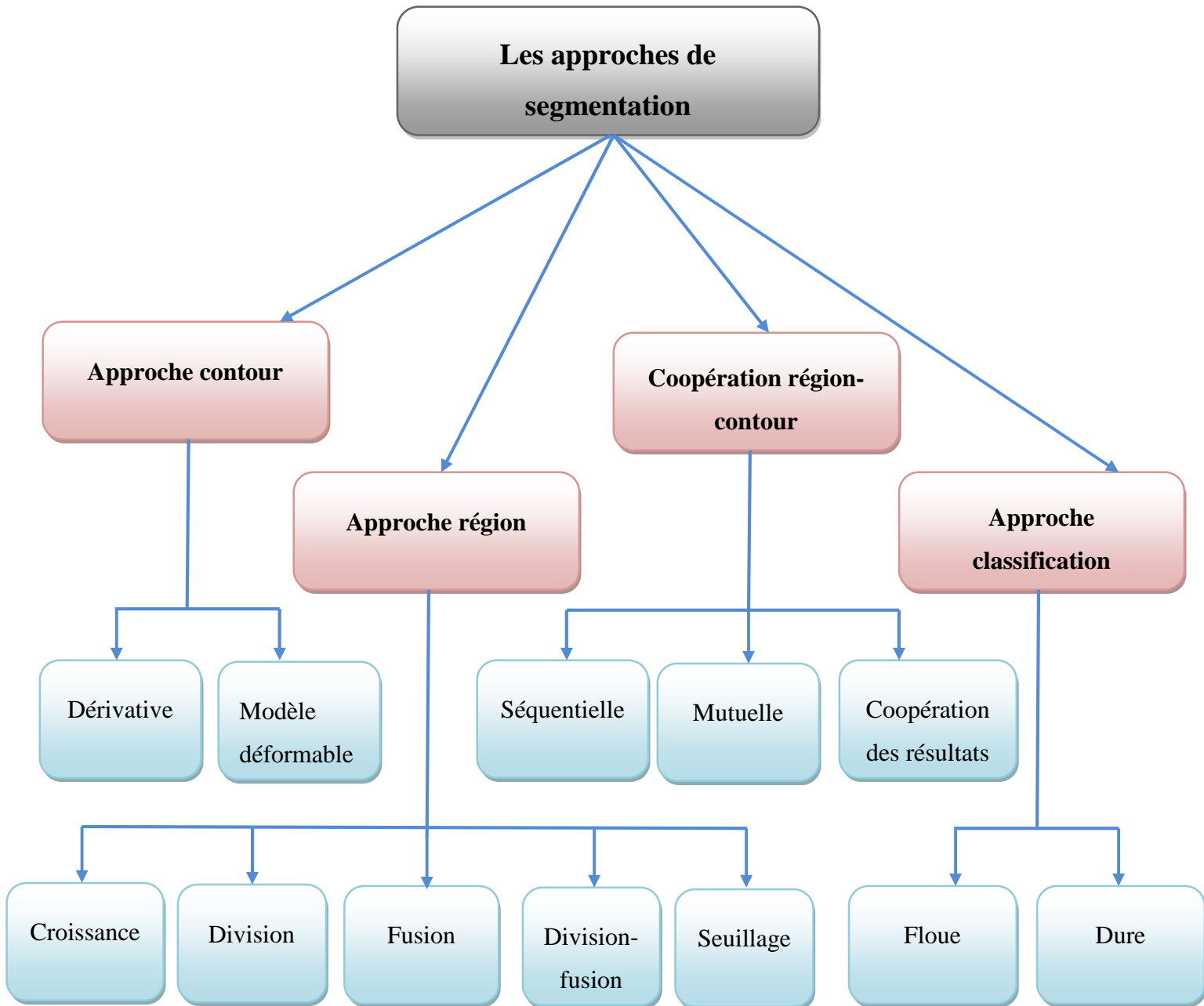
La segmentation est une étape importante pour l'extraction des informations qualitatives de l'image. Elle consiste à découper l'image considérée en régions ayant une homogénéité selon un critère prédéfini (niveau de gris, moments, coefficients d'ondelettes ...).

L'union des régions obtenues doit redonner l'image initiale. La segmentation d'image n'est pas considérée comme un but en soi, mais dépend du type de traitement fixé a priori [W1].

## **3. Différentes approches de la segmentation :**

Une multitude de méthodes de segmentation sont proposées dans la littérature, elles se répartissent en quatre grandes familles : la segmentation par approche contour, région, coopération région-contour et classification [W1].

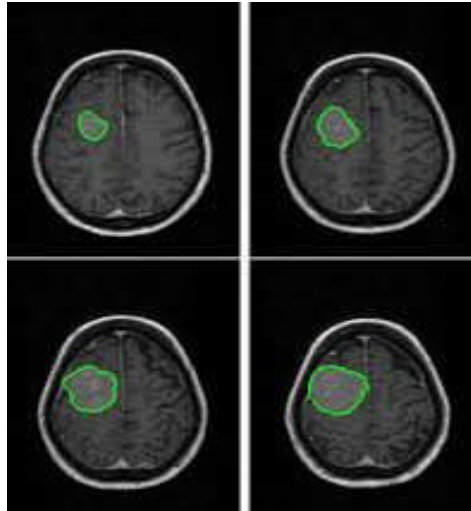




**Figure 1:** schéma représentant les approches de segmentation.

### 3.1. Approche contour :

L'approche contour permet de délimiter les différentes régions par leurs frontières. Les premiers modèles de segmentation cherchent à extraire les contours des objets présents dans l'image. Ils s'appuient sur la détection des changements abrupts de la fonction de luminance ou du niveau de gris [3].



**Figure 2:** la segmentation par contour.

Nous présentons dans ce qui suit les différentes méthodes adaptées pour la détection des contours dans des images en niveaux de gris. Pour ces dernières, deux familles de méthodes sont distinguées :

### 3.1.1. Méthodes dérivatives :

Les méthodes dérivatives sont les plus utilisées pour détecter des transitions d'intensité par différenciation numérique. Ce sont des méthodes locales qui balayent l'image avec un masque définissant la zone d'intérêt. À chaque position, un opérateur est appliqué afin de détecter les transitions significatives au niveau de l'attribut de discontinuité choisi. Le résultat est une image binaire constituée de points de contours et de points non-contours. Pour obtenir des régions homogènes il est nécessaire d'utiliser en aval de ces méthodes des traitements qui ferment les contours ouverts et remplissent les zones délimitées par ces contours. Etant donné que les opérateurs de dérivation sont très sensibles au bruit, les images bruitées doivent être lissées au préalable. Le lissage et la dérivation sont en pratique réunies dans un seul filtre [2].

### 3.1.2. Les méthodes variationnelles (les modèles déformables) :

Les algorithmes de segmentation fondés sur les modèles déformables ont l'avantage, par rapport aux méthodes dérivatives, de fournir des contours ou surfaces fermés. Ces méthodes sont connues sous le nom de contour actif ou *snak*.

Un contour actif est défini comme une courbe minimisant une énergie et évoluant de manière itérative à partir d'une position initiale proche du contour. Cette approche consiste à combiner les deux étapes citées précédemment à savoir l'extraction et le chainage en une seule étape. Le principe des contours actifs est de faire évoluer un contour initial autour de l'objet d'intérêt vers une position d'équilibre, c'est-à-dire une direction des bords de l'objet à détecter. Entre deux positions différentes du contour (deux

itérations), le mouvement des points est fait par une équation impliquant des forces qui agissent sur le contour. Ces forces dépendent des données de l'image (tel que le gradient et l'intensité) et des propriétés du contour (rigidité, élasticité...) [2].

### **3.2. Approche région :**

L'approche duale de la détection des contours pour la décomposition d'une image en ses formes élémentaires est l'approche par régions. Elle repose sur la recherche de zones possédant des attributs communs, soit de luminosité, soit, plus rarement, de textures [3].

#### **3.2.1. Seuillage :**

Une des méthodes les plus simples est la segmentation par seuillage. Cette approche considère que les objets sont uniquement caractérisés par leur intensité. Le choix de deux niveaux de gris est alors mis en œuvre pour isoler la structure. Généralement, des opérations de morphologie mathématique permettent ensuite de raffiner la segmentation : érosion, dilation, ouverture, fermeture, extraction de composante connexe.

Ce type d'approche est parfois suffisant pour la segmentation d'objets fortement contrastés, mais nécessite généralement une interaction avec l'utilisateur. Ces méthodes ne prennent en compte ni l'effet de volume partiel, ni les inhomogénéités d'intensité, ni le bruit [4].

#### **3.2.2. Croissance de région:**

Ce type de segmentation permet de sélectionner un pixel ou un ensemble de pixels de l'image, appelé germe, autour duquel on fait croître une région. Les régions sont construites en ajoutant successivement à chaque germe les pixels qui lui sont connexes et qui vérifient un critère de similarité. La croissance s'arrête lorsque tous les pixels ont été traités.

La littérature en traitement d'images est riche en méthodes de segmentation par croissance de régions.

Trémeau et Borel proposent un algorithme de segmentation qui combine une croissance de régions suivie d'un processus de fusion de régions. Cet algorithme procède par un balayage séquentiel de l'image et considère le premier pixel comme un germe. Il tente alors de faire croître ce germe le plus longtemps possible en y agrégeant les pixels voisins.

L'avantage des méthodes de croissance de région des pixels de départ. Un choix de critère de similarité, aussi qu'un ordre mal adapté selon lequel les pixels voisins sont examinés, peuvent entraîner des phénomènes de sous segmentation ou de sur segmentation [5].

### 3.2.3. La division (Split) :

L'image originale est découpée en régions, elles même redivisées si elles ne sont pas homogènes au sens d'un critère donné, et ainsi de suite. L'algorithme est récursif et se termine lorsque toutes les régions sont homogènes. Nous pouvons citer comme exemple l'algorithme de Olhander qui utilise les pics de l'histogramme de niveaux de gris comme critère de division. Un inconvénient des méthodes de division est qu'elles se basent sur des statistiques globales ; Une petite région située à côté d'une grande sera donc rarement restituée [6].

L'algorithme se résume en les étapes suivantes :

1. Diviser l'image initiale en 4 régions (principe de structure quadtree) où l'on a  $H(\text{images}) = \text{Faux}$ .
2. Si pour une région  $R_i$ , le prédicat d'homogénéité  $H$  n'est pas vérifié ( $H(R_i) = \text{faux}$ ), alors diviser la région en 4.
3. L'algorithme s'arrête lorsque toutes les régions satisfont le prédicat  $H$  [7].

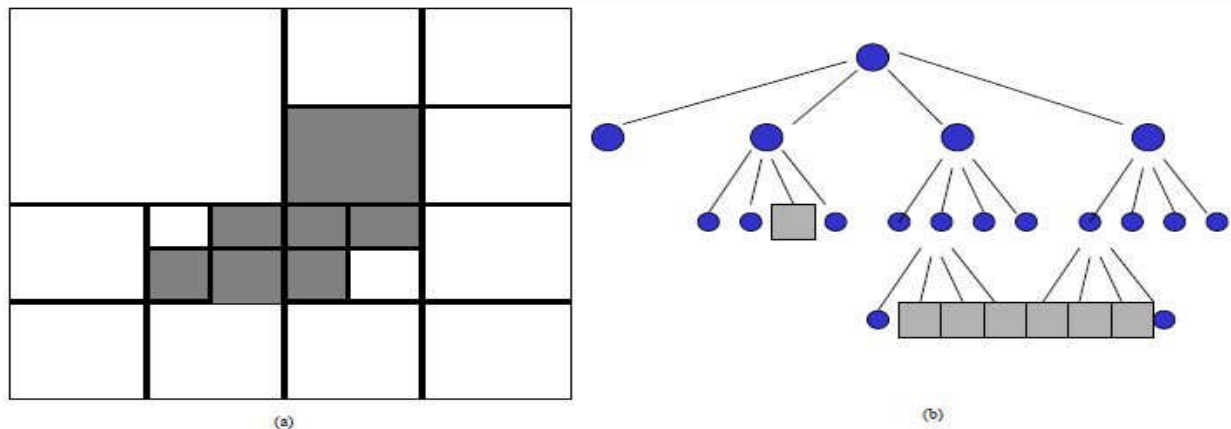


Figure 3: Structure quadtree.

### 3.2.4. La fusion (Merge) :

Cette technique consiste à regrouper des régions si elles sont semblables au sens d'un critère d'homogénéité défini antérieurement et, bien sûr, si elles sont connexes. Ici, les régions initiales sont des

pixels ou des fenêtres carrées appelées sites élémentaires. Ces fenêtres conduisent à une moins bonne résolution, mais améliorent les temps de calcul.

La fusion est très coûteuse en temps de calcul. En effet, elle implique à chaque itération une recherche sur toute l'image des deux régions les plus aptes à fusionner.

Des méthodes permettent de réduire les temps de calcul. Elles consistent à réaliser une recherche de minima locaux et à fusionner les régions correspondantes. Malheureusement, la segmentation dépend alors de l'ordre dans lequel la fusion des différentes régions est réalisée. D'autre part, seules des caractéristiques locales sont utilisées pour autoriser ou interdire la fusion de deux régions [6].

L'algorithme commence par une image sur-segmentée, pour ensuite :

1. Fusionner tout couple de régions adjacentes qui vérifie un critère d'homogénéité.
2. Définir un prédicat Fusionne ( $R_i, R_j$ ) où  $R_i$  et  $R_j$  sont deux régions adjacentes.

Cette méthode est basée sur la labellisation, elle permet d'affecter une même valeur à tous les pixels appartenant à un même objet. Alors qu'elle s'applique sur des images binaires résultant de la division en quadtree [7].

### 3.2.5. Division-Fusion (Split and Merge):

Ces méthodes combinent les deux méthodes décrites précédemment, une phase de division de toutes les régions non homogènes et une phase de fusion de toutes les régions adjacentes de sorte que la région résultante respecte toujours le critère d'homogénéité. Ces méthodes font appel à la théorie des graphes. Ainsi, elles peuvent être classées selon la structure du graphe utilisé [8].



**Figure 4:** Image représente exemple de division/fusion.

### 3.3. Coopération contour-région:

Les approches de segmentation seule présente des inconvénients intrinsèques. Les approches contours travaillent trop localement et ne permettent pas de construire des objets très structurés entre eux à cause de leur incapacité à exploiter les informations globales de l'image. Aussi, ces approches sont très sensibles aux bruits de l'image.

Cependant, les approches régions utilisent des critères souvent très globaux, et les frontières délimitant ces objets ne sont généralement pas significatives.

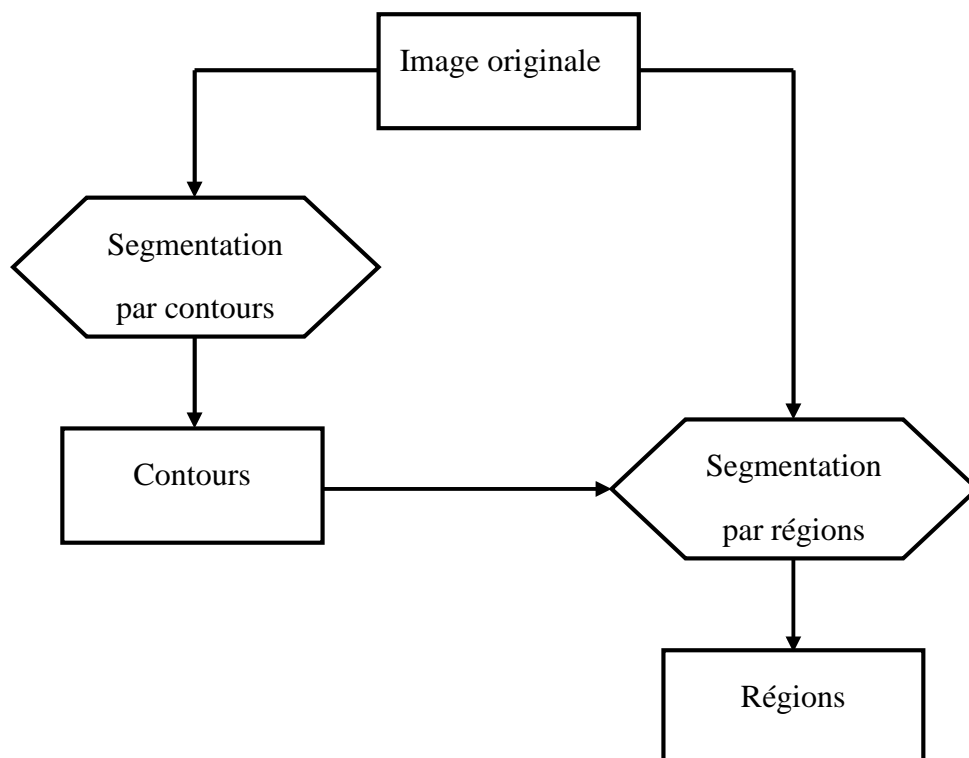
On constate que les points forts de chaque approche peuvent donc permettre de corriger les inconvénients de l'autre. C'est l'idée de base des méthodes de coopérations [9].

Il existe trois formes de coopération contours- régions :

#### 3.3.1. Coopération séquentielle :

Le principe général de la coopération séquentielle est que l'une des techniques, par régions ou par contours, est réalisée en premier lieu. Son résultat sera par la suite exploité par l'autre technique.

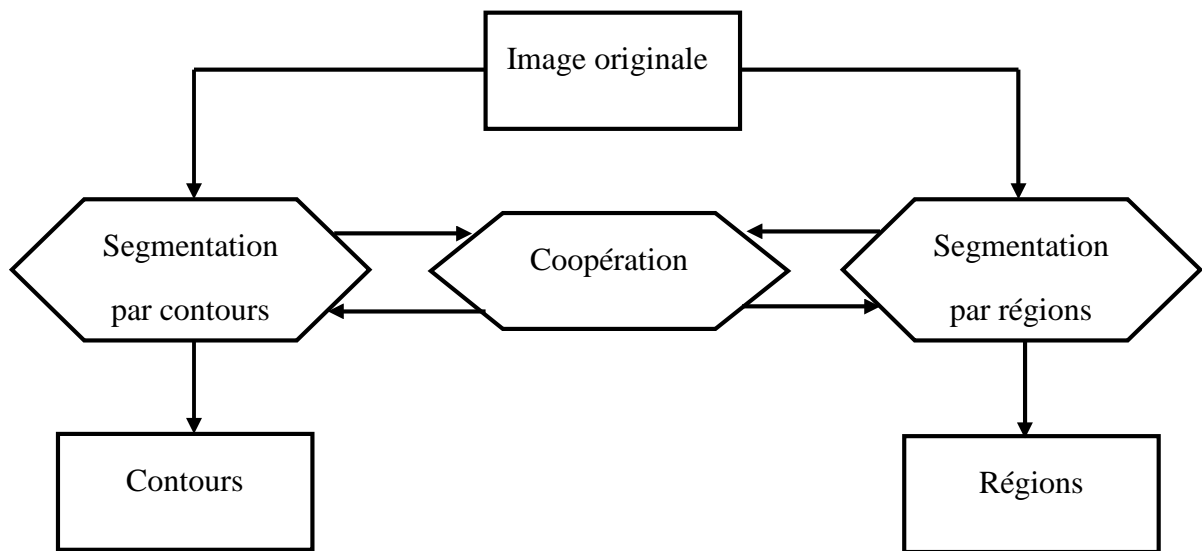
L'intégration de l'information provenant de la segmentation par contours dans une segmentation par régions est l'une des formes de coopération les plus courantes (figure suivante) [10] .



**Figure 5:** Principe de la coopération séquentielle.

### 3.3.2. Coopération mutuelle :

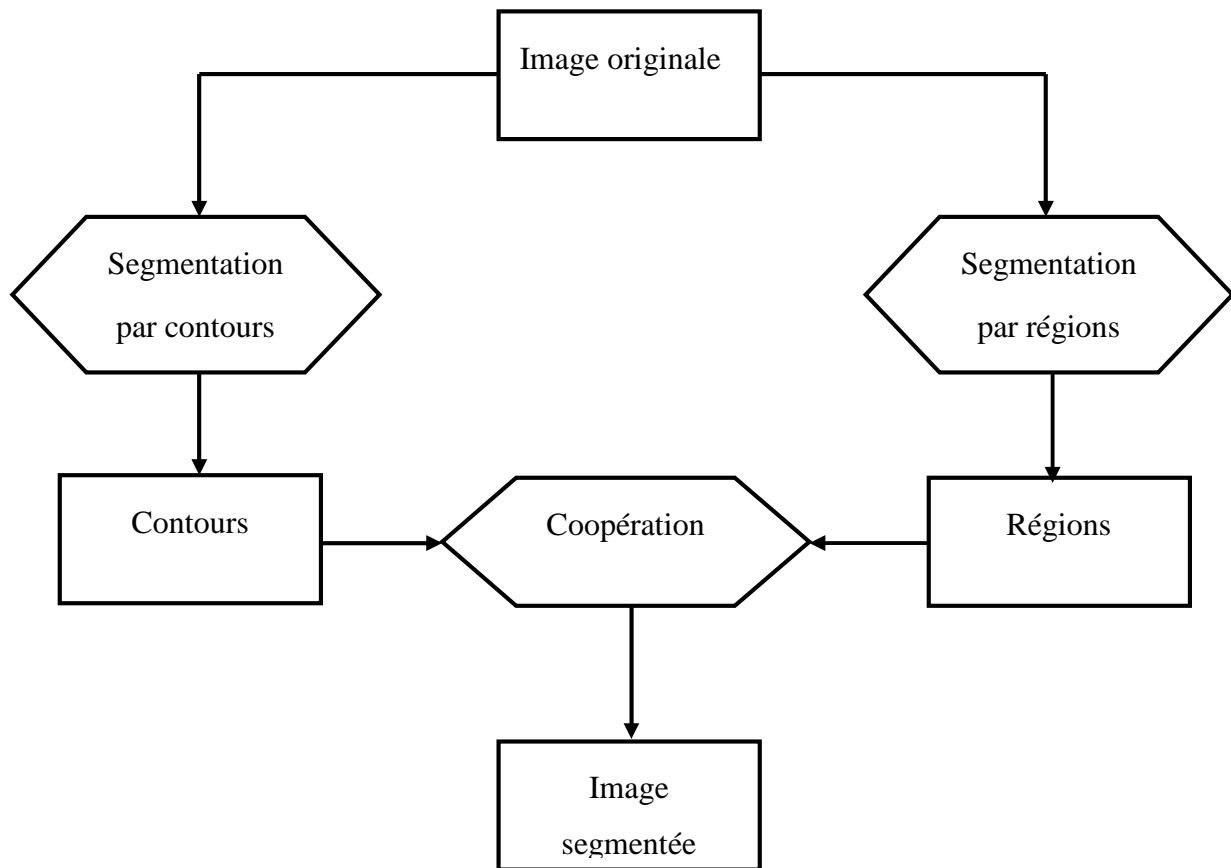
Dans l'approche de coopération mutuelle, les différentes techniques de segmentation sont exécutées en parallèle, tout en échangeant mutuellement des informations (figure suivante). L'information échangée sert à aider la technique de segmentation dans la prise de décision dans le cas de lacunes ou d'informations insuffisantes. La coopération permet de prendre des décisions plus sûres et plus fiables [10].



**Figure 6:** Principe de la coopération mutuelle.

### 3.3.3. Coopération des résultats :

Dans la coopération des résultats, les deux types de segmentations sont réalisés d'une façon parallèle et indépendante, et la coopération sera faite au niveau de leurs résultats respectifs (figure suivante). Ils seront intégrés dans le but d'atteindre une meilleure segmentation que celle obtenue par une seule des techniques. Cette intégration peut être faite sous forme de complémentarité ou de recherche de consensus [10].



**Figure 7:** Principe de la coopération des résultats.

### 3.4. Approche Classification :

Les méthodes de classification permettent de regrouper des objets en groupes ou classes d'objets plus homogènes. Les objets regroupés ont des caractéristiques communes, ils sont similaires mais se distinguent clairement des objets des autres classes. Généralement les résultats des méthodes de classification sont plus ou moins différents [11].

#### 3.4.1. Classification dure :

Dans la classification dure, on trouve deux types de méthodes qu'on va présenter dans ce qui suit :

##### 3.4.1.1. 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 telle ou telle classe.



Nous pouvons donc citer des algorithmes de classification pixels supervisés comme celui de Bayes, Les K plus proches voisins (K-PPV) ou les réseaux de Neurones Multicouches [12].

### **3.4.1.2.Méthodes non supervisées :**

L'intérêt des méthodes non supervisées est qu'elles ne nécessitent aucun apprentissage 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) [12].

Nous pouvons donc citer l'algorithme de classification pixels non-supervisés comme k-means.

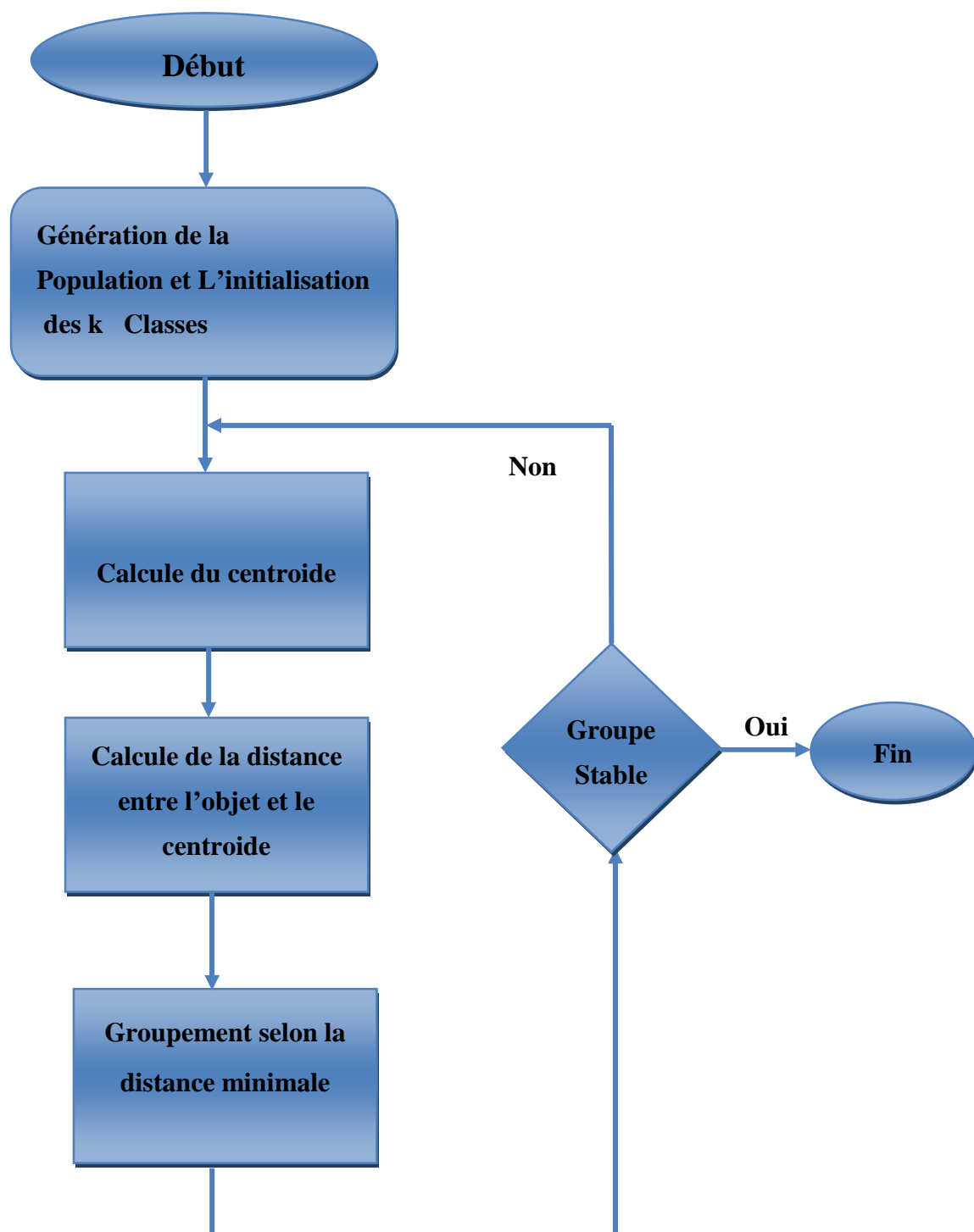
#### **3.4.1.2.1. L'algorithme de K-means :**

L'algorithme K-means créé par MacQueen en 1967 est l'algorithme de Clustering le plus connu et le plus utilisé car il s'avère être très simple à mettre en œuvre et efficace. Il suit une procédure simple de classification d'un ensemble d'objets en un certain nombre K de parties (cluster) fixé a priori. Dans cet algorithme, chaque partie est caractérisée par son centre qui se trouve être la moyenne des éléments composant le cluster [13].

#### **3.4.1.2.2. Les étapes de l'Algorithme de K-means:**

- Choisir au hasard le centre de chacune des K classes.
- Effectuer chaque élément à la classe dont le centre lui est le plus proche.
- Déplacer chaque centre vers la moyenne des éléments de la classe
- Répéter de 2 à 3 jusqu'à convergence [13]

#### **3.4.1.2.3. Organigramme de l'algorithme K-Means :**



### 3.4.2. Classification floue :

Dans une méthode de classification floue, on affecte au pixel un degré d'appartenance pour chacune des classes de la partition qui indique la probabilité que le pixel y appartienne [2]. La classification floue autorise le chevauchement des régions. [W2]

On présente dans ce qui suit un algorithme de classification très populaire, basé sur la logique floue, connu pour son efficacité et sa robustesse.

#### 3.4.2.1. L'algorithme des c-moyens flous (fuzzy c-means) :

C'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 objectif. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une région donnée.

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  $U = [u_{ik}]$  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  $k$ .

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

- $u_{ik} \in [0, 1]$
- $\sum_k u_{ik} = 1$  ; ceci  $\forall i$

L'algorithme du FCM fait évoluer la partition (Matrice  $U$ ) en minimisant la fonction objectif suivante  $J_m(U, C) = \sum_i \sum_k (u_{ik})^m \cdot \|x_i - c_k\|^2$  ;

Où :

$m > 1$  est un paramètre contrôlant le degré de flou (généralement  $m = 2$ ) ;

$c_k$  : le centre de la classe  $k$ .

#### 3.4.2.2. Les étapes de l'algorithme de Fuzzy C- means:

1. Choisir le nombre de classes:  $c$  ;

2. Initialiser la matrice de partition U ainsi que les centres  $c_k$  (initialisation aléatoire) ;

3. Faire évoluer la matrice de partition et les centres suivant les deux équations :

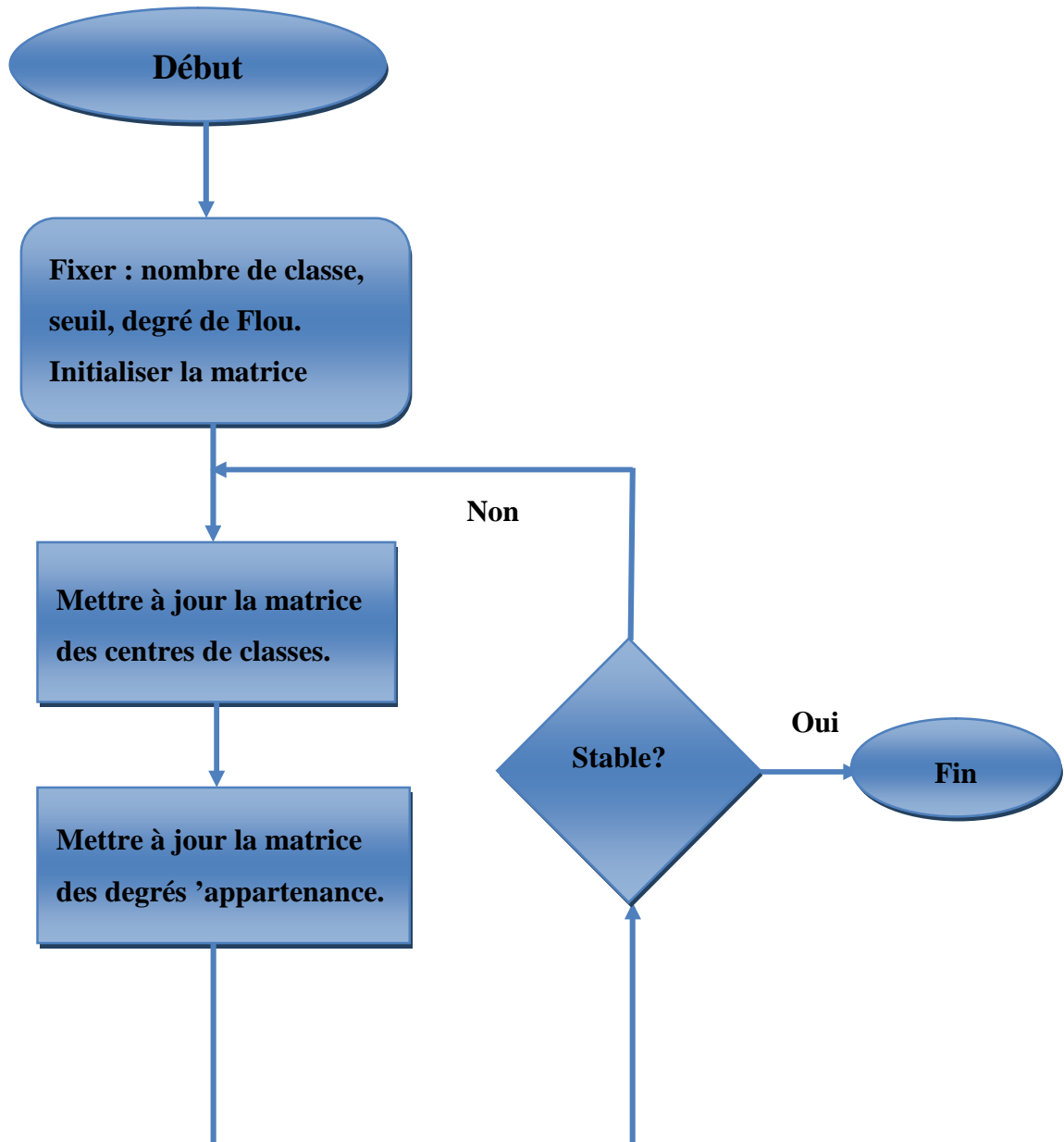
$$(1) u_{ik} = 1 / (\sum_{j=1, c} (d_{ik}/d_{ij})^{2/(m-1)}) ; // \text{mise à jour des degrés d'appartenances.}$$

$$\text{Où : } d_{ij} = \|x_i - c_j\| ;$$

$$(2) c_k = (\sum_i (u_{ik})^m \cdot x_i) / (\sum_i (u_{ik})^m) ; // \text{mise à jour des centres.}$$

4. Test d'arrêt :  $|J^{(t+1)} - J^{(t)}| < \text{seuil}$  ;

### 3.4.2.3. Organigramme de L'algorithme de Fuzzy C-means :



#### **4. Conclusion :**

La segmentation d'image est considérée comme l'étape fondamentale de plusieurs processus d'analyse d'image dédiés à la détection ou l'identification des objets.

Dans ce chapitre, nous avons présenté les différentes approches de segmentation par contours, segmentation par région, segmentation par coopération contour-région qu'elle soit séquentielle, mutuelle ou bien des résultats et segmentation par classification qu'elle soit dure (supervisée et non supervisée) ou floue (L'algorithme de FCM).

## *Chapitre 2 : Réseaux de neurones*

---

## **1. Introduction :**

Les réseaux de neurones sont des modèles théoriques de traitement de l'information inspirés des observations relatives au fonctionnement des neurones biologiques et du cortex cérébral.

Afin d'éclaircir et bien situé notre approche dans le domaine de réseaux de neurones, nous présentons dans ce chapitre, la topologie, les types et le mode d'apprentissage de ces derniers. Nous donnons aussi la contribution de réseaux de neurones en segmentation d'images.

## **2. Définition des réseaux de neurones :**

Tout d'abord, ce que l'on désigne habituellement par réseau de neurones est en fait un réseau de neurones artificiels basé sur un modèle simplifié de neurone. Ce modèle permet certaines fonctions du cerveau, comme la mémorisation associative, l'apprentissage par l'exemple, le travail en parallèle, mais le neurone artificiel est loin de posséder toutes les capacités du neurone biologique. Les réseaux de neurones biologiques sont ainsi beaucoup plus compliqués que les modèles mathématiques et informatiques.

Il n'y a pas de définition universellement acceptée de réseau de neurones. On considère généralement qu'un réseau de neurones est constitué d'un grand ensemble d'unités. (Ou neurones), ayant chacune une petite mémoire locale. Ces unités sont reliées par des canaux de communication (les connexions, aussi appelées synapses d'après le terme biologique correspondant), qui transportent des données numériques. Les unités peuvent uniquement agir sur leurs données locales et sur les entrées qu'elles reçoivent par leurs connexions.

Certains réseaux de neurones sont des modèles de réseaux biologiques, mais d'autres ne le sont pas. Historiquement l'inspiration pour les réseaux de neurones provient cependant de la volonté de créer des systèmes artificiels sophistiqués, voire intelligents, capables d'effectuer des opérations semblables à celles que le cerveau humain effectue de manière routinière, et d'essayer par là d'améliorer la compréhension du cerveau. La plupart des réseaux de neurones ont une certaine capacité d'apprentissage. Cela signifie qu'ils apprennent à partir d'exemples, de même que les enfants apprennent à distinguer les chiens des chats à partir d'exemples de chiens et de chats. Le réseau peut ensuite dans une certaine mesure être capable de généraliser, c'est-à-dire de produire des résultats corrects sur des nouveaux cas qui ne lui avaient pas été présentés au cours de l'apprentissage [14].

### 3. Topologie des réseaux de neurones :

La topologie d'un réseau de neurone est la façon dont les neurones sont reliés (d'une manière générale l'architecture locale du réseau). Il n'existe pas une méthode formelle pour déterminer le nombre de neurone ou de couche ni les connexions nécessaires pour résoudre un problème donnée (sauf le nombre de neurones de la couche d'entrée et de sortie) [15].

Nous présentons les différentes topologies :

#### 3.1. Réseau multicouche :

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avale [16].

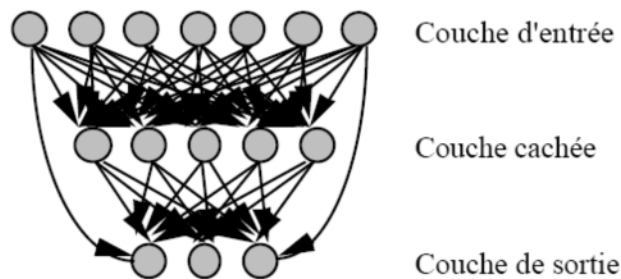


Figure 8: Réseau multicouche.

#### 3.2. Réseau à connexions locales:

Ils ont une structure multicouche. Chaque neurone a des liens avec un nombre réduit et localisé de neurones de la couche aval. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique. C'est un réseau de type « feedforward ». C'est un modèle plus proche du comportement des neurones physiques [W3].

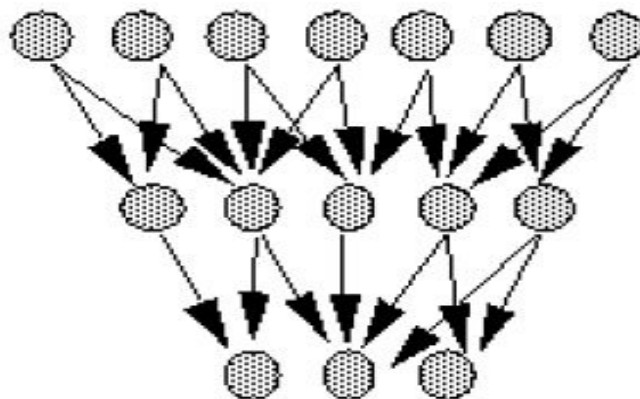


Figure 9: Réseau à connexion local.



### 3.3. Réseau à connexions récurrentes :

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche [16].

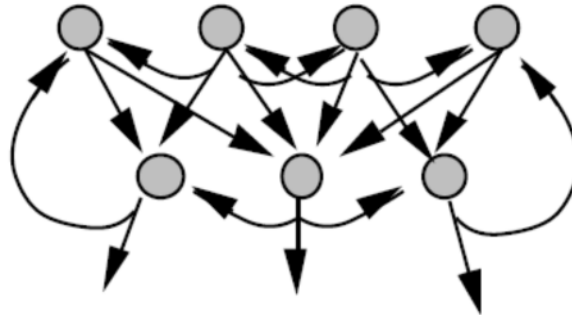


Figure 10: Réseau à connexion récurrentes.

### 3.4. Réseau à connexion complète :

C'est la structure d'inter connexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau [16].

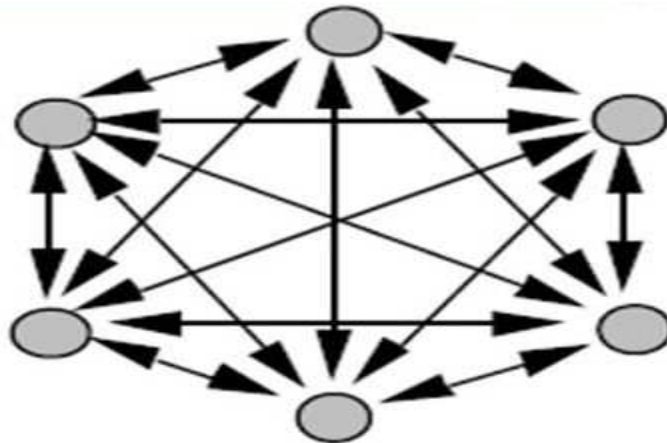


Figure 11: Réseau à connexion complète.

## 4. Types de réseau de neurone:

On distingue deux grandes familles de réseaux de neurones: les réseaux bouclés et les réseaux non bouclés [17]

### 4.1. Réseau de neurone non bouclé (Feed\_forward) :

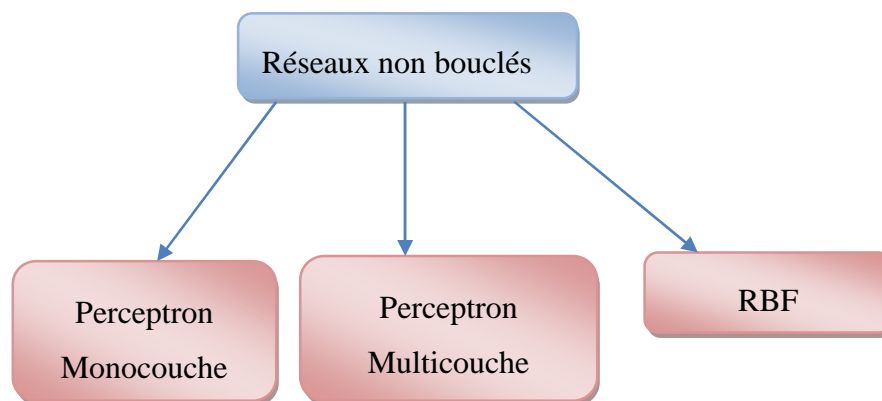
Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées par composition des fonctions réalisées par chacun des neurones le composant. L'entrée au réseau désigne l'information reçue par ce dernier de l'environnement extérieur. Dans un tel réseau, l'information circule

dans une seule direction, de l'entrée vers la sortie. Ce type de réseaux n'autorise pas le retour en arrière de l'information. Le temps ne joue aucun rôle fonctionnel dans un réseau de neurones non bouclé. Pour cette raison, ces réseaux sont aussi qualifiés de réseaux statiques, par opposition aux réseaux dynamiques.

Ce type de réseau comprend deux groupes d'architectures: les réseaux monocouches et les réseaux multicouches. Ils diffèrent par l'existence ou non de neurones intermédiaires appelés neurones cachés entre les unités d'entrées et les unités de sorties appelées nœuds sources ou nœuds d'entrée et nœuds de sortie respectivement.

Un réseau non-bouclé monocouche possède une couche d'entrée recevant les stimuli à traiter par l'intermédiaire des nœuds sources. Cette couche se projette en une couche de sortie composée de neurones (nœuds de calcul) transmettant les résultats du traitement au milieu extérieur. Un réseau non-bouclé multicouche se caractérise par la présence d'une ou de plusieurs couches cachées, dont les nœuds de calcul correspondants s'appellent neurones cachés. Les couches cachées s'interposent entre l'entrée du réseau et sa sortie. Leur rôle est d'effectuer un prétraitement des signaux d'entrée, reçus par la couche d'entrée en provenance du milieu extérieur et de transmettre les résultats correspondant à la couche de sortie où sera déterminée la réponse finale du réseau avant qu'elle soit transmise au milieu extérieur [17].

La Figure suivante représente les réseaux de neurones non bouclés les plus utilisés et Dans ce qui suit, nous allons faire une description de chacun d'eux.

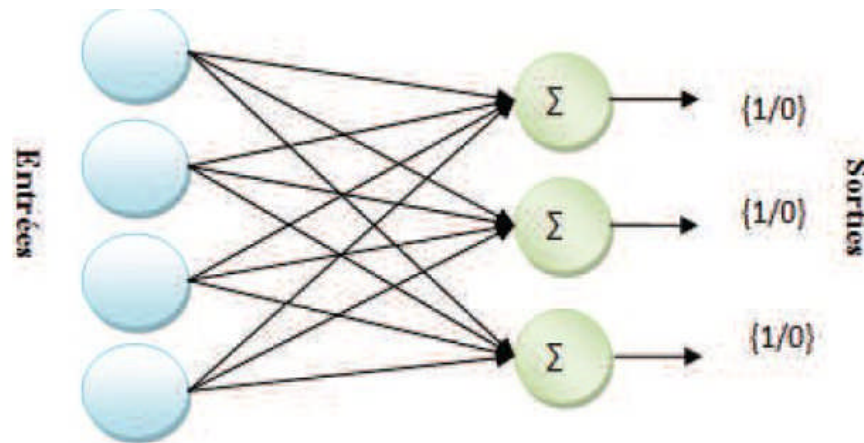


**Figure 12:**Topologie de réseau de neurones non bouclés.

#### 4.1.1. Le perceptron monocouche (Perceptron simple) :

Le perceptron est le premier modèle de réseau de neurones. Le but du perceptron est d'associer des formes en entrée à des réponses (Figure suivante). Le perceptron se compose de deux couches : la couche d'entrée et la couche de sortie qui donne la réponse correspondant à la stimulation présente en entrée. Les cellules de la première couche répondent en oui/non. La réponse «oui» correspond à une valeur «1» et la

réponse « non » correspond à une valeur « 0 » à la sortie du neurone. Les cellules d'entrée sont reliées aux cellules de sortie grâce à des synapses d'intensité variable. L'apprentissage du perceptron s'effectue en modifiant l'intensité de ces synapses. Les cellules de sortie évaluent l'intensité de la stimulation en provenance des cellules de la première couche en effectuant la somme des intensités des cellules actives [12].



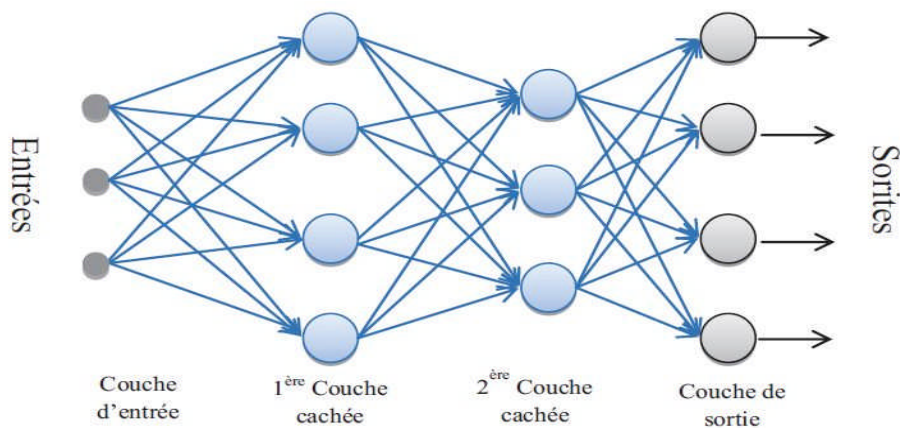
**Figure 13:** Schéma général de perceptron simple.

#### 4.1.2. Le perceptron multicouche (PMC) :

Le perceptron Multicouche est un réseau orienté de neurones artificiels organisé en couches et où l'information voyage dans un seul sens, de la couche d'entrée vers la couche de sortie.

La figure (Fig. Suivante) donne l'exemple d'un réseau contenant une couche d'entrée, deux couches cachées et une couche de sortie. La couche d'entrée représente toujours une couche virtuelle associée aux entrées du système. Elle ne contient aucun neurone. Les couches suivantes sont des couches de neurones. Dans l'exemple illustre, il y a 3 entrées, 4 neurones sur la première couche cachée, trois neurones sur la deuxième couche cachée et quatre neurones sur la couche de sortie. Les sorties des neurones de la dernière couche correspondent toujours aux sorties du système.

Dans le cas général, un perceptron Multicouche peut posséder un nombre de couches quelconque et un nombre de neurones (ou d'entrées) par couche également quelconque [12].



**Figure 14:**Exemple de réseau de type perceptron Multicouche.

#### 4.1.3. Les réseaux à fonction radiale (RBF):

Les réseaux à fonctions de base radiales (RBF) ou plus simplement réseaux à bases radiales ont été proposés par J. Moody et C. Darken. Ils comportant une couche d'entrée, une couche cachée et une couche de sortie.

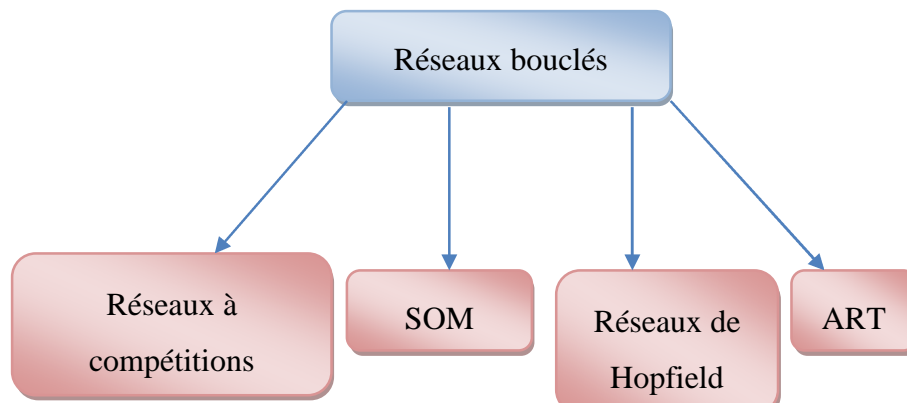
Pour un réseau comportant  $n$  entrées et  $m$  unités cachées, l'activation des neurones cachés est donnée par une fonction de type gaussienne. Chacun de ces neurones ne s'active donc de manière significative que pour des valeurs d'entrée relativement proches des centres des gaussiennes. Les connexions prévenantes des neurones d'entrée ne sont pas pondérées.

L'apprentissage se fait dans ces réseaux par modification des poids des connexions entre les neurones cachés, les neurones de sortie, les centres et les variances des gaussiennes [12].

#### 4.2. Les réseaux de neurones bouclés (récurrents) :

Dans un réseau bouclé, appelé aussi réseau récurrent ou dynamique, il est possible de trouver au moins un cycle où l'information peut revenir à son point de départ. Cela suppose que la sortie d'un neurone peut être fonction d'elle-même. Or cela n'est possible que si la notion de temps est prise en compte. Ce type de réseau est notamment utilisé pour la modélisation dynamique de processus non linéaires [17].

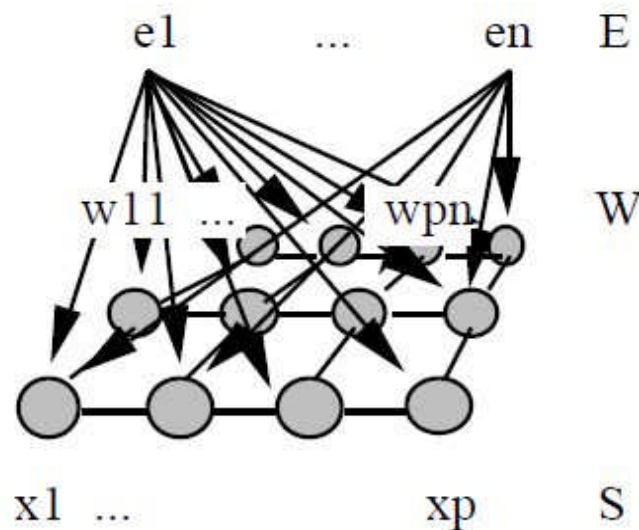
La Figure suivante représente les réseaux de neurones bouclés les plus utilisés et Dans ce qui suit, nous allons faire une description de chacun d'eux.



**Figure 15:** Topologie de réseau de neurones bouclés.

#### 4.2.1. Les cartes auto-organisatrices de Kohonen (SOM) :

Ce modèle de carte auto-organisatrice appartient à la classe des réseaux à compétition. Les neurones de la couche de sortie entrent en compétition, de telle façon qu'habituellement, un seul neurone de sortie est activé pour une entrée donnée. Cette compétition entre les neurones est réalisée grâce à des connexions latérales inhibitrices. Ces cartes s'organisent par rapport aux exemples d'entrée présentés en respectant les contraintes topologiques de l'espace d'entrée. Il y a mise en correspondance de l'espace d'entrée avec l'espace du réseau. Les zones voisines de l'espace d'entrée sont voisines sur la carte auto-organisatrice.



**Figure 16:** Architecture d'une carte auto-organisatrice.

Chaque neurone est connecté à ses 4 plus proches voisins. Ces connexions sont de poids fixes. Tous les neurones sont connectés aux entrées par des connexions plastiques [18].

### 4.2.2. Les réseaux de Hopfield :

Les réseaux de Hopfield sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à chaque autre neurone et il n'y a aucune différenciation entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non-linéaire et sont capables de trouver un objet stocké en fonction de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est l'entrepôt de connaissances mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non-supervisé [W4].

### 4.2.3. Les réseaux ART :

Les réseaux ART ("Adaptative Resonance Theorie") sont des réseaux à apprentissage par compétition. Le problème majeur qui se pose dans ce type de réseaux est le dilemme « stabilité/plasticité ». En effet, dans un apprentissage par compétition, rien ne garantit que les catégories formées vont rester stables. La seule possibilité, pour assurer la stabilité, serait que le coefficient d'apprentissage tende vers zéro, mais le réseau perdrait alors sa plasticité. Les ART ont été conçus spécifiquement pour contourner ce problème. Dans ce genre de réseau, les vecteurs de poids ne seront adaptés que si l'entrée fournie est suffisamment proche, d'un prototype déjà connu par le réseau. On parlera alors de résonnance. A l'inverse, si l'entrée s'éloigne trop des prototypes existants, une nouvelle catégorie va alors se créer, avec pour prototype, l'entrée qui a engendrée sa création. [W4]

### 4.2.4. Réseaux de neurones à compétition :

Un seul neurone de sortie est activé pour une entrée donnée, Mise en correspondance de l'espace d'entrée avec l'espace du réseau [19].

L'ajustement des poids synaptiques, suite à la présentation d'une forme, n'est pas automatiquement réalisé pour tous les neurones constituant le réseau [20].

## 5. Apprentissage :

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement [21].

### 5.1. Apprentissage supervisé :

Dans les quels le système apprend à reconnaître des formes à partir d'un échantillon d'apprentissage qui associe les modalités portées par des variable censées caractériser une forme, et la forme elle-même. sur cet échantillon les résultats correspondant aux divers ensembles d'informations données au système sont connus. C'est à partir de là que le système se paramètre [22].

### 5.2. Apprentissage non supervisé :

Les réseaux à apprentissage non supervisé, sont utilisés lorsque l'utilisateur du réseau n'est pas en mesure de présenter au système un échantillon mettant en regard une somme d'informations et la forme qu'elle est censée représenter. Le réseau s'auto-organise de façon à découvrir des formes récurrentes dans les informations qu'il reçoit, mais il le fait sans aide extérieure, contrairement aux réseaux à apprentissage supervisé. Le plus connu de cette catégorie de réseaux est celui de Kohonen [22].

### 5.3. Le mode Hybride :

Le mode hybride reprend en fait les deux autres approches, puisque une partie des poids va être déterminée par apprentissage supervisé et l'autre partie par apprentissage non-supervisé.

Pour ces trois types d'apprentissage, il y a également un choix traditionnel entre [W5] :

- l'apprentissage << off-line >> : toutes les données sont dans une base d'exemples d'apprentissage qui sont traités simultanément ;
- l'apprentissage << on-line >> : Les exemples sont présentés les uns après les autres au fur et à mesure de leur disponibilité.

### 5.4. Les règles d'apprentissage:

#### 5.4.1. Règle de Hebb :

Elle s'exprime de la façon suivante :

"Si 2 cellules sont activées en même temps alors la force de la connexion augmente".

La modification de poids dépend de la coactivation des neurones présynaptique et post synaptique, ainsi que le montre la table 1.  $x_i$  et  $x_j$  sont respectivement les valeurs d'activation des neurones  $i$  et  $j$ ,  $\partial w_{ij}$  (dérivée partielle du poids) correspond à la modification de poids réalisée.

$x_i$	$x_j$	$\partial w_{ij}$
0	0	0
0	1	0
1	0	0
1	1	+

**Tableau 1:**La loi de Hebb.

La loi de Hebb peut être modélisée par les équations suivantes ( $w(t+1)$  est le nouveau poids,  $w_{ij}(t)$  l'ancien) :

$$w_{ij}(t+1) = w_{ij}(t) + \partial w_{ij}(t)$$

$\partial w_{ij}(t) = x_i \cdot x_j$  (la coactivité est modélisée comme le produit des deux valeurs d'activation)

L'algorithme d'apprentissage modifie de façon itérative (petit à petit) les poids pour adapter la réponse obtenue à la réponse désirée. Il s'agit en fait de modifier les poids lorsqu'il y a erreur seulement.

- 1/ Initialisation des poids et du seuil  $S$  à des valeurs (petites) choisies au hasard.
- 2/ Présentation d'une entrée  $E_l = (e_1, \dots, e_n)$  de la base d'apprentissage.
- 3/ Calcul de la sortie obtenue  $x$  pour cette entrée :

$$a = \sum (w_i \cdot e_i) - S \text{ (la valeur de seuil est introduite ici dans le calcul de la somme pondérée)}$$

$x = \text{signe}(a)$  ( si  $a > 0$  alors  $x = +1$  sinon  $a \leq 0$  alors  $x = -1$  )

- 4/ Si la sortie  $x$  est différente de la sortie désirée  $d_l$  pour cet exemple d'entrée  $E_l$  alors modification des poids ( $\mu$  est une constante positive, qui spécifie le pas de modification des poids) :

$$w_{ij}(t+1) = w_{ij}(t) + \mu \cdot (x_i \cdot x_j)$$

- 5/ Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2 [18].



### 5.4.2. Correction d'erreur :

Cette règle s'inscrit dans le paradigme d'apprentissage supervisé, c'est -à-dire dans le cas où l'on fournit au réseau une entrée et la sortie correspondante. Si on considère  $y$  comme étant la sortie calculée par le réseau, et  $d$  la sortie désirée, le principe de cette règle est d'utiliser l'erreur ( $d-y$ ), afin de modifier les connexions et de diminuer ainsi l'erreur globale du système. Le réseau va donc s'adapter jusqu'à ce que  $y$  soit égal à  $d$  [W6].

### 5.4.3. Apprentissage de Boltzmann :

Les réseaux de Boltzmann sont des réseaux symétriques récurrents. Ils possèdent deux sous-groupes de cellules, le premier étant relié à l'environnement (cellules dites visibles) et le second ne l'étant pas (cellules dites cachées). Cette règle d'apprentissage est de type stochastique ( qui relève partiellement du hasard) et elle consiste à ajuster les poids des connexions, de telle sorte que l'état des cellules visibles satisfasse une distribution probabiliste souhaitée [W6].

### 5.4.4. Apprentissage par compétition :

La particularité de cette règle, c'est qu'ici l'apprentissage ne concerne qu'un seul neurone.

Le principe de cet apprentissage est de regrouper les données en catégories. Les neurones similaires vont donc être rangés dans une même classe, en se basant sur les corrélations des données, et seront représentés par un seul neurone.

Dans un réseau à compétition simple, chaque neurone de sortie est connecté aux neurones de la couche d'entrée, aux autres cellules de la couche de sortie (connexions inhibitrices) et à elle-même (connexion excitatrice). La sortie va donc dépendre de la compétition entre les connexions inhibitrices et excitatrices [W3].

Voici un tableau récapitulatif des différents types d'apprentissages:

Paradigme	Règle d'apprentissage	architecture	tâches
supervisé	Correction d'erreur	Perceptron Simple ou Multi couche	Classification, approximation de fonctions, prédiction, contrôle
	Bolzmman	Récurrente	Classification
	Hebb	Multicouche non bouclés	Analyse de données, classification
	Par compétition	A compétition	Catégorisation au sein d'une classe ; Compression de donnée
			Classification, Catégorisation au sein d'une classe
Non supervisé	Correction d'erreur	Multicouches non bouclé	Analyse de données
	Hebb	Non bouclé ou à compétition	Analyse de données, compression de données
	Par compétition	à compétition	Catégorisation, Compression de données
		Cartes de Kohonen	Catégorisation, Analyse de données
		ART	Catégorisation
Hybride	Correction d'erreur et par compétition	RBF	Classification, Approximation de fonctions, Prédiction, Contrôle

**Tableau 2:** Les différents types d'apprentissages.

## 6. Réseaux de neurones en segmentation d'images :

Classiquement, en segmentation d'images médicales, les réseaux de neurones sont utilisés comme classifieurs. Les poids synaptiques sont déterminés par apprentissage sur une base d'images dont le résultat de segmentation est connu. On parle alors de réseau de neurones supervisé. Souvent, les neurones d'entrées sont les différentes images disponibles et les neurones de sorties nous donnent alors les différentes classes recherches. Il est en autre possible d'introduire des informations a priori en plus des volumes et donc de donner plus de robustesse à cette classification. Mais l'inconvénient majeur de cette méthode par réseaux de neurones est l'étape d'apprentissage qui demande une intervention manuelle pour donner cette vérité terrain dont le réseau a besoin pour calculer les poids synaptique [15].

## 7. Applications :

Se trouvant à l'intersection de différents domaines (informatique, électronique, science cognitive, neurobiologie et même philosophie) , l'étude des réseaux de neurones est une voie prometteuse de l'Intelligence Artificielle, qui a des applications dans de nombreux domaines :

- **Industrie** : contrôle qualité, diagnostic de panne, corrélations entre les données fournies par différents capteurs, analyse de signature ou d'écriture manuscrite...

- **Finance** : prévision et modélisation du marché (cours de monnaies...), sélection d'investissements, attribution de crédits...

- **Télécommunications et informatique** : analyse du signal, élimination du bruit, reconnaissance de formes (bruits, images, paroles), compression de données...

- **Environnement** : évaluation des risques, analyse chimique, prévisions et modélisation météorologiques, gestion des ressources...

- Défense** : guidage de missile, suivi de cible, reconnaissance du visage, radar, sonar, lidar, traitement du signal, compression de données, suppression du bruit.[W7].

## 8. Conclusion :

Les réseaux de neurones sont depuis quelque temps un point de focalisation des médias, du public et des scientifiques. Les travaux menés dans le domaine des sciences de la cognition artificielle ont été marquées par quelques apports non négligeables mais surtout par beaucoup d'optimisme.

Dans ce chapitre nous avons présenté la topologie de réseau de neurones, les types et le mode d'apprentissage.et les différentes applications d'utilisation de ce réseau.

# *Chapitre 3 : Imagerie Médicale cérébrale*

---

## 1. Introduction

L'importance que revêt l'imagerie médicale tient d'abord au fait qu'une image est un concentré d'information bien plus efficace qu'un texte ou qu'une explication verbale. L'interprétation des images médicales est un des domaines de recherche les plus encourageants, étant donné qu'il offre des facilités pour le diagnostic et les décisions thérapeutiques d'un grand nombre de maladies tel que le cancer et en particulier le cancer du cerveau. Le cerveau est le moteur du corps humain, siège de la pensée et des émotions c'est le centre où s'intègrent et où s'élaborent les grandes fonctions motrices, sensibles.

À travers ce chapitre, nous présentons le système nerveux et les différentes méthodes de l'imagerie médicale dont on trouve les méthodes morphologiques ainsi que les méthodes fonctionnelles.

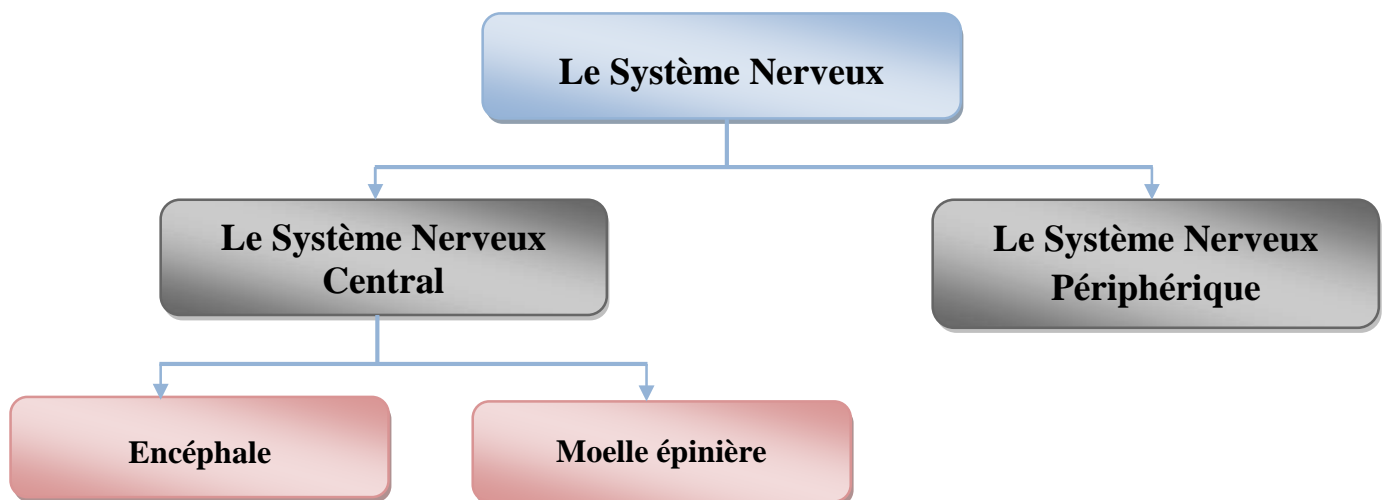
## 2. Le système nerveux :

### 2.1.Définition :

Le système nerveux est un système en réseau formé des organes des sens, des nerfs, du cerveau, de la moelle épinière, etc. Il coordonne les mouvements musculaires, contrôle le fonctionnement des organes, véhicule les informations sensorielles et motrices vers les effecteurs, et, chez les animaux dotés d'un cerveau limbique, régule les émotions, et ceux dotés d'un cerveau cognitif, régule l'intellect [W8].

### 2.2.Classification du système nerveux :

On distingue le système nerveux périphérique, constitué des nerfs crâniens et rachidiens et le système nerveux central, composé de l'encéphale et de la moelle épinière.



**Figure 17:** Représentation schématique de la classification fonctionnelle du système nerveux.

### 2.2.1. Système nerveux périphérique :

Prolongement du système nerveux central, le système nerveux périphérique comprend l'ensemble des nerfs et de leurs renflements (ganglions nerveux). Les nerfs, rattachés par une extrémité au système nerveux central, se ramifient à l'autre extrémité en une multitude de fines branches innervant l'ensemble du corps. Il existe des nerfs crâniens et des nerfs rachidiens. Ces derniers, au nombre de 31 paires, se divisent en une branche postérieure et une branche antérieure. Les branches antérieures peuvent rester indépendantes (nerfs intercostaux) ou s'anastomoser en plexus (brachial, lombaire, sacré) [W9].

### 2.2.2. Système nerveux central :

Encore appelé névraxe, le système nerveux central (S.N.C.) est formé de milliards de neurones (cellules nerveuses) connectés entre eux et d'un tissu de soutien interstitiel (névroglie). Il comprend l'encéphale (cerveau, cervelet, tronc cérébral), protégé par le crâne, et la moelle épinière, long cordon blanchâtre d'environ 40 à 45 centimètres de long enveloppé dans une gaine méningée et logé dans la colonne vertébrale [W9].

#### 2.2.2.1. Encéphale :

L'encéphale est la partie du système nerveux central incluse dans la boîte crânienne (ou crâne). Dans le langage courant, cerveau et encéphale sont deux termes équivalents, mais au sens strict, le cerveau ne correspond qu'à une partie de l'encéphale : les hémisphères cérébraux [W9].



**Figure 18:** Imager représente l'Encéphale.

### 2.2.2.2. Moelle épinière :

La moelle épinière est incluse dans la colonne vertébrale (ou rachis), qui la protège, mais elle n'occupe pas sa longueur totale : elle s'étend de la base du crâne à la première vertèbre lombaire. Il existe une nette segmentation, facilement observable grâce aux 31 paires de nerfs spinaux (ou nerfs rachidiens). À chaque étage de la moelle épinière, la jonction de la racine dorsale et de la racine ventrale forme le tronc nerveux périphérique. Les informations sensibles atteignent la moelle par la racine dorsale. La racine ventrale, formée par les axones des motoneurones, des neurones sympathiques préganglionnaires et des neurones parasympathiques, oriente, à l'inverse, la commande motrice vers les muscles et les viscères [W9].

## 3. Imagerie médicale (neuroimagerie) :

### 3.1. Définition :

L'imagerie médicale est un ensemble de techniques constituant à mettre en image différentes régions ou différents organes de l'organisme. Il existe plusieurs types d'imageries médicales qui sont plus ou moins adaptés en fonction des zones à étudier. On distingue notamment la radiographie, le scanner, l'IRM (Imagerie par résonance magnétique), tomographie, et l'écotrophysiologie.

L'imagerie médicale est certainement l'un des domaines de la médecine qui a le plus progressé ces vingt dernières années. Ces récentes découvertes permettent non seulement un meilleur diagnostic mais offrent aussi de nouveaux espoirs de traitement pour nombreuses maladies [23].

### 3.2. Classification des méthodes de neuroimagerie :

Les méthodes de neuroimagerie d'investigation cérébrale peuvent être anatomiques et fonctionnelles. Chacune de ces méthodes sera brièvement décrite :

#### 3.2.1. Méthodes morphologiques (anatomiques) :

Ces méthodes permettent d'analyser la structure du cerveau normal et pathologique ainsi que la structure osseuse de la boîte crânienne. Elles comprennent la radiographie simple, la tomodensitométrie (scanner X) et l'imagerie par résonance magnétique nucléaire (IRM). Il s'agit de méthodes radiologiques qui utilisent des ondes électromagnétiques (rayons X dans le cas de la radiographie simple et CT, ondes de radiofréquence, RF, dans le cas de l'IRM) [24].

##### 3.2.1.1. Radiographie X :

Se base sur la mesure d'un paramètre biophysique qui est le coefficient d'absorption aux rayons X. elle utilise donc des rayons X (rayonnements électromagnétiques n'appartenant pas au spectre visible)

En raison des leurs propriétés de pénétration de la manière vivante. Suite au passage d'un faisceau de rayons X à travers l'organe exploré (extrémité céphalique et cerveau).on obtient une image volumique de l'organe exploré avec des intensités différentes de l'image sur un film radiographique, de blanc au noir, selon la quantité de rayons X qui n'a pas été absorbée par le segment traversé. Le niveau d'intensité dépend du coefficient d'absorption aux rayons X qui est spécifique a chaque tissu et qui dépend de sa densité en électrons. Plus la densité est importante, plus le coefficient d'absorption est élevé et l'intensité de l'image est importante (hyperdensité, en blanc sur la radiographie). Par rapport a ce principe, les os qui ont une densité élevée en électrons laisseront passer très peu de rayons X ce qui fait que le cliché radiographique n'est pas impressionné (apparaît en blanc). Par contre, les tissus mous ont une densité en électrons moindre et apparaîtront en gris sur la radiographie tandis que les liquides apparaîtront en noir (hypodensité).

Cette technique permet de différencier les tissus mous de l'os, de détecter des fractures de la boîte crânienne ou des processus pathologique cérébraux calcifiés. C'est une technique qui ne permet pas de visualiser des détails de l'organe exploré. Les effets nocifs des rayons X font qu'une radiographie ne peut pas être souvent répétée. Une exposition aux rayons X est susceptible, si elle excède les normes admissibles, de produire des effets nocifs très divers, allant, suivant la dose reçus, de la nausée au cancer [24].



**Figure 19:** une image qui représente la radiographie.

### **3.2.1.2. Tomodensitométrie (computer tomographie, scanner X, CT scan) :**

Il s'agit d'une technique qui a le même principe que la simple radiographie X et qui consiste à faire une reconstruction des tissus à partir d'une analyse tomographique ( coupe par coupe) obtenue en



soumettant l'organe exploré au balayage d'un faisceau de rayons X. Dans le cas d'une exploration cérébrale, l'émetteur tourne autour de la tête de du patient en même temps que les récepteurs chargés de mesurer l'intensité des rayons après leur passage dans le corps. On se base donc sur la mesure du même coefficient d'absorption aux rayons X, comme c'est le cas dans la radiographie. Les données obtenus sont ensuite traitées par ordinateur ce qui permet de recomposer des vues en coupe de l'organe exploré. L'avantage de cette technique est son accessibilité et sa capacité à fournir des images détaillées de l'organe exploré avec une résolution spatiale relativement bonne. Par ailleurs, on peut faire ressortir les vaisseaux sanguins en injectant une substance de contraste généralement iodée qui a la propriété de fortement absorber les rayons X et donc de rendre très visibles les tissus où ce produit est présent [24].

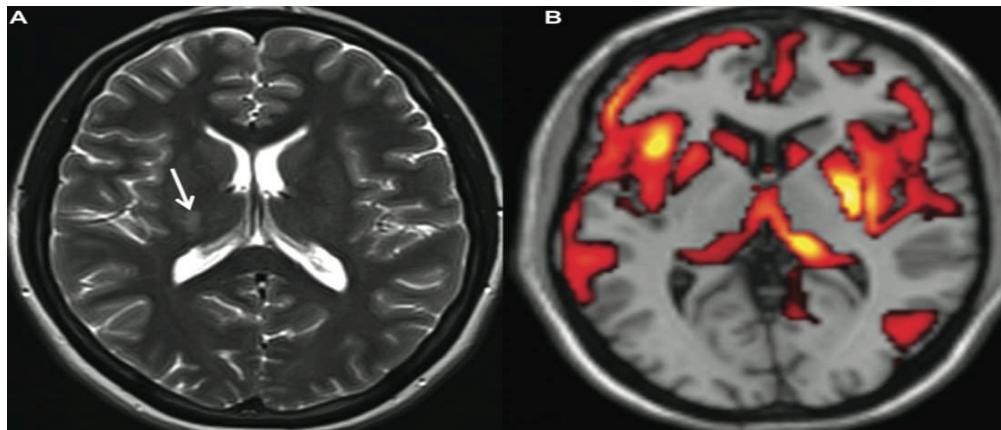
### **3.2.1.3. Imagerie par Résonance Magnétique Nucléaire (IRM) :**

#### **3.2.1.3.1. IRM classique :**

Il s'agit d'une technique de radiologie basée sur le principe de l'imagerie par résonance magnétique nucléaire (RMN) portant sur couplage entre caractéristiques magnétiques (moment magnétique) de certains noyaux atomiques et un champ magnétique externe. Les propriétés des certains noyaux atomiques font qu'en présence d'un champ magnétique ils se comportent comme des petits aimants et ont la tendance à s'aligner dans la direction de ce champ. Très généralement, si on perturbe temporairement cet état d'équilibre (la relaxation) après l'arrêt de la source perturbatrice (RF) sera accompagné de l'émission d'un signal, le signal RMN. Le traitement informatique de ce signal enregistré au niveau de chaque unité du volume exploré (voxel) en présence d'une variation spatiale du champ magnétique principal, nous permet d'obtenir une image de l'organe exploré, d'où appellation d'Imagerie par Résonance Magnétique (IRM).

Dans l'imagerie médicale, on explore ce phénomène pour des noyaux l'hydrogène (protons) présents dans l'eau des tissus biologiques. L'IRM cérébrale montre donc une image de distribution des protons d'eau dans le cerveau. L'IRM fournit des images tridimensionnelles de l'organe exploré, ce qui constitue un avantage important. Les images obtenues ont une excellente résolution spatiale qui est supérieure à celle fournie par le scanner X. Pour effectuer une IRM. On a donc besoin d'un aimant très puissant pour créer le champ magnétique statique, une source d'onde RF (antenne de RF) et un récepteur du signal RMN émis lors de la relaxation des protons (antenne de RF).

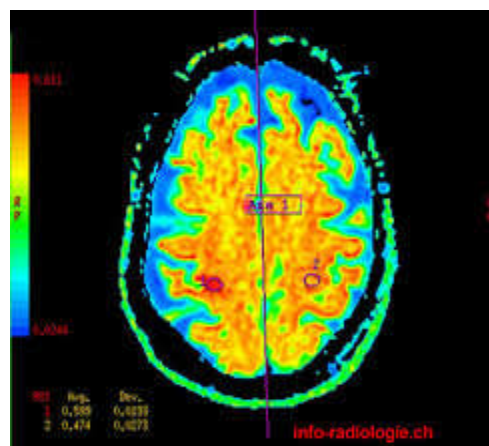
Contrairement au scanner, l'IRM ne présente pas d'inconvénients majeurs pour la santé, le champ magnétique et les ondes RF ne sont pas nocifs pour l'organisme. Ceci signifie qu'un examen IRM peut être effectué d'une manière répétitive, sans danger pour le sujet. Le seul danger réel qui existe est l'impact accidentel avec tout objet ferromagnétique étant donnée la présence du champ magnétique intense [24].



**Figure 20:** une image qui représente l'IRM classique.

### 3.2.1.3.2. IRM de diffusion (méthode récente) :

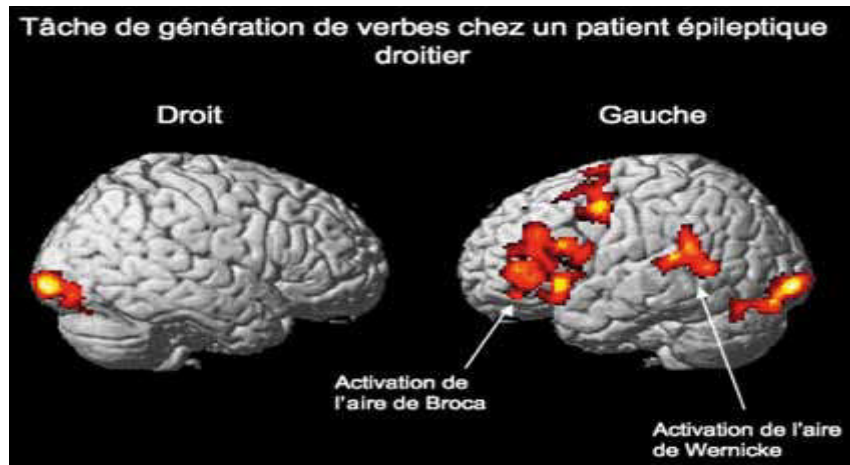
L'IRM de diffusion est une technique basée sur l'IRM et qui utilise des gradients du champ magnétique pour calculer en chaque point de l'image la distribution de probabilité de la diffusion des molécules d'eau. L'architecture des axones en faisceaux parallèles, ainsi que leur couche de myéline, facilite en effet la diffusion des molécules d'eau le long de leur axe. Cette diffusion privilégiée dans une direction est dite « anisotrope ». Cette diffusion étant contrainte par les tissus environnants, cette modalité d'imagerie permet d'obtenir indirectement la position, l'orientation et l'anisotrope des faisceaux de matière blanche du cerveau [24].



**Figure 21:** une image qui représente l'IRM de diffusion.

### 3.2.2. Méthodes fonctionnelles :

Les méthodes fonctionnelles d'imagerie cérébrale explorent le fonctionnement du cerveau. Parmi ces méthodes, l'IRM fonctionnelle a la particularité d'explorer à la fois deux aspects, anatomique et fonctionnel. Selon la manière d'aborder le fonctionnement neuronal [24].



**Figure 22:**une image qui représente l'IRM fonctionnelle.

### 3.2.2.1. Médecine nucléaire :

#### 3.2.2.1.1. Définition :

La médecine nucléaire est une spécialité qui permet le diagnostic, le pronostic et le suivi thérapeutique de nombreuses pathologies.

Elle permet d'étudier le fonctionnement des organes. Les deux types d'examen de la médecine nucléaire en l'administration dans l'organisme de très faibles doses dites (traceuse), d'un médicament radio pharmaceutique (composé d'un produit physiologique et d'un isotope radioactif) des appareils d'imagerie appelés (caméra) détectent les rayonnements émis par l'isotope radioactif, ce qui permet la réalisation de clichés qui seront ensuite analysés.

La médecine nucléaire peut aussi permettre de traiter des pathologies grâce aux médicaments radio pharmaceutiques qui détruisent les cellules (malades).

La tomographie analyse les variations locales du débit sanguin dans un tissu en suivant la diffusion d'une substance radio-active injectée par émission de simples photons (TEMP) et la tomographie par émission de positions (TEP). Cette dernière est l'une des méthodes d'imageries les plus employées au cours des vingt dernières années [23].

#### 3.2.2.1.2. Tomographie par Emission Mono-Photonique (TEMP) :

Dite aussi SPECT (anglais: Single Photon Emission Computed Tomography), la TEMP est une technique tomographique de médecine nucléaire qui permet de réaliser des images et des reconstructions en trois dimensions des organes pour refléter leur métabolisme au moyen d'une gamma caméra (enregistre le photon gamma) qui tourne autour du patient. Donc le patient se voit injecter un produit radioactif qui émettra des photons (rayonnement gamma) qui seront mesurés et quantifiés. Le principe de la TEMP et

assez proche de la TEP sauf que la TEP détecte des couples de photons générés par les collisions positions-électrons. Les traceurs radioactifs TEMP se désintègrent plus lentement que ceux utilisés en TEP, ce qui présente à la fois un avantage (pas besoin de produire les radiosotopes sur place) et un désavantage (temps de demi-vie long, donc difficile de répéter l'examen avant son élimination complète de l'organisme)[24].

### 3.2.2.1.3. Tomographie par Emission de Positons (TEP) :

La TEP (Anglais : Positons Emission Tomography, PET) est une méthode de médecine nucléaire qui permet une évaluation tridimensionnelle de l'activité métabolique de l'organe exploré. Ceci est possible grâce aux émissions produites par les positons issus de la désintégration d'un produit radioactif injecté au préalable. La TEP offre donc la possibilité de suivre la consommation et/ou la distribution de certaines molécules dans nos organes y compris dans le cerveau. Ainsi, on obtient des informations indirectes sur l'activité des neurones. On va injecter une substance spécifique marquée avec un radio-isotope (élément radioactif) qui a la propriété d'émettre des positons. Ainsi, après l'injection le patient sera installé dans la proximité d'une caméra à positons qui va les détecter et permettra de localiser et quantifier les phénomènes enregistrés. Suite à un traitement informatique, les données enregistrées vont être analysées et transformées de façon à obtenir des images correspondant à de multiples coupes de l'organe étudié. Ainsi, si l'on veut voir de quelle manière, et en quelle quantité, le cerveau consomme du glucose, on injecte au patient une molécule analogue, mais marquée au fluor émetteur de positons. Si on veut explorer la distribution du sang dans le cerveau, on va injecter de l'eau radioactive. Une vue d'ensemble du dispositif TEP et une carte typique de distribution du radiotraceur injecté [24].

### 3.2.2.1.4. Comparaison TEMP et TEP :

#### TEP

- De nombreux marqueurs => de nombreuses fonctions.
- Demi-vie réduite => présence d'un cyclotron (accélérateur de particules).
- Très invasif => nombre très limité d'exams par patient.

#### TEMP (SPECT)

- Demi-vie des isotopes (marqueurs) plus longue.
- Résolution spatiale plus faible (7mm) [23].

### 3.2.2.2. IRM fonctionnelle (IRMf) :

L'IRM fonctionnelle (IRMf) permet de suivre quasiment en continu les variations d'activité cérébrale pendant plusieurs dizaines de minutes. Les premières images d'activité cérébrale par IRMf

publiées par John Belliveau et ses collaborateurs ont été produites en utilisant un traceur paramagnétique introduit par voie intraveineuse et permettant d'obtenir des images du volume sanguin cérébral régional. Cette approche était très voisine de la méthodologie utilisée en TEP. Elle en possédait les mêmes limitations, notamment en termes de résolution temporelle. Mais très vite on sut utiliser la déoxyhémoglobine qui présente l'avantage d'être un traceur paramagnétique endogène. Dès 1992, Ken Kwong et ces collaborateurs du même groupe publièrent les premières images d'IRMf utilisant cette méthodologie surnommée BOLD ( Blod Oxygen Level Dependent ). Cette approche n'a cessé depuis de se développer, notamment grâce à la mise au point de techniques d'imagerie IRM ultra-rapide permettant d'obtenir une cartographie BOLD 3D du cerveau en quelques secondes.

L'IRMf se base sur le couplage local entre l'augmentation de l'activité neuronale et la réponse hémodynamique qui signale globalement que le débit sanguin régional augmente dans la proximité d'une population neurale dont l'activité électrique augmente. L'IRMf repose non pas sur la mesure directe des augmentations de débit ou de volume sanguin, mais sur des modifications concomitantes dans les propriétés magnétiques du sang (liées à l'état de l'hémoglobine, oxygénée activités). Ces modifications vont tout simplement faire varier l'intensité du signal RMN puisque lorsqu'on effectue une IRM également une IRM. Cela signifie qu'en IRMf on compare toujours le signal RMN lors des deux conditions, une condition de contrôle et une autre d'activité (tâche). [24]

### **3.2.2.3. Electrophysiologie :**

#### **3.2.2.3.1. Définition :**

L'électrophysiologie est l'étude des phénomènes électriques et électrochimiques qui se produisent dans les cellules ou les tissus des organismes vivants et en particulier dans les neurones et les fibres musculaires.

Contrairement aux autres méthodes d'imagerie fonctionnelle et anatomique du cerveau, la magnétoencéphalographie (MEG) et l'électroencéphalographie (EEG) enregistrent directement l'activité électrique du cerveau. Grâce à leur excellente résolution temporelle, de l'ordre de la milliseconde, elles permettent de suivre en temps réel la chronologie des opérations mentales, et d'étudier la dynamique des phénomènes cérébraux. D'autre part, ces modalités sont entièrement non invasives c'est-à-dire elles détectent des phénomènes issus du cerveau, sans excitation par un rayonnement externe, comme c'est le cas dans les autres méthodes d'imagerie. En effet, le sujet est placé dans un champ magnétique intense en IRM ou un produit radioactif est injecté en TEP [23].

### 3.2.2.3.2. Electroencéphalographie (EEG) et Potentiels Evoqués (PE) :

#### A. Electroencéphalographie (EEG) :

Permet de visualiser l'activité électrique du cerveau, en neurologie la principale application de l'EEG est l'épilepsie, mais elle traite aussi les troubles du sommeil, les déficits sensoriels.... Elle est aussi utilisée en neurosciences cognitives pour étudier les corrélations neuronales de l'activité mentale, depuis les processus moteurs jusqu'au processus complexes de la cognition (attention, mémoire, lecture) [23].

#### B. Potentiels Evoqués (PE) :

Le Potentiel Evoqué(PE) désigne le signal électrique mesuré par EEG, produit par le système nerveux en réponse à une stimulation externe (son, Lumière) ou interne (prise de décision, préparation motrice). Ce signal est très faible (de l'ordre du micro-Volt) et il est nécessaire de répéter l'enregistrement un grand nombre de fois de façon à moyenner toutes ces mesures et à obtenir une caractérisation fiable et robuste du potentiel évoqué. Le PE est diffus, ce qui limite l'information spatiale fournie par cet examen. Il est donc difficile d'identifier avec précision les structures cérébrales qui sont à l'origine de ce signal EEG. par conséquent, l'enregistrement des PE se réalise conjointement avec d'autre technique d'imagerie cérébrale qui présentent une meilleure résolution spatiale.

On distingue des PE exogènes déclenchés par une stimulation sensorielle et des PE endogènes (Cognitifs) liés à une activité cognitive qui n'est pas directement liée à une stimulation extérieure [24].

### 3.2.2.3.3. Magnétoencéphalographie (MEG) :

La MEG mesure les variations des champs magnétiques produits par les courants ioniques principalement générés par les potentiels post-synaptiques des cellules pyramidales du cortex cérébral. Contrairement aux champs électriques enregistrés par EEG, les champs magnétiques ne sont pas sensibles aux variations de conductivité, ce qui explique qu'ils ne sont pas altérés par les différentes couches (liquide céphalo-rachidien, méninges, os, graisse, peau) qu'ils doivent traverser pour gagner la surface du crâne.

Les relations physiques qui unissent les champs électriques aux champs magnétiques sont définies par les équations de Maxwell.

Le champ magnétique produit par un seul potentiel post-synaptique étant trop faible pour être détecté par La MEG, C'est l'activité synchrone des millions de neurones qui est mesurée, de la même manière que pour l'EEG.

Pour détecter des champs magnétiques très faibles, la MEG utilise des senseurs de types SQUID (Superconducting Quantum Interference Device) qui possède des propriétés superconductive (résistance

électrique égale à 0). Un très grand nombre de SQUIDS (de 100 à 306 senseurs) sont disposés autour de la tête dans un casque. Les enregistrements MEG s'effectuent dans une chambre blindée contre les champs électromagnétiques [24].

#### **3.2.2.3.4. Comparaison EEG et MEG :**

L'EEG et la MEG sont des techniques non invasives qui mesurent l'activité électrique et magnétique générée par le cerveau avec une résolution temporelle inégalée. Elles donnent donc des informations directes sur l'activité neuronale en cours. Les principaux atouts de la MEG par rapport à l'EEG sont un nombre plus élevé de senseurs ainsi que l'absence de la sensibilité aux variations de conductivité et à l'anisotropie, ce qui simplifie la résolution du problème inverse et explique la meilleure résolution spatiale de la MEG. De plus, la MEG est plus aisée à utiliser que l'EEG puisqu'elle ne nécessite pas de coller des électrodes. Les principaux désavantages de la MEG, par rapport à l'EEG, sont sa faible accessibilité principalement liée à son coût très élevé (achat du système, consommation importante d'hélium liquide) et le fait qu'elle est principalement sensible à l'activité neuronale du cortex localisée des sillons et non aux sources orientées radialement [23].

### **4. Conclusion :**

Dans ce chapitre nous avons brièvement présenté les méthodes anatomiques et fonctionnelles de neuroimagerie (imagerie médicale). Il s'agit des techniques qui permettent d'observer la morphologie et la fonction du système nerveux et plus particulièrement du cerveau dans des conditions normales et pathologiques. L'imagerie cérébrale est structurelle et fonctionnelle. L'imagerie structurelle cherche à identifier, localiser et mesurer les différentes parties du système nerveux. L'imagerie fonctionnelle cherche à caractériser le cerveau en action.



# *Chapitre 4 : PCM pour l'interprétation des images médicales cérébrales*

---



## 1. Introduction :

Le clustering, ou la classification, est une étape de base du traitement d'une image. Cette opération a pour but de séparer différentes zones homogènes d'une image, afin d'organiser les objets en groupes (clusters) on utilise la méthode de segmentation.

Dans le cadre de notre projet nous mettrons en application L'algorithme de segmentation qui est le Fuzzy C-Means à l'aide de fonctions implémentées en java et implémenté le résultat de cette segmentation dans un réseau de neurones de type Perceptron multicouche (PCM).

## 2. Fuzzy C-Means :

### 2.1.Présentation de l'algorithme FCM :

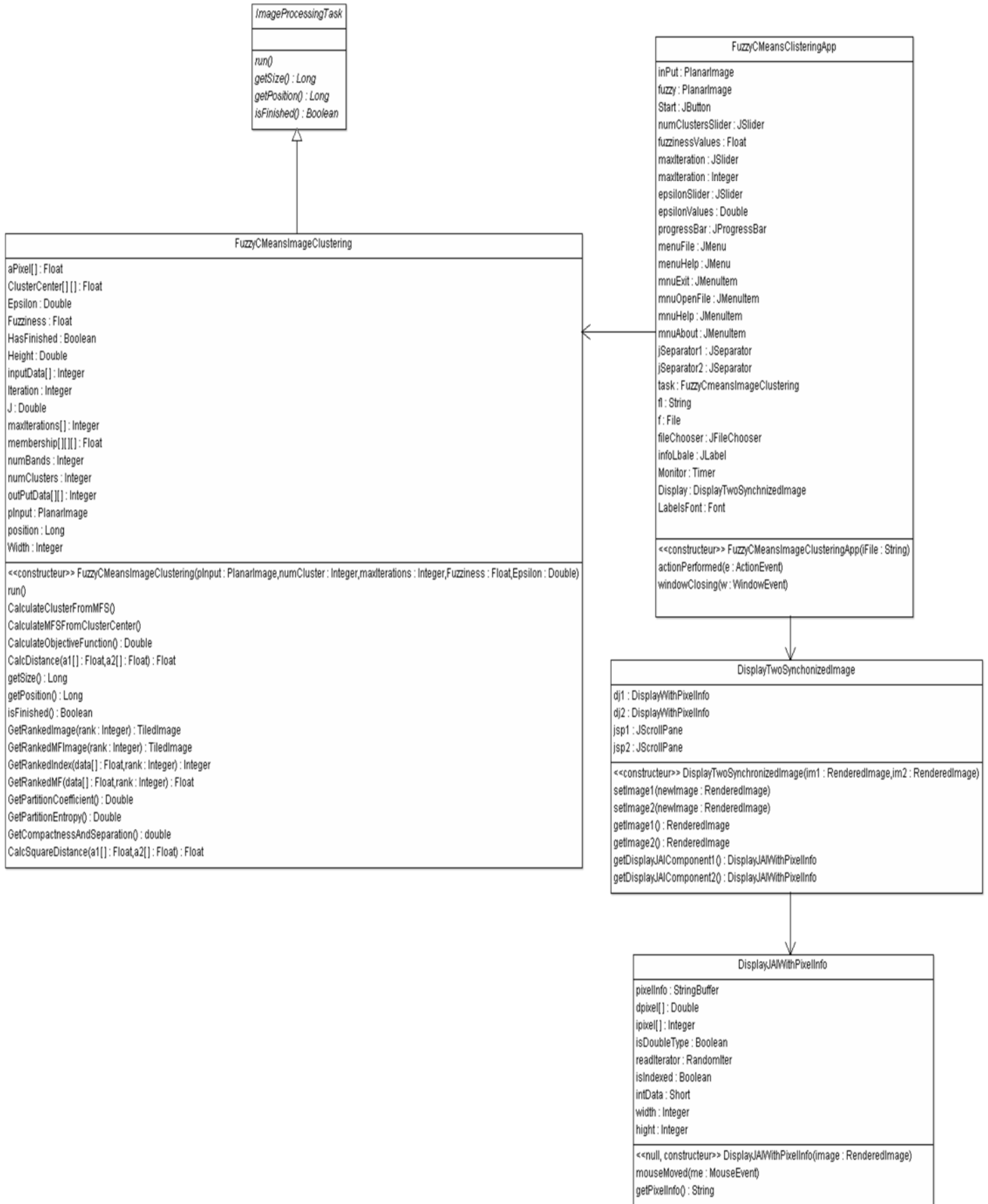
Comme on l'a vu précédemment l'algorithme des c-moyens flous est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité. L'algorithme nécessite de connaître le nombre de classe au préalable et génère les classes par un processus itératif en minimisant la fonction objectif

### 2.2.Description des Méthodes de L'algorithme FCM :

Classes	Description
Classe « ImageProcessingTask »	Cette classe utilise le concept de taille de la tâche de traitement et de la position, si l'on peut estimer la taille de la tâche, une application peut obtenir la position du traitement, à savoir à quel point nous sommes sur la tâche.
Classe « DisplayJAIWithPixelInfo »	Cette classe montre comment on peut étendre la classe DisplayJAI. Nous allons remplacer la méthode mouseMoved de DisplayJAI quand la souris est déplacée, et on stocke des informations sur le pixel sous la souris (mais pas affiché).

Classe « DisplayTwoSynchronizedImages »	Cette classe représente un JPanel qui contient deux instances défilants de DisplayJAIWithPixelInfo, les barres de défilement des deux images sont synchronisées de sorte défilent une image défiler l'autre automatiquement.
<b>Classe « FuzzyCMeansImageClustering »</b>	Cette classe implémente un algorithme de clustering de base Fuzzy C -Means comme une tâche de traitement d'image. Cette mise en œuvre tente d'accélérer les choses, mais a besoin de garder toutes les données d'image sur la mémoire. Cette mise en œuvre ne traite que des données de pixels entiers comme mais peuvent regrouper les données à N dimensions.
Classe « FuzzyCMeansImageClusteringApp »	Cette application affiche une utilisation de l'algorithme FuzzyCMeansImageClustering. L'image originale et en cluster sont présentées, côte à côte et synchronisé. Les données d'entrée sont lues à partir d'un fichier et à la fin de la tâche des différents résultats de clustering commandés peuvent être affichés, ainsi que sauvegardé.

### 2.3.Diagramme de classes FCM :



## 2.4. Description de la Classe « FuzzyCMeansImageClustering »

Cette classe implémente un algorithme de clustering de base Fuzzy C -Means comme une tâche de traitement d'image. Cette mise en œuvre tente d'accélérer les choses, mais a besoin de garder toutes les données d'image sur la mémoire. Cette mise en œuvre ne traite que des données de pixels entiers comme mais peuvent regrouper les données à N dimensions.

### Paramètre:

- pInput : Image en entrée.
- numClusters : le nombre de clusters désiré.
- maxIterations : le nombre maximum d'itérations.
- Fuzziness : le degré de flou.
- epsilon - une petite valeur utilisée pour vérifier si le regroupement a convergé (seuil d'arrêt).

### Méthodes

1. **Run () :**  
cette méthode exécute l'algorithme classique FCM :
  - a) Calcule le centre de cluster.
  - b) Met à jour le degré d'appartenance.
  - c) Réalise des statistiques et répète à partir de « a ».

```
public void run() {  
    double lastJ;  
    lastJ = calculateObjectiveFunction();  
    for(iteration=0;iteration<maxIterations;iteration++) {  
        calculateClusterCentersFromMFs();  
        calculateMFsFromClusterCenters();  
        j = calculateObjectiveFunction();  
        if (Math.abs(lastJ-j) < epsilon) break;  
        lastJ = j;    }  
    hasFinished = true;  
    position = getSize();}
```

2. **getSize()** : Cette méthode renvoie la taille estimée pour cette tâche. La valeur est une approximation, juste pour être en mesure de donner à l'utilisateur un retour sur le temps de traitement. La valeur est calculée comme le nombre de boucles dans la méthode Run.

```
public long getSize()
{
    return (long)maxIterations*
    (
        (numClusters*width*height*(2*numBands))+
        (width*height*numBands*numClusters)+
        (numClusters*width*height*(numBands+numClusters))+
        (numClusters*width*height*(2*numBands))
    )
}
```

3. **getPosition()** : cette méthode retourne une mesure sur la progression de cet Algorithme.

```
public long getPosition()
{
    return position;
}
```

4. **isFinished()** : cette méthode retourne vrai si la classification est terminée.

```
public boolean isFinished()
{
    return (position == getSize());
}
```

5. **GetRankedImage()** : Cette méthode retourne le rang de l'image tel que les pixels sont les centre de classes du n<sup>ème</sup> meilleur choix.

```

public TiledImage getRankedImage(int rank)
{
    SampleModel sampleModel =
    RasterFactory.createBandedSampleModel(DataBuffer.TYPE_INT,width,height,numBands);
    WritableRaster raster =
        RasterFactory.createWritableRaster(sampleModel,new Point(0,0));
    int[] pixelArray = new int[numBands];
    for(int h=0;h<height;h++)
    for(int w=0;w<width;w++)
    {
        int aCluster = getRankedIndex(membership[w][h],rank);
        for(int band=0;band<numBands;band++) pixelArray[band] =
        (int)clusterCenters[aCluster][band];
        raster.setPixel(w,h,pixelArray);
    }
    TiledImage pOutput = new TiledImage(pInput,false);
    pOutput.setData(raster);
    return pOutput;    }

```

6. **GetRankedMFImage()** : cette méthode retourne le degré d'appartenance

```

public TiledImage getRankedMFImage(int rank)
{
    SampleModel sampleModel =
        RasterFactory.createBandedSampleModel(DataBuffer.TYPE_BYTE,
                                                width,height,1);
    ColorModel colorModel = PlanarImage.createColorModel(sampleModel);
    WritableRaster raster =
        RasterFactory.createWritableRaster(sampleModel,new Point(0,0));
    for(int h=0;h<height;h++)
    for(int w=0;w<width;w++)
    {
        int aCluster = (int)(255*getRankedMF(membership[w][h],rank));
        raster.setPixel(w,h,new int[] {aCluster});
    }
}

```

7. **GetPartitionCoefficient()** : cette méthode renvoie la répartition de la mesure de coefficient du cluster.

```
public double getPartitionCoefficient()    {  
    double pc = 0;  
    for(int h=0;h<height;h++)  
        for(int w=0;w<width;w++)  
            for(int c=0;c<numClusters;c++)  
                pc += membership[w][h][c]*membership[w][h][c];  
    pc = pc/(height*width);  
    return pc;    }
```

8. **GetPartitionEntropy()** : cette méthode renvoie la partition entropie du cluster valide.

```
public double getPartitionEntropy()    {  
    double pe = 0;  
    for(int h=0;h<height;h++)  
        for(int w=0;w<width;w++)  
            for(int c=0;c<numClusters;c++)  
                pe += membership[w][h][c]*Math.Log(membership[w][h][c]);  
    pe = -pe/(height*width);    return pe;    }
```

9. **GetCompatnessAndSeparation()** : cette méthode retourne la mesure de compacité et de séparation des cluster valide.

```

public double getCompactnessAndSeparation()    {
    double cs = 0;
    for(int h=0;h<height;h++)
        for(int w=0;w<width;w++)            {
            int index = (h*width+w)*numBands;
            for(int b=0;b<numBands;b++)
                aPixel[b] = inputData[index+b];
            for(int c=0;c<numClusters;c++)    {
                float distancePixelToCluster =
calcSquaredDistance(aPixel,clusterCenters[c]);
                cs += membership[w][h][c]*membership[w][h][c]*
                    distancePixelToCluster*distancePixelToCluster;            }
            cs /= (height*width);
        }
    float minDist = Float.MAX_VALUE;
    for(int c1=0;c1<numClusters-1;c1++)
        for(int c2=c1+1;c2<numClusters;c2++)    {
            float distance = calcSquaredDistance(clusterCenters[c1],clusterCenters[c2]);
            minDist = Math.min(minDist,distance);        }
    cs = cs/(minDist*minDist);    return cs;    }

```

Pour plus de détails sur les autres classes de FCM voir l'annexe.

### 3. Perceptron Multicouches :

#### 3.1.Presentation PCM :

Le Perceptron multicouche est un réseau orienté de neurones artificiels organisés en couches et où l'information ne circule que dans un seul sens : de la couche d'entrée vers la couche de sortie. La couche d'entrée représente toujours une couche virtuelle associée aux entrées du système et ne contient donc aucun neurone. Les couches suivantes, elles, sont constituées de neurones. Les sorties des neurones de la dernière couche correspondent toujours aux sorties du système. Un Perceptron multicouche peut posséder un nombre de couches quelconque et chaque couche peut comporter un nombre de neurones (ou d'entrées, si il s'agit de la couche d'entrée) également quelconque.

Les neurones sont reliés entre eux par des connexions pondérées. Ce sont les poids de ces connexions qui gouvernent le fonctionnement du réseau et programment une application de l'espace des entrées vers l'espace des sorties à l'aide d'une transformation non linéaire. La création d'un Perceptron multicouche pour



résoudre un problème donné passe donc par l'inférence de la meilleure application possible telle que définie par un ensemble de données d'apprentissage constituées de paires de vecteurs d'entrées et de sorties désirées

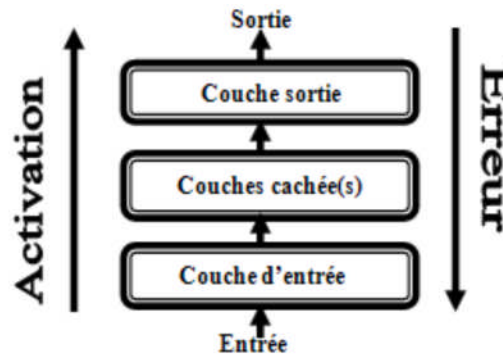


Figure 23: Le fonctionnement de réseau MLP.

### 3.2.Fonctionnement :

Un PMC fonctionne en deux temps dans:

#### 3.2.1. Phase d'apprentissage :

Les PMC utilisent, pour modifier leurs poids, un algorithme d'apprentissage qu'est fondamentale. Elle consiste à faire apprendre au réseau les différentes configurations des vecteurs de paramètres caractéristiques des images de fonds que l'on souhaite classifier Pour cela on utilise :

##### 3.2.1.1. La propagation :

Qui consiste à présenter une configuration d'entrée au réseau, puis à la propager à celle de sortie en passant par les couches cachées. Pour chaque neurone, On calcule :

$$y_i = G \left( \sum_{j=0,d} W_{jk} \phi_j \right) = G(a_k)$$

$W_{jk}$  : poids de la connexion de la cellule j à la cellule k.

$a_k$  : activation de la cellule k.

$g$  : fonction d'activation.

$$g(x) = \begin{cases} X & \text{linéaire} \\ \frac{1}{1 + e^{-R}} & \text{logistic(sigmoid)} \end{cases}$$

### 3.2.1.2. Rétro propagation :

Cet algorithme repose sur la minimisation de l'erreur quadratique entre les sorties calculées et celles souhaitées; L'erreur considérée comme une fonction des poids synaptiques. Cette erreur représente la somme des différences au carré entre les réponses calculées et celles désirées pour tous les exemples contenus dans l'ensemble de l'apprentissage.

#### Principe :

1. Evaluation de l'erreur  $E^j$  (ou  $E$ ) due à chaque connexion :

$$E = \delta_j Z_i = \frac{\partial a_j}{\partial W_{ij}} \frac{\partial E^i}{\partial a_j} = \frac{\partial E^i}{\partial w_{ij}}$$

#### Idée :

Calculer l'erreur sur la connexion  $w_{ji}$  en fonction de l'erreur après la cellule  $j$

Pour les cellules de la couche de sortie.

$$\frac{\partial E^i}{\partial y_k} = g'(a_k) \cdot (u_k(x_i) - y_k) \delta_k = \frac{\partial E^i}{\partial a_k} = g'(a_k)$$

Pour les cellules d'une couche cachée :

$$\delta_j = \frac{\partial E^i}{\partial a_j} = \sum_k \frac{\partial E^i}{\partial a_k} \frac{\partial a_k}{\partial a_j} = \sum_k \delta_k \frac{\partial a_k}{\partial z_j} \frac{\partial z_j}{\partial a_j} = g'(a_j) \cdot \sum_k w_{jk} \delta_k$$

Avec

$a_i$  : activation de la cellule  $i$ .

$z_i$  : sortie de la cellule  $i$ .

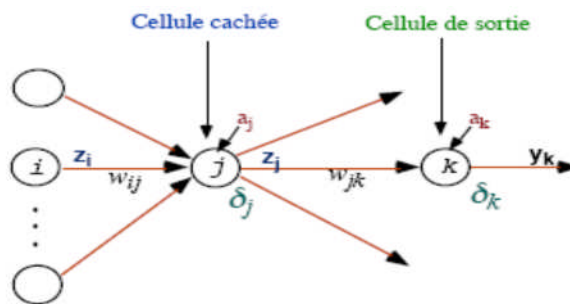


Figure 24: Evaluation de l'erreur.

2. Modification des poids

On suppose gradient à pas (constant ou non):  $\eta(t)$

Si apprentissage stochastique (après présentation de chaque exemple).

$$\Delta W_{\pi} = \eta(t) \delta_j a_i$$

Si apprentissage total (après présentation de l'ensemble des exemples).

$$\Delta W_{\pi} = \eta(t) \sum_n \delta_j^n a_i^n$$

### 3.2.2. Phase de test :

Consiste à présenter un vecteur de paramètres inconnu en entrée du réseau et à observer la réponse qu'il fournit.

## 4. Conclusion :

A travers ce chapitre, nous avons présenté l'algorithme Fuzzy C-Means implémenté et une étude sur le fonctionnement de perceptron multicouche.

Dans le chapitre suivant, nous allons présenter les tests effectués et l'évaluation des résultats obtenus.

# *Chapitre 5 Implémentation et Expérimentation*

---

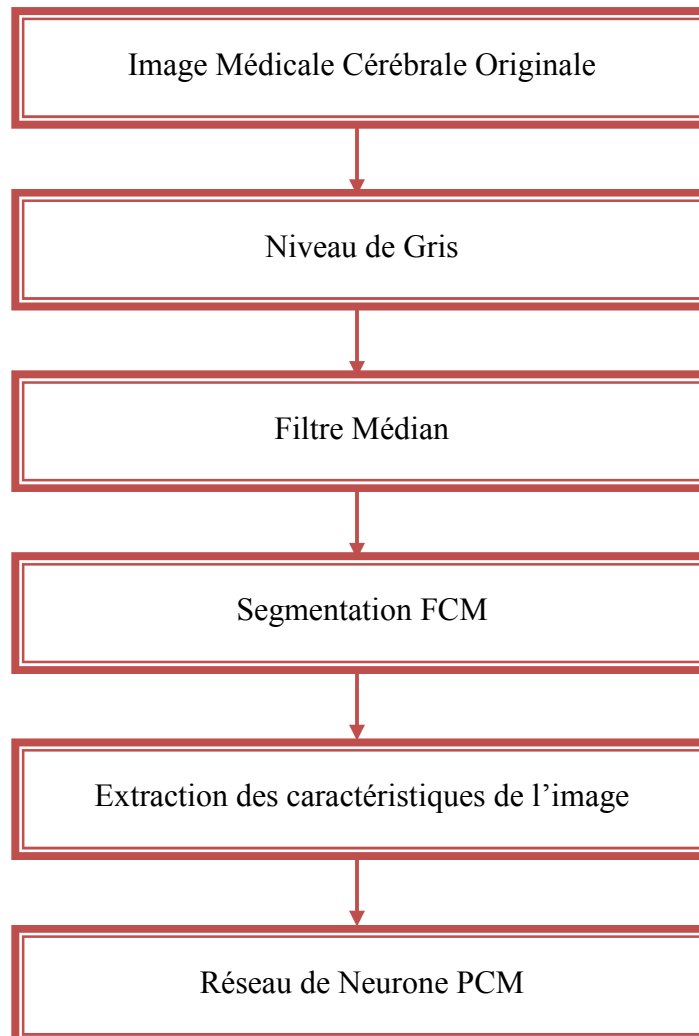
## 1. Introduction :

L'objectif de ce travail est de proposer une méthode d'interprétation dans les images médicale cérébrale . Le but de cette méthode est de faciliter la détection d'une anomalie par le réseau de neurone.

Ce chapitre dresse la synoptique de la démarche préconisée ainsi la présentation des différentes interfaces de l'application réalisée.

## 2. Synoptique de la démarche préconisée :

La synoptique de notre démarche est décrite par la Figure suivante :



**Figure 25:** Les principales étapes pour l'interprétation d'une image médicale

## 2.1. Base d'image :

Dans ce travail, nous avons utilisé la base d'images, Téléchargé sur la base de données d'images cérébrales de l'université de Harvard qui est disponible sur le site internet :

<http://www.med.harvard.edu/AANLIB/cases/case1/case.html>.

## 2.2. Prétraitement :

Après l'acquisition des images médicales cérébrales on passe au prétraitement programmé en *java* qui se fait en deux étapes suivantes :

### 2.2.1. Niveau de gris :

Le but de cette étape est de rendre tous les images à traiter au niveau de gris pour quelle soit facile à manipulé par la méthode de segmentation.

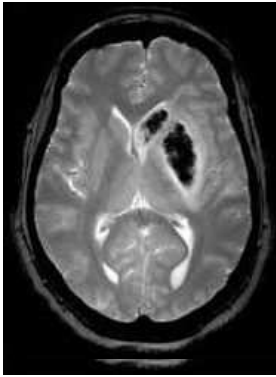
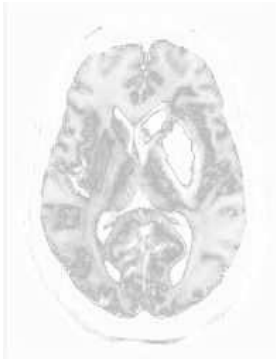
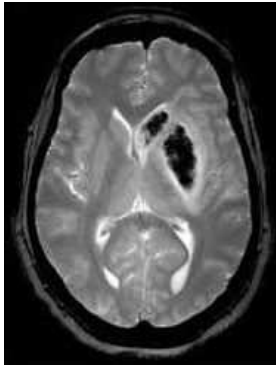

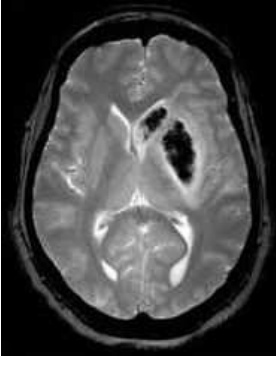
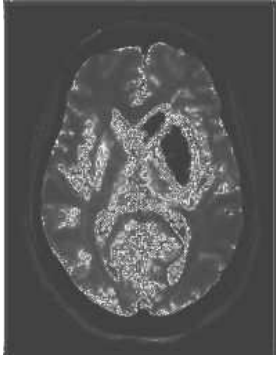
### 2.2.2. Filtre Médian :

Les filtres médians sont un type de filtres passe-bas dont le principe est de faire la médiane des valeurs des pixels avoisinants. On conclure que globalement, le filtre médian présente une meilleure fidélité à l'image initiale. De plus, il préserve bien les contours.

## 2.3. Segmentation :

Les images médicales sont segmentées par la méthode FCM détaillé précédemment, FCM représente pour chaque pixel, le degré de son appartenance à chaque classe. Il permet aussi de représenter une image pour chaque classe dont le niveau de gris reflète l'appartenance des pixels.

Le tableau suivant représente les résultats obtenus par FCM

Paramètre				Résultat Visuel	
Nombre de classe	Degré de Floue	Nombre d'itération	Epsilon	L'image a Segmenté	L'image Segmenté
2	2.0	100	1		
10	3.0	10	5		
26	2.5	5	5		

**Tableau 3:**les résultats obtenus par FCM.

## Discutions des résultats :

Le résultat de FCM représente pour chaque pixel, le degré de son appartenance à chaque classe. il permet aussi de représenter une image pour chaque classe dont le niveau de gris reflètent l'appartenance des pixels.

FCM modélise l'effet du bruit comme nos résultats l'ont montré cette modélisation dépend fortement de l'indice de flou  $m$  que l'on prend entre 2 et 3.

FCM nous offre des cartes d'appartenances facilement interprétables, nous pouvons en effet, connaître les régions de l'image possédant une plus grande appartenance à un tissu donné. la classification floue permet de tenir compte du bruit dans l'image grâce au facteur de flou  $m$ , celui-ci peut être ajusté de telle manière à avoir la tolérance au bruit désirée

### 2.4.Extraction des caractéristiques de l'image segmentée:

Le choix des attributs pour caractériser une région est un problème difficile qui demande toute l'expérience du traiteur d'images. Ce choix dépend de l'image à traiter et du problème à résoudre. Après avoir segmenté les images par «FCM », nous allons extraire les caractéristiques des ces images.

### 2.5.Classification par le réseau MLP (PMC) :

Après avoir obtenu les caractéristiques d'une image on crée un réseau de neurone sous matlab en suivant les étapes suivantes :

1. On crée la matrice qui contient les caractéristiques de l'image ( $a = [xxx]$  où  $xxx$  : les caractéristiques)
2. On normalise notre matrice pour rendre les valeurs entre  $[0,1]$  en utilisant les 2 fonctions suivantes :

$$\triangleright a = a - \min(a(:)); \quad (\text{fonctions utilisées sous matlab})$$

$$\triangleright a = \frac{a}{\max(a(:))};$$

3. Puis on met notre matrice normalisée sous forme de vecteur en utilisant la fonction qui suit :

$$r = \text{reshape}(a', 1, \text{par});$$

#### Paramètres:

$r$  : c'est le vecteur qui reçoit la matrice.

$a'$  : c'est notre matrice.

$\text{par}$ : nombre de caractéristique de notre matrice.



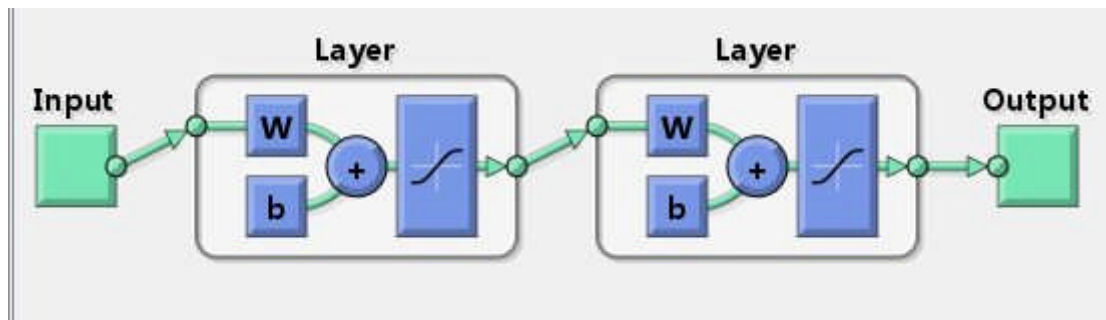
4. On crée notre réseau de type perceptron multicouches (feed-forward), il contient trois couches : la couche d'entrée (input layer) qui recevra les caractéristiques, la couche cachée et la couche de sortie.

Les entrées et les sorties désirées sont divisées en trois ensembles :

- Un ensemble d'apprentissage.
- Un ensemble de validation.
- Un ensemble de test.

L'ensemble d'apprentissage est utilisé pour entrainer le réseau. Les informations se propagent de la couche d'entrée vers la couche de sortie dans un seul sens.

Le réseau de neurone proposé est illustré dans la figure suivante :



**Figure 26:** le modèle du réseau de neurone proposé.

### 3. Application:

Nous présentons dans cette partie les différentes interfaces de notre application.

#### 3.1. Environnement matériel et logiciel de programmation :

Notre application a été réalisée sur un PC de type AMD E-300, 2Go de RAM sous Windows 7. Les langages utilisés sont: l'environnement Netbeans et Matlab version 2010.

#### 3.2. Présentation de quelques vues :

L'application est composée d'une fenêtre principale, elle contient des menus permettant d'accéder aux différentes étapes de fonctionnement de notre application.



Figure 27: Fenêtre principale de l'application.

Le Bouton « a propos » donne des informations sur le projet :

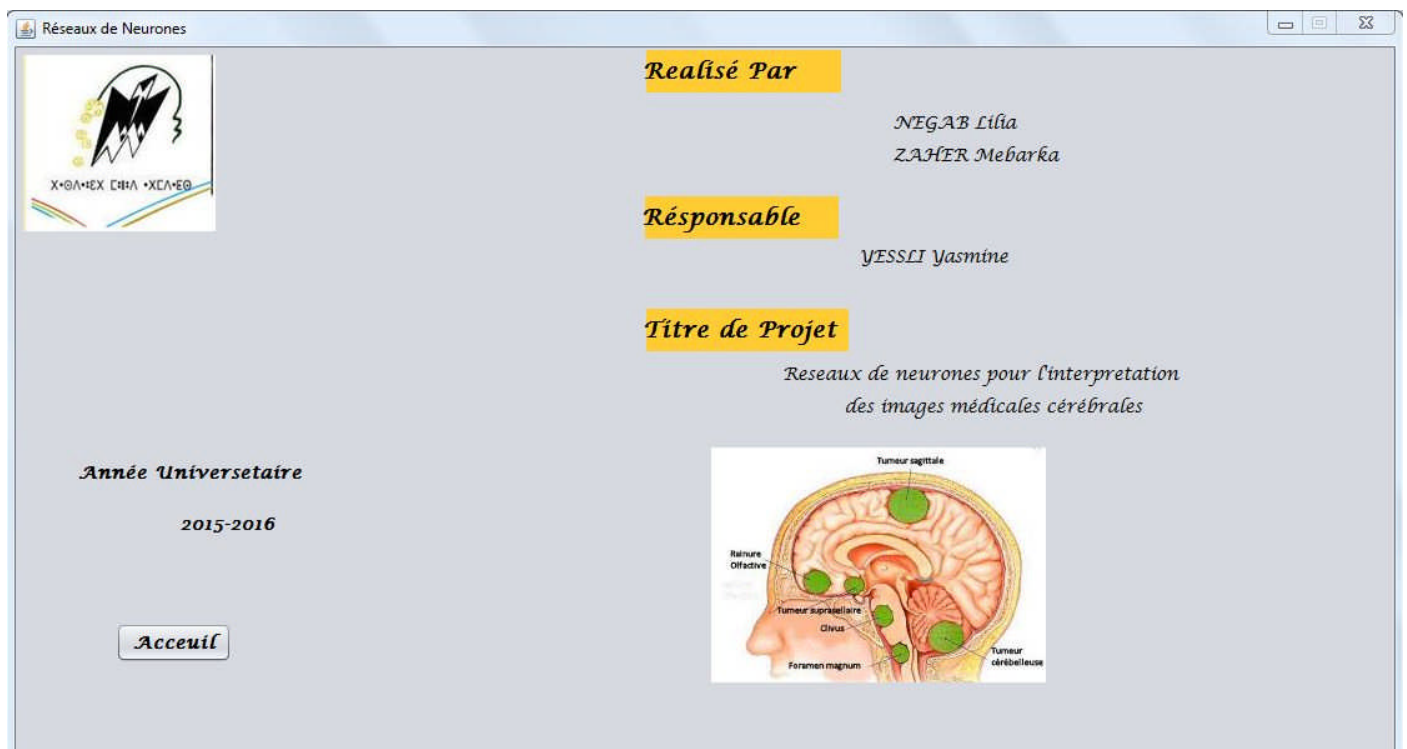


Figure 28: Information sur le logiciel.

En exécutant le bouton « entrer » depuis le menu de démarrage l'interface suivante apparaît :

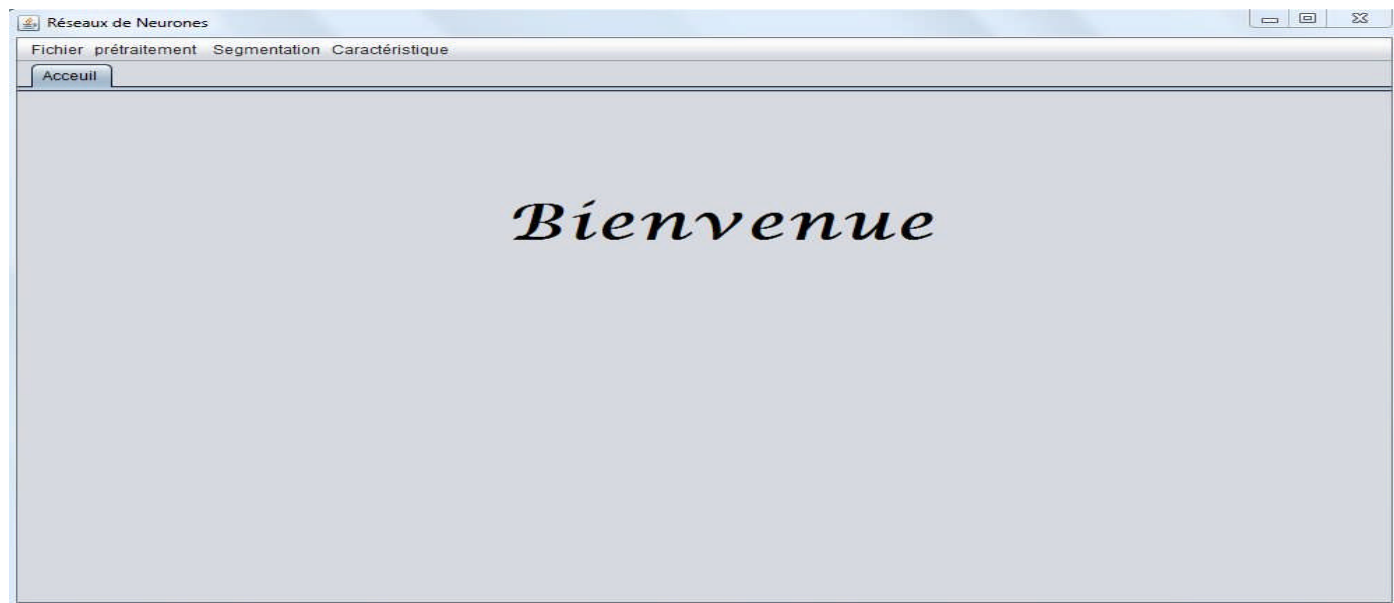


Figure 29:fenêtre de traitement du logiciel.

Le Menu « Segmentation » donne l'interface qui nous permet de segmenter l'image par l'algorithme FCM.

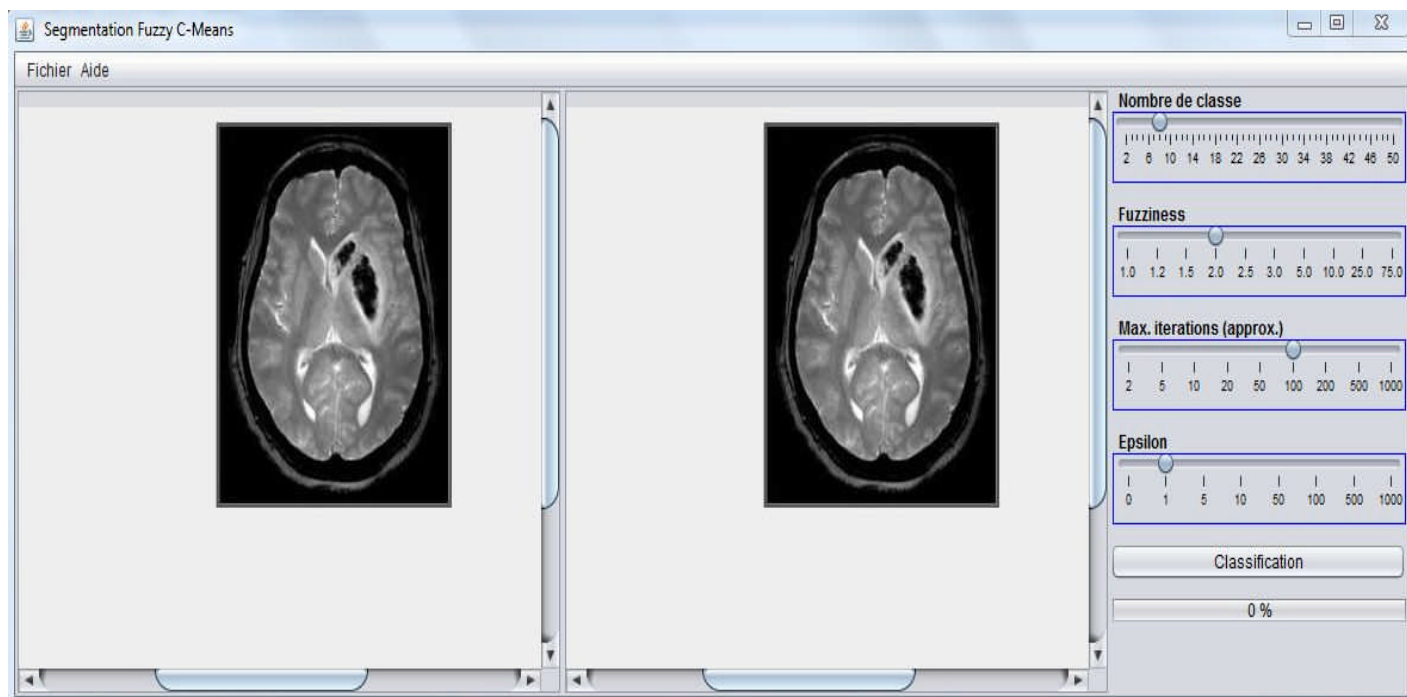
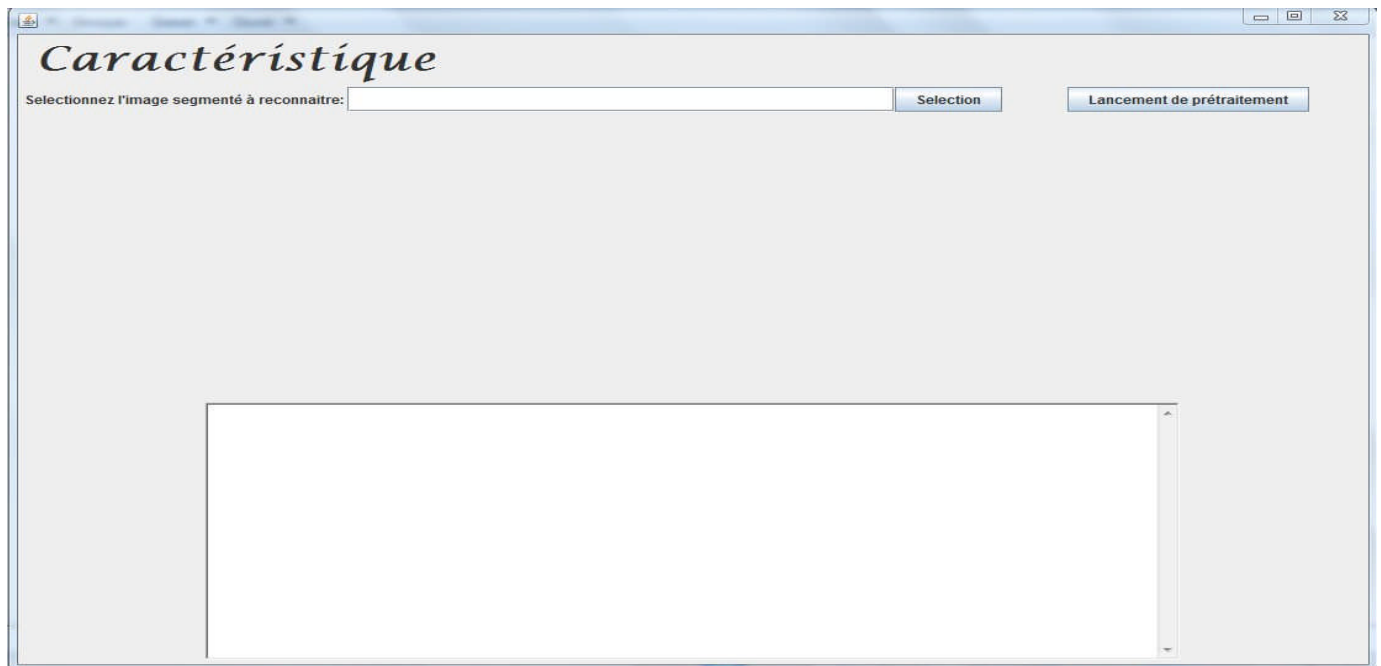
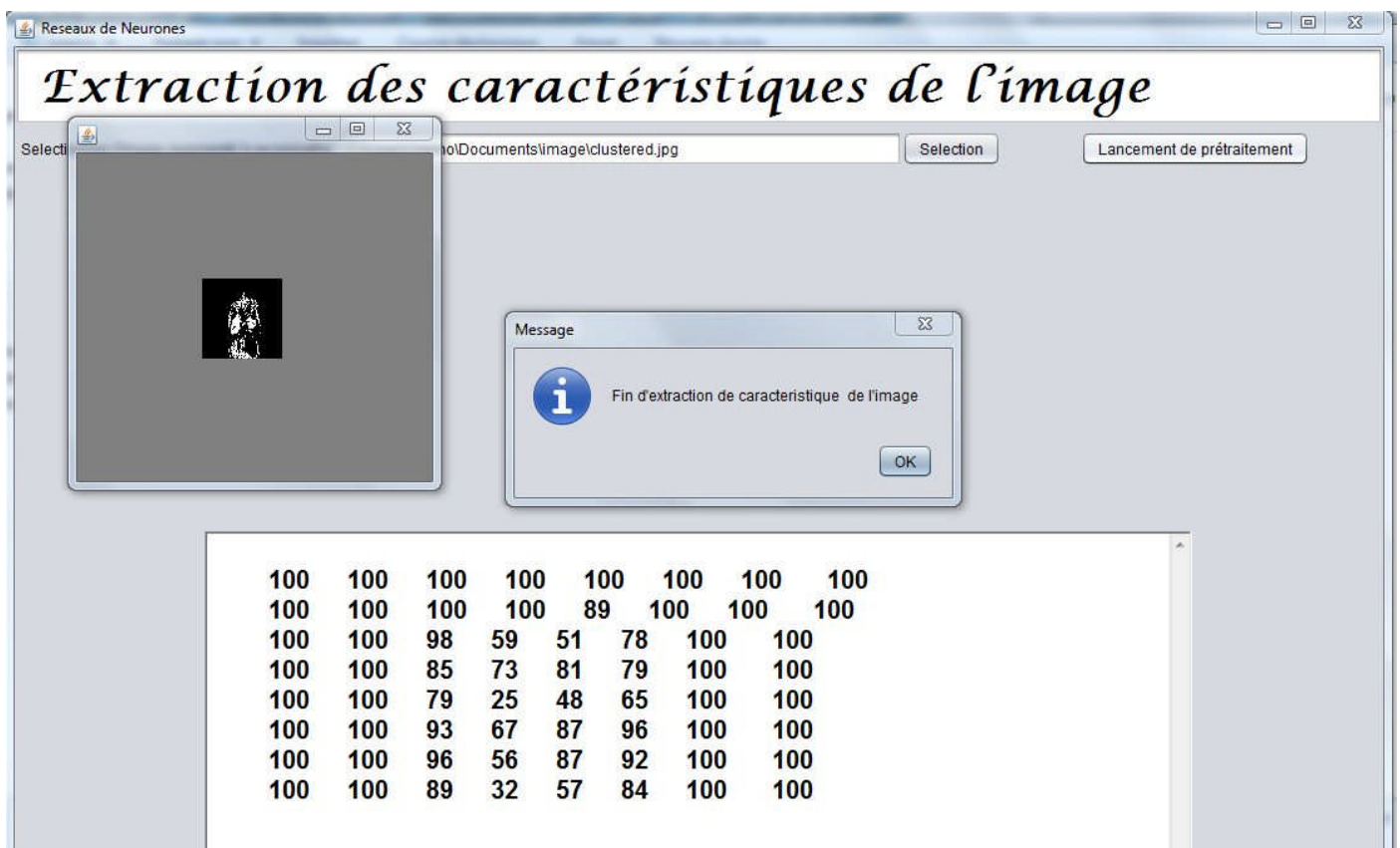


Figure 30:Interface de Fuzzy c means.

Le Menu « Caractéristique» donne l'interface qui nous permet d'extraire le nombre de pixel d'une image segmenté par FCM :

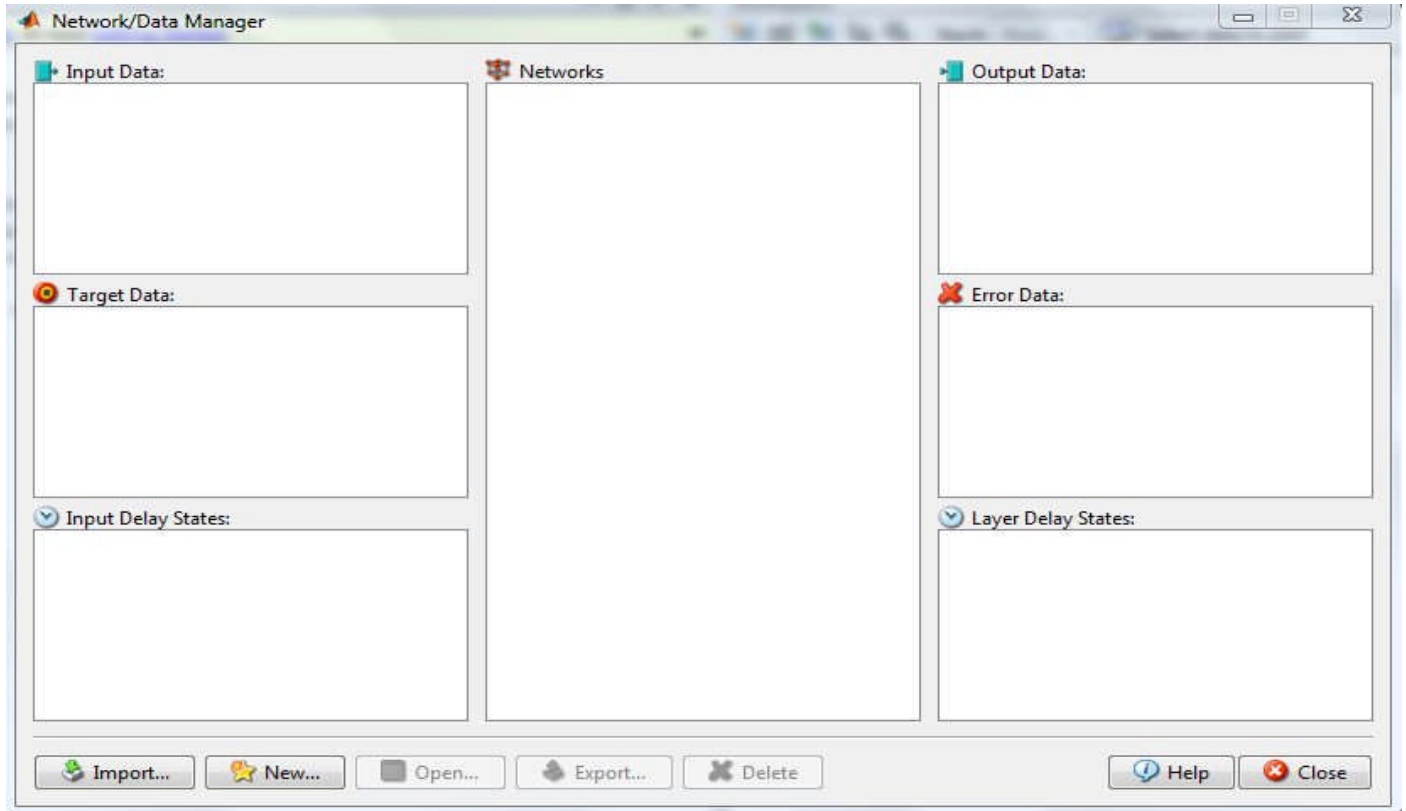


**Figure 31:**Interface de l'extraction de caractéristiques avant la sélection d'une image.

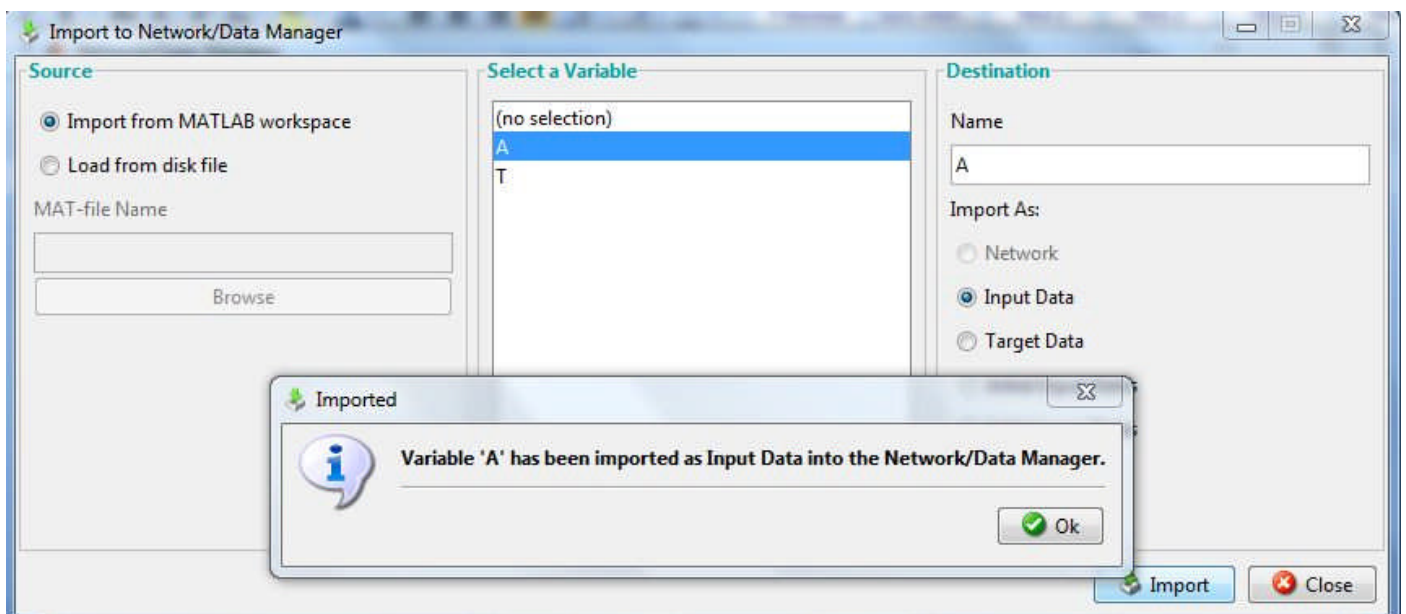


**Figure 32:**Interface de l'extraction de caractéristiques après la sélection d'une image.

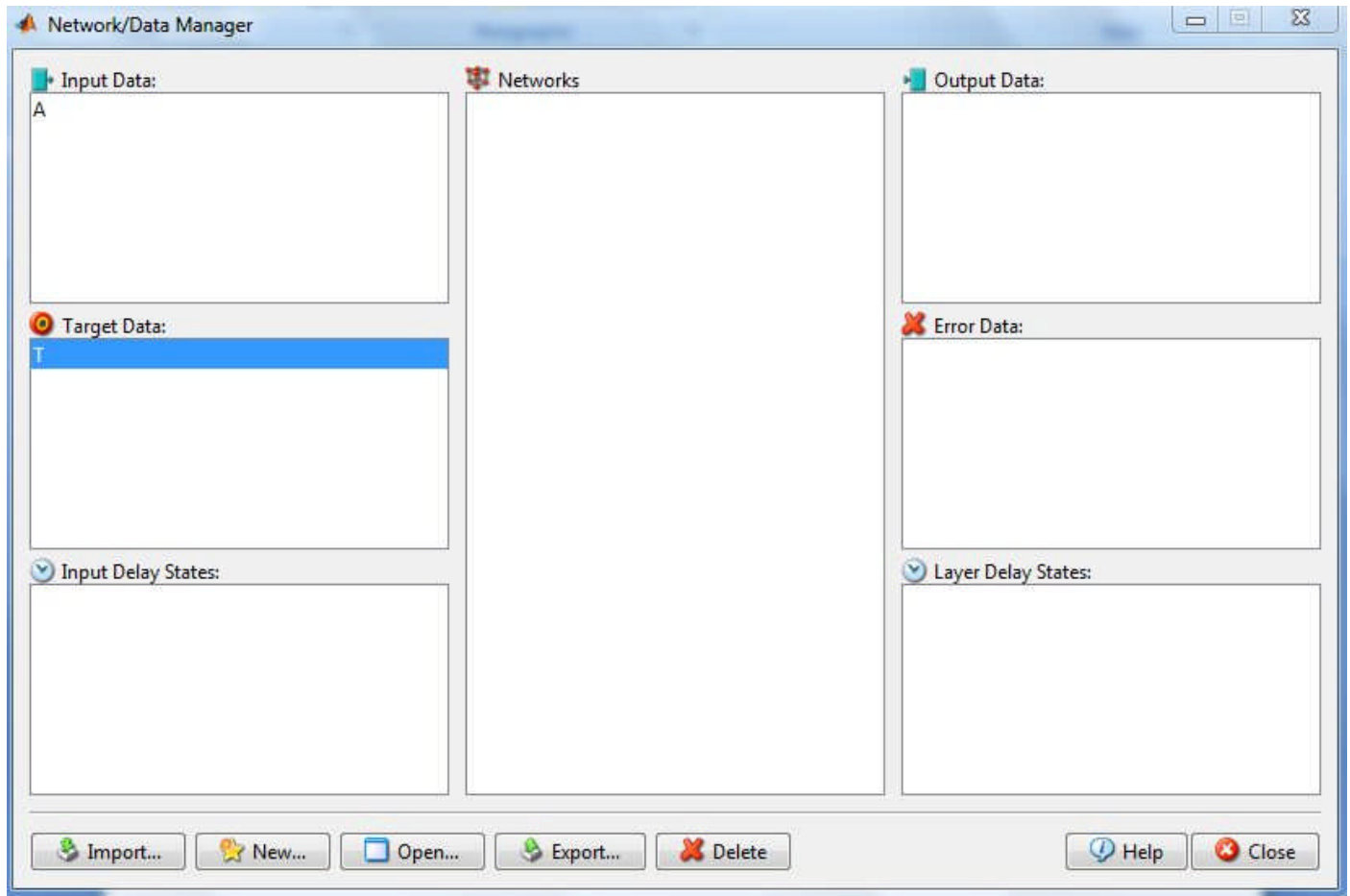
Pour créer notre réseau de neurone nous tapons sous matlab la commande data qui correspond à nos données (caractéristiques de l'image) puis la commande targt qui correspond aux résultats attendus suivie de la commande nntool pour avoir la figure suivante :



En cliquant sur le bouton import une autre fenêtre s'affichera pour importer les données et les targts comme le montre la figure suivante :

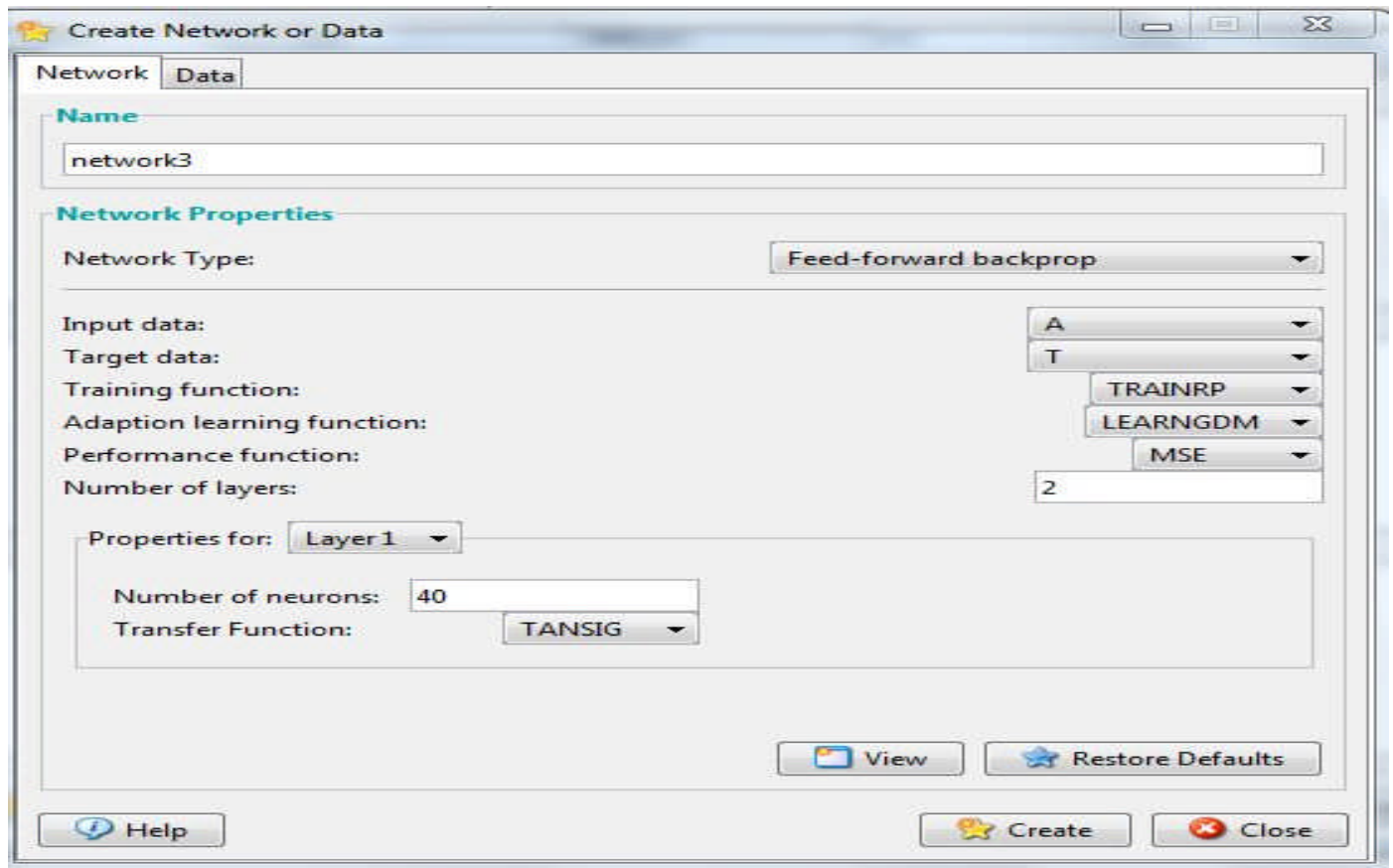


Après avoir importé les données et les targs on obtient ce qui suit :

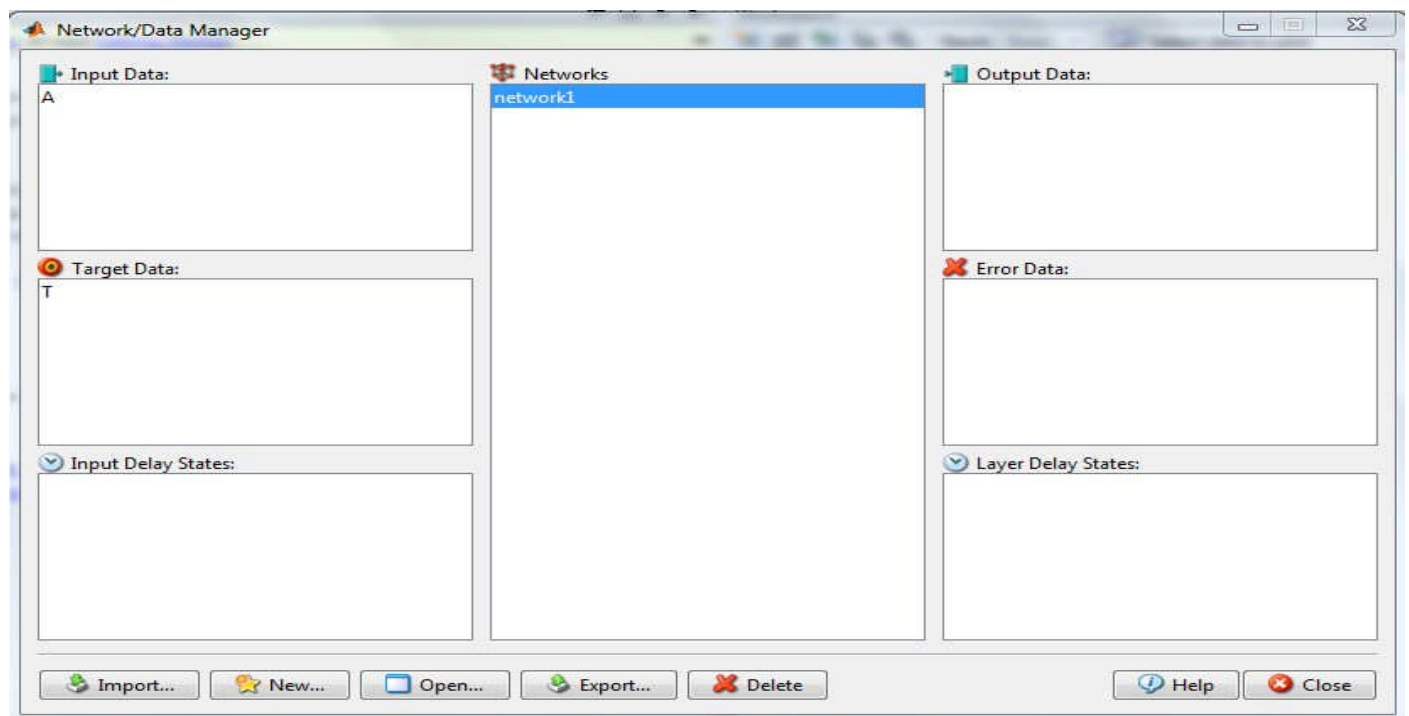


En cliquant sur le bouton new une nouvelle fenêtre s'affichera pour faire le choix des paramètres de notre réseau de neurones, la figure suivante le montre :

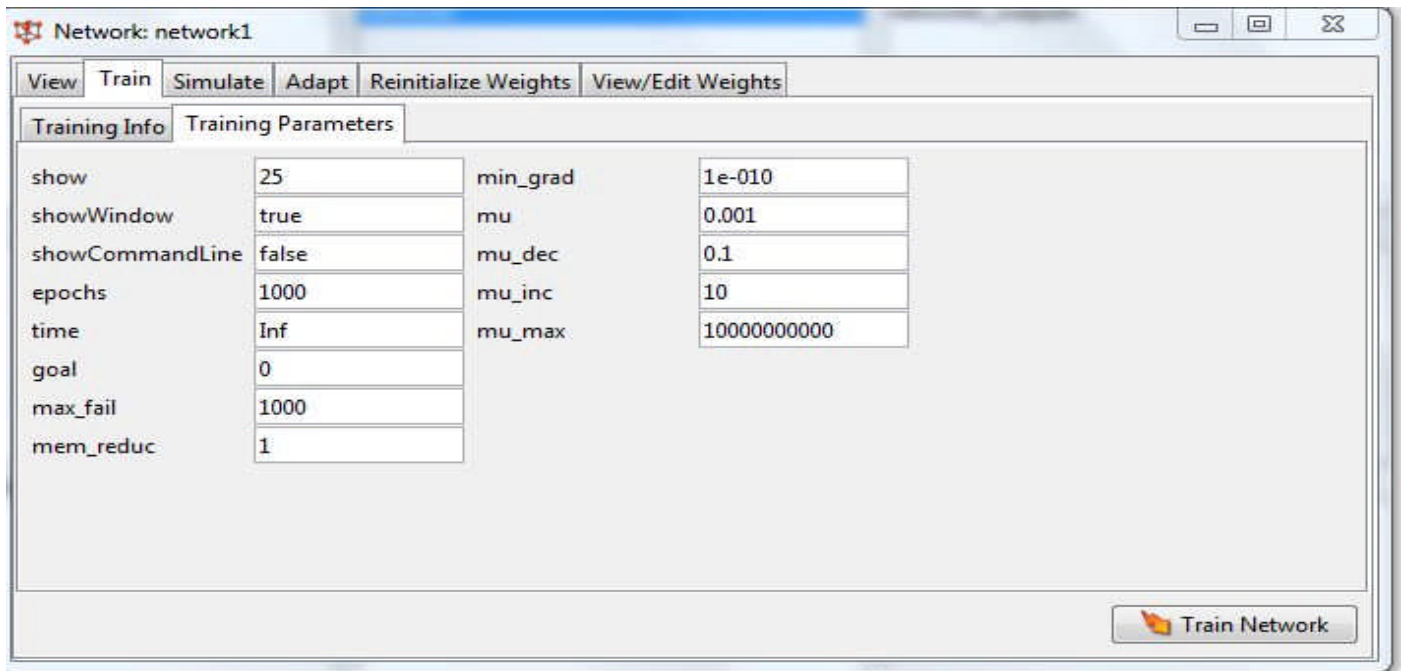




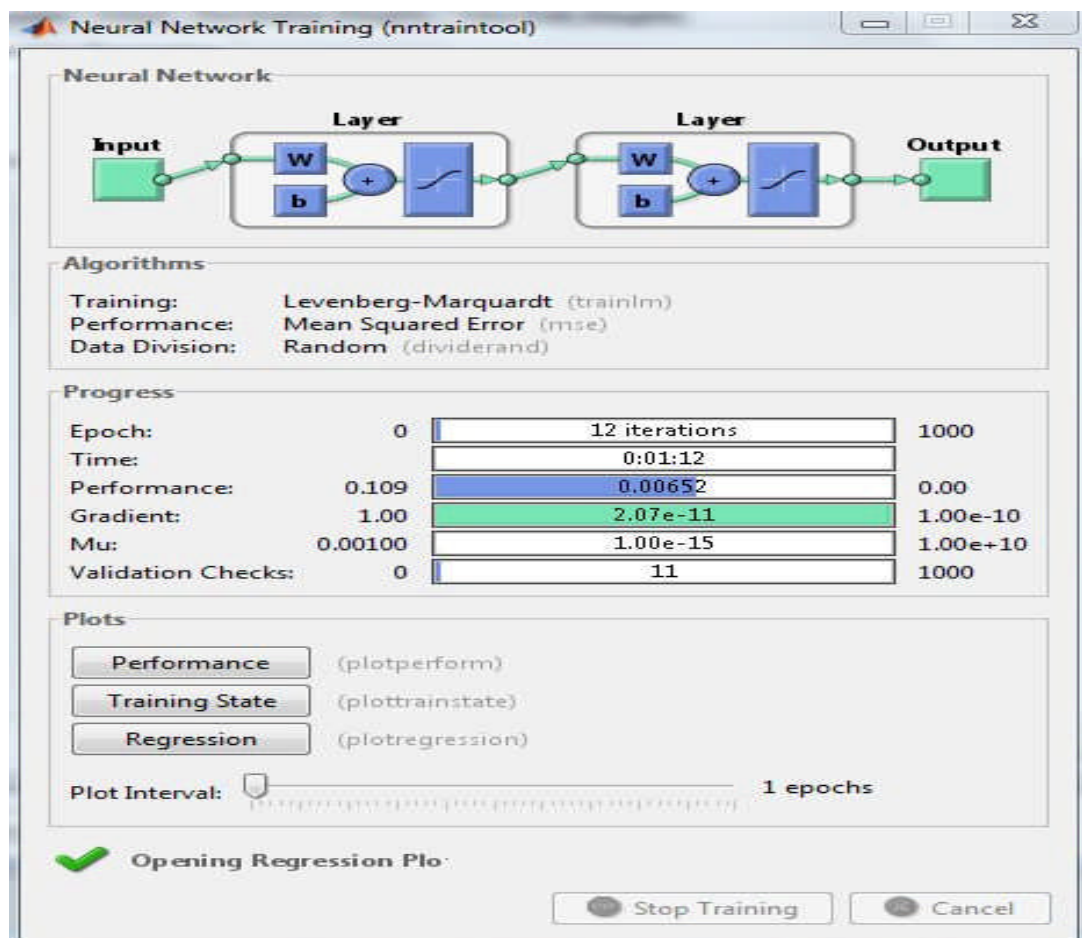
En cliquant sur le bouton create on aura :



Après avoir sélectionné network1 (le nom de notre réseau) on clique sur open et on obtient ce qui suit

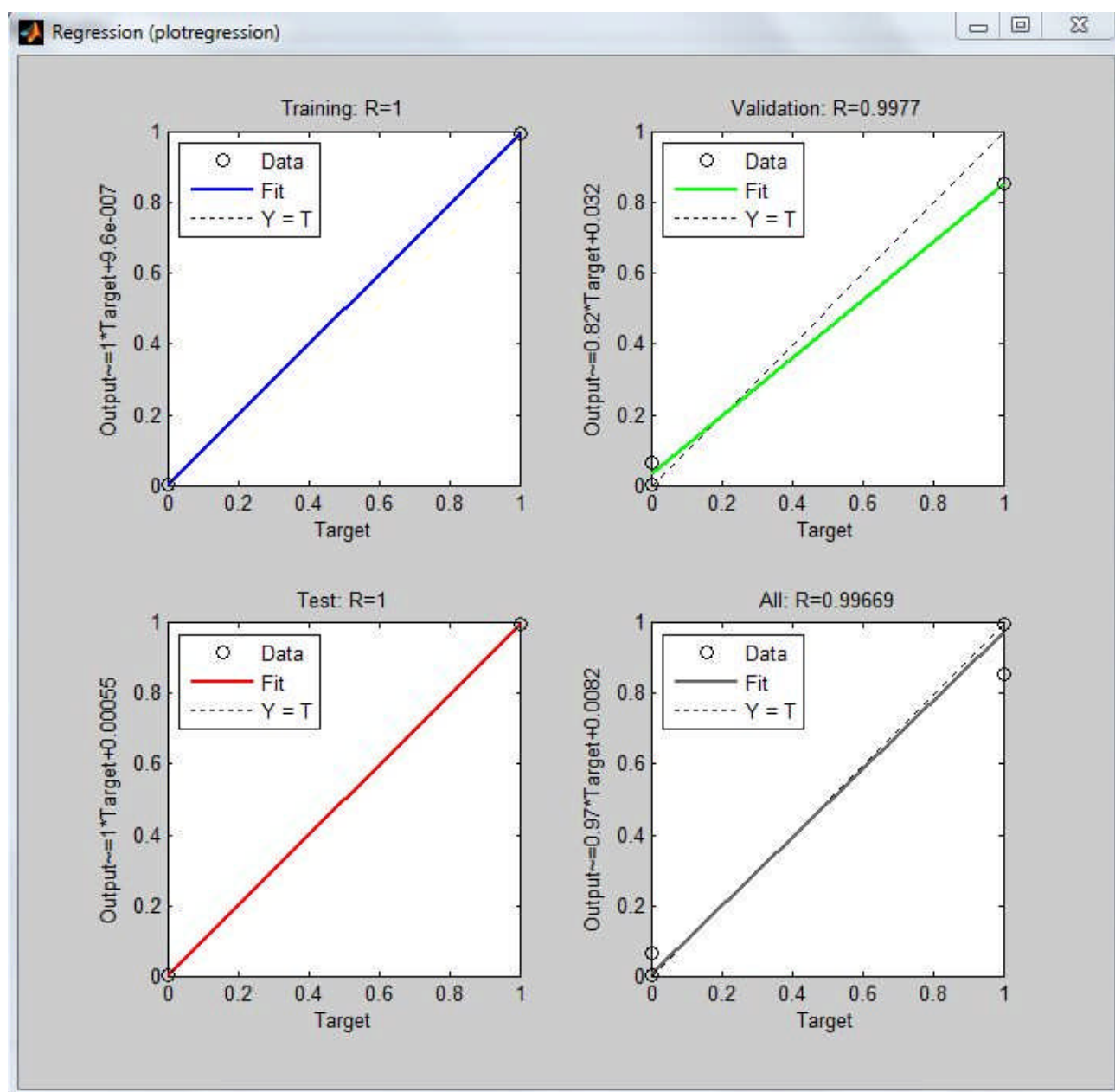


En cliquant sur le bouton train network on aura :





En cliquant sur le bouton Regression on aura les résultats avec un apprentissage de 100%, un teste de 100% et une validation de 99% comme le montre la figure suivante :



**Figure 33:** Interface de régression du réseau de neurone.

### 3.3. Discussion des résultats :

La segmentation par une approche classification de type « Fuzzy C means » a nécessité la détermination de caractéristique. Bien que cette méthode a donné des résultats encourageants et ces attributs extraits à partir des régions sont présentés aux entrées du réseau de neurone MLP conçu pour la détection. Comme il est difficile pour un problème donné de savoir l'architecture optimale qui sert à le résoudre, on a essayé plusieurs tests de création du réseau en changeant les paramètres et le nombre de neurone afin d'avoir les meilleurs résultats. Pour arriver aux résultats obtenus nous avons donné 14 image comme entrées et choisi 40 neurones pour la couche cachée ce qui nous donne un apprentissage de 100%, un teste de 100% et une validation de 99% comme le montre la figure33.

Dans les graphes de la figure précédente : le trait continue correspond a la fitness qu'on compare a  $y=t$ , les outputs représentent les sorties calculées et les targets représentent les sorties désirées et pour dire que le résultat est bon il faut avoir  $\text{output}=\text{target}$ . (bleu c'est apprentissage, rouge c'est le test, vert c'est la validation et le gris c'est la moyenne des trois autres).

### 4. Conclusion :

Au cours de ce dernier chapitre, nous avons décrit l'étape de réalisation de notre système. Aussi, nous avons présenté les étapes suivies pour développé notre application dans le but de détecter une anomalie d'une image médicale cérébrale. Et nous avons terminé par quelques interfaces démonstratives notre application.

# Conclusion Générale

---

L'objectif de notre mémoire c'est l'implémentation d'un réseau de neurone pour la détection d'une tumeur cérébrale sur une image médicale. Ce réseau de neurone a pour rôle d'aide à l'interprétation des images médicales et fournir au médecin des informations précises sur l'existence ou non d'anomalie sur ces dernières.

Afin de réaliser notre objectif, nous avons fait une étude sur les approches de segmentation d'image. Après avoir balayé les différentes approches existantes, nous avons approfondi notre étude sur l'approche classification floue (algorithme C-Moyennes Floues- FCM). FCM est un algorithme qui permet d'obtenir une classification floue tel que son nom l'indique, c'est-à-dire qu'un pixel peut appartenir à plusieurs classes avec des degrés d'appartenances compris entre  $[0,1]$ .

Le travail présenté a permis d'intervenir et d'améliorer les différentes étapes de l'analyse des images médicales. Après une estimation globale et un choix adéquat des prétraitements, nous avons fait appel à : la segmentation par une approche de classification « FCM », puis l'extraction des caractéristiques de ces images et faire rentrer ces dernières dans un réseau de neurone de type MLP sous matlab. Les résultats de la segmentation et la détection obtenus montrent que le système a pu donner des meilleures performances en reconnaissance. Il demeure évident que ces méthodes ne remplaceront jamais l'œil du clinicien, mais en lui offrant un outil d'interprétation plus rapide et minutieux, elles seront toujours un outil fiable d'aide au diagnostic ainsi qu'à la détection de la pathologie.

En perspective nous envisageons de reprendre le traitement des images en utilisant un autre type réseau de neurones par exemple RBF sans passer par la segmentation.

# *Bibliographie :*

[1] : pablo andres, arbeláez escalante « Une approche métrique pour la segmentation d'images » Thèse pour l'obtention du titre de docteur en sciences, université paris dauphine U.F.R. mathématiques de la décision, Promotion 2005.

[2] : HOUASSINE Charif « Segmentation d'images par une approche biomimétique hybride. » Mémoire de magister pour l'obtention du diplôme de magister en génie électrique et électrotechnique, Laboratoire de recherche sur l'électrification des entreprises industrielles. Promotion 2012.

[3] : Cours rédigé par maitre Henri « La segmentation par régions»

[4] : Benôt Scherrer « Segmentation des tissus et structures sur les IRM cérébrales : agents markoviens locaux coopératifs et formulation bayésienne » HAL Id.

[5] : Larbi Kahina « Segmentation d'images basé sur la modalisation statique d'histogramme » Mémoire de magister pour l'obtention du diplôme de magister en Automatique Option : Traitement d'images et Reconnaissance de formes, UMMTO, Promotion 2012.

[6] : catherine rouquet, frederic chausse, roland chapuis, pierre bonton «Unsupervised Segmentation of Road Images.A Multicriteria Approach» Laboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique

[7] : Berrichi Fatima Zohra, Benyettou Mohamed « L'Interprétation des Images Médicales Cérébrales : Segmentation par L'approche Split et Merge » Laboratoire de Modélisation et Optimisation des Systèmes Industriels, Faculté des Sciences, Département d'Informatique, USTO.

[8] : S.AMEUR, Z. AMEUR « revue des approches de segmentation d'images texturées : exemple des images météorologiques », Laboratoire d'analyse et de modélisation des phénomènes aléatoires

[9] : Bentayeb Nawel, Attar Tassadit « Segmentation d'Imagerie Médical» mémoire de fin d'étude pour l'obtention du diplôme de Master en Système informatique, UMMTO, promotion 2014/2016.

[10] : Imane SEBARI et Dong-Chen HE, « les approches de segmentation d'image par coopération régions-contours » Article de centre d'applications et de recherches en télédétection (CARTEL), 2008

[11] : Hadjer LAGUEL « Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images IRM basé sur les systèmes multi-agents.» Mémoire de Fin d'Étude pour l'obtention du diplôme d'ingénieur d'état Option : Traitement d'images, Université des Sciences et de la Technologie Houari Boumediene.

[12] : Ouarda Assas « Classification floue des images »Thèse de doctorat en sciences, Université de Batna, promotion 2013.

[13] : Aiche Yasmine , Koulougli Dalila « Implémentation d'une méthode de Clustering pour le calcul de la probabilité de pertinence a priori d'un document » Mémoire de fin d'étude pour l'obtention du diplôme de master en informatique, promotion 2013.

[14] : Gérard Dreyfus « Reconnaissance de formes par réseau de neurones » Ecole Polytechnique, juillet 2002.

[15] : Nassima Mezhoud «Une approche hybride pour la segmentation d'images Application aux images médicales » Thèse de doctorat en sciences spécialité informatique, l'université de Constantine, promotion 2012.

[16] : Aoudjit-Lagab.R « Les réseaux de neurones» cours master2 RMSE, promotion 2016.

[17] : Mohamed Lamine TALBI «Analyse et traitement du signal électro- cardiographique (ECG) » Présentée pour obtenir le diplôme de Doctorat en Sciences Spécialité : Electronique, Option : Traitement du signal, promotion 2011

[18] : Claude TOUZET «Les réseaux de neurones artificiels », cours, exercices et travaux pratiques, promotion Juillet 1992.

[19] : Gérald PETITJEAN «introduction aux réseaux de neurones ».

**[20]** : « chapitre 8 : Les réseaux de neurones compétitifs ».

**[21]** : Éric Hervet « Réseaux de neurones» INFO6517 Université de Moncton Faculté des sciences, Département d'informatique Moncton, N.-B., CANADA E1A 3E9.

**[22]** : Philippe Paquet «L'utilisation des réseaux de neurones artificiels en finance », Documents de recherche n°1997-1.

**[23]** : Habib Mahdi, Sbahi Toufik « Segmentation d'images IRM : application sur des IRM de patients atteints l'Alzheimer » Mémoire de master pour l'obtention de diplôme de master en informatique, UMMTO, Promotion 2015

**[24]** : livre Monica Baciú « Bases de neurosciences, Neuroanatomie fonctionnelle»

# Webographie :

[W1]: <http://blogmatlab.blogspot.com/2011/04/la-segmentation-dimages.html#more>

[W2]: <http://www.tsi.enst.fr/pages/enseignement/ressources/mti/gric/schedule.html>

[W3]: <http://webcache.googleusercontent.com/search?q=cache:http://helene.boyer.free.fr/epita/FG/expooses/25.pdf>

[W4]: <http://www.peoi.org/Courses/Coursesfr/neural/neural4.html>

[W5]: <http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie005.html#toc13>

[W6]: <http://www.peoi.org/Courses/Coursesfr/neural/neural5.html>

[W7]: <http://www.peoi.org/Courses/Coursesfr/neural/neural6.html>

[W8]: <http://www.techno-science.net/?onglet=glossaire&definition=911>

[W9]: [http://www.larousse.fr/encyclopedie/divers/systeme\\_nerveux/73116](http://www.larousse.fr/encyclopedie/divers/systeme_nerveux/73116)

# Annexe :

## 1. L'image numérique :

### Définition :

On appelle l'image numérique toute image (dessin, icône, photographie), acquise ou créée, traitée et stockée sous forme binaire.

L'image numérique peut subir les opérations suivantes :

- Le traitement grâce à des outils informatiques, de façon à la transformer, à en modifier la taille, les couleurs, d'y ajouter ou d'en supprimer des éléments, d'y appliquer des filtres variés.
- Le stockage sur un support informatique (disquette, disque dur, CD-ROM).

### Les types d'images numériques

Les images numériques se divisent en deux grandes classes :

#### 1. Les images vectorielles :

Les images vectorielles sont adaptées ou travaillent sur des objets dont on connaît les paramètres de traçage, elles sont décrites en termes de formes élémentaires : lignes, cercles, rectangles, splines, courbe de Bézier, etc. Les formes sont décrites par des attributs géométriques et par des attributs d'épaisseur, de couleur.

#### 2. Les images matricielles :

Encore appelées images Bitmap, l'image matricielle se représente sous forme d'une matrice de point (image binaire, points allumés et points éteints ; image couleur : points de couleur définie). Ces points codés sont rangés en lignes et en colonnes avec la correspondance simple suivante : un élément de la matrice (point codé) correspond à un point de l'écran de l'ordinateur.

#### 3. Codage de Couleurs :

Une image bitmap est représentée par un tableau de pixels dont chaque case *contient une valeur*. La valeur stockée dans une case du tableau est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur).

Il existe plusieurs standards de codage de la profondeur :

##### 3.1. Le Codage en Noir et blanc :

Une image en noir et blanc est constituée uniquement de bits à 0 ou 1. Chaque pixel est soit noir, soit blanc. Il faut un bit pour coder un pixel (0 pour noir, 1 pour blanc). Ce type de codage peut convenir pour un plan ou un texte, mais on voit ses limites lorsqu'il s'agit d'une photographie.

##### 3.2. Le codage en niveaux de gris :



Une image en niveaux de gris contient des tons de gris variant entre le noir et le blanc. Si on code chaque pixel sur 2 bits, on aura 4 possibilités (noir, gris foncé, gris clair, blanc). L'image codée sera très peu nuancée. En général, on code chaque pixel sur 8 bits = 1 octet. On a alors 256 possibilités (on dit 256 niveaux de gris).

### **3.3. Codage des Couleurs**

Les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités.

En effet, avant de représenter l'image couleur, il faut Choisir un modèle de représentation de la couleur (ou mode colorimétrique), Puis choisir la profondeur de couleur.

## **2. Tumeur cérébrale :**

Une tumeur cérébrale est une masse de cellules anormales qui se multiplient dans le cerveau de façon incontrôlée.

## **3. Les types de tumeurs cérébrales :**

Il existe 2 grands types de tumeurs cérébrales selon leur nature cancéreuse ou non :

### **1. Les tumeurs bénignes (non cancéreuses) :**

Elles se forment assez lentement et restent le plus souvent isolées des tissus cérébraux voisins. Elles ne se propagent pas à d'autres parties du cerveau ni à d'autres organes et sont généralement plus faciles à extraire par chirurgie que les tumeurs malignes. Cependant, certaines tumeurs pourtant bénignes demeurent inextirpables du fait de leur localisation.

### **2. Les tumeurs malignes (cancéreuses) :**

Il n'est pas toujours facile de les distinguer des tissus voisins. De ce fait, il est parfois difficile de les extraire entièrement sans endommager le tissu cérébral avoisinant.

Des examens, comme l'imagerie par résonnance magnétique (IRM), PET scan (tomoscintigraphie par émission de positons) et la tomodensitométrie (« CT scan »), permettent de localiser précisément la tumeur. Une biopsie (prélèvement de tissu tumoral afin de l'analyser) est essentielle pour déterminer la nature bénigne (non cancéreuse) ou maligne (cancéreuse) de la tumeur.

Les tumeurs cérébrales se distinguent aussi selon leur provenance et leur localisation On distingue :

## 1. Les tumeurs cérébrales primaires :

Sont celles qui prennent naissance dans le cerveau. Elles peuvent être bénignes (non cancéreuses) ou malignes (cancéreuses). Leur dénomination provient du tissu cérébral dans lequel elles se développent.

Parmi les tumeurs malignes les plus fréquentes on retrouve :

### 1.1. Les tumeurs gliales, ou gliomes (tumeurs malignes) :

Représentant 50 à 60 % de l'ensemble des tumeurs cérébrales. Elles se forment à partir des cellules gliales, cellules intervenant comme structure de soutien des cellules nerveuses (neurones).

**1.2. Les médulloblastomes (tumeurs malignes) :** se développent à partir de la moelle épinière au stade embryonnaire. Ce sont les tumeurs cérébrales les plus courantes chez les enfants.

- Enfin, parmi les tumeurs primaires bénignes, plus rares que les tumeurs primaires malignes, on retrouve les hémangioblastomes, les méningiomes, les adénomes hypophysaires, les ostéomes, les pinéalomes, etc.

## 2. Les tumeurs secondaires ou métastatiques :

Sont malignes (cancéreuses) et proviennent d'autres organes où existe un cancer et dont les cellules tumorales ont migré vers le cerveau et s'y multiplient. Les cellules tumorales sont transportées par le sang et se développent le plus souvent à la jonction entre la substance blanche et la substance grise du cerveau. Ces tumeurs secondaires sont plus fréquentes que les tumeurs primaires. D'ailleurs, on estime que 25 % des personnes décédées de cancers de toutes natures sont porteuses de métastases cérébrales. Parmi les tumeurs les plus fréquemment à l'origine de métastases cérébrales : le cancer du sein, le cancer du poumon, le cancer de la peau (mélanome), le cancer du rein, le cancer du côlon, etc.

## 3. Description des Méthodes de L'algorithme FCM :

### 1. classe « ImageProcessingTask »

Cette classe abstraite représente une tâche de traitement d'image qui est exécuté sur son propre thread. Cette classe utilise le concept de taille de la tâche de traitement et de la position, si l'on peut estimer la taille de la tâche, une application peut obtenir la position du traitement, à savoir à quel point nous sommes sur la tâche.

#### Méthodes :

#### 1. Run() :

```
public void run();
```

Cette méthode fera le traitement de l'image en vrac. Il est déclaré comme abstrait comme un rappel pour le programmeur, qui doit la mettre en œuvre.

## 2. `getSize()` :

```
public abstract long getSize();
```

Cette méthode retourne la taille de la tâche de traitement d'image. La taille peut être une valeur estimée mesurée dans une unité. Cette méthode doit être mise en œuvre sur les classes qui héritent de celui-ci et son résultat doit être constant.

Renvoie: la taille de la tâche de traitement d'image.

## 3. `getPosition()` :

```
public abstract long getPosition();
```

Cette méthode retourne la position de la tâche de traitement d'image, à savoir le nombre d'étapes de traitement ont déjà été fait. Les classes qui héritent de celui-ci doit mettre en œuvre cette méthode.

Renvoie : la position du traitement de la tâche.

## 4. `isFinished()`

```
public abstract boolean isFinished();
```

Cette méthode renvoie true si la tâche est terminée. Cette méthode doit être mis en œuvre sur les classes qui héritent de celui-ci.

Renvoie: Vrai si la tâche de traitement est terminé

## 2. Classe « `DisplayJAIWithPixelInfo` »

Cette classe montre comment on peut étendre la classe `DisplayJAI`. Nous allons remplacer la méthode `mouseMoved` de `DisplayJAI` quand la souris est déplacée, et on stocke des informations sur le pixel sous la souris (mais pas affiché).

### Paramètre:

- `image` : un `RenderedImage` pour l'affichage.

### Méthodes :

#### 1. `DisplayJAIWithPixelInfo()` :

Cette méthode permet de créer les tableaux et les instances nécessaires Pour obtenir les données d'image et enregistre la classe pour écouter le mouvement de la souris

```

public DisplayJAIWithPixelInfo(RenderedImage image)    {
    super(image);
    readIterator = RandomIterFactory.create(image, null);
    width = image.getWidth();    height = image.getHeight();
    int dataType = image.getSampleModel().getDataType();
    switch(dataType){    case DataBuffer.TYPE_BYTE:
                            case DataBuffer.TYPE_SHORT:
                            case DataBuffer.TYPE_USHORT:
                            case DataBuffer.TYPE_INT: isDoubleType = false; break;
                            case DataBuffer.TYPE_FLOAT:
                            case DataBuffer.TYPE_DOUBLE: isDoubleType = true; break;    }
    if (isDoubleType) dpixel = new double[image.getSampleModel().getNumBands()];
    else ipixel = new int[image.getSampleModel().getNumBands()];
    isIndexed = (image.getColorModel() instanceof IndexColorModel);
    if (isIndexed)    {
        IndexColorModel icm = (IndexColorModel)image.getColorModel();
        int mapSize = icm.getMapSize();
        byte[][] templutData = new byte[3][mapSize];
        icm.getReds(templutData[0]);
        icm.getGreens(templutData[1]);
        icm.getBlues(templutData[2]);
        lutData = new short[3][mapSize];
        for(int entry=0;entry<mapSize;entry++)    {
            lutData[0][entry] = templutData[0][entry] > 0 ?
                templutData[0][entry] : (short)(templutData[0][entry]+256);
            lutData[1][entry] = templutData[1][entry] > 0 ?
                templutData[1][entry] : (short)(templutData[1][entry]+256);
            lutData[2][entry] = templutData[2][entry] > 0 ?
                templutData[2][entry] : (short)(templutData[2][entry]+256);
        }    }
}

```

## 2.mouseMoved() :

Cette méthode sera appelée lorsque la souris est déplacée sur l'image affichée

```

public void mouseMoved(MouseEvent me)    {
    pixelInfo.setLength(0);    int x = me.getX();    int y = me.getY();
    if ((x >= width) || (y >= height))    {
        pixelInfo.append("No data!");
        return;    }
    if (isDoubleType)    {
        pixelInfo.append("(floating-point data) ");
        readIterator.getPixel(me.getX(),me.getY(),dpixel);
        for(int b=0;b<dpixel.length;b++)
            pixelInfo.append(dpixel[b]+",");
        pixelInfo = pixelInfo.deleteCharAt(pixelInfo.length()-1);    }
    else    {    if (isIndexed)    {
        pixelInfo.append("(integer data with colormap) ");
        readIterator.getPixel(me.getX(),me.getY(),ipixel);
        pixelInfo.append("Index: "+ipixel[0]);
        pixelInfo.append(" RGB:"+lutData[0][ipixel[0]]+","+
                        lutData[1][ipixel[0]]+","+
                        lutData[2][ipixel[0]]);    }
    else    {    pixelInfo.append("(integer data) ");
        readIterator.getPixel(me.getX(),me.getY(),ipixel);
        for(int b=0;b<ipixel.length;b++)
            pixelInfo.append(ipixel[b]+",");
        pixelInfo = pixelInfo.deleteCharAt(pixelInfo.length()-1);    }    }    }

```

### 3.getPixelInfo() :

Cette méthode permet des accès externes à l'information de pixel qui a été obtenu dans la méthode mouseMoved.

Renvoie: les informations de pixels, formatées sous forme de chaîne

```

public String getPixelInfo()
{
    return pixelInfo.toString();
}

```

### 3. Classe « DisplayTwoSynchronizedImages »

Cette classe représente un JPanel qui contient deux instances défilants de DisplayJAIWithPixelInfo, les barres de défilement des deux images sont synchronisées de sorte défilent une image défile l'autre automatiquement.

## Paramètres:

- IM1 - la première image (côté gauche)
- IM2 - la seconde image (côté droit)

## Méthodes :

### 1. setImage1() :

Cette méthode modifie la première image à afficher.

```
public void setImage1(RenderedImage newimage)    {  
    dj1.set(newimage);
```

### 2. setImage2

Cette méthode modifie la deuxième image à afficher.

```
public void setImage2(RenderedImage newimage)    {  
    dj2.set(newimage);
```

### 3. getImage1

Cette méthode retourne la première image.

```
public RenderedImage getImage1()    {  
    return dj1.getSource();    }
```

### 4. getImage2

Cette méthode retourne la seconde image.

```
public RenderedImage getImage2()    {  
    return dj2.getSource();    }
```

### 5. getDisplayJAComponent1

Cette méthode retourne la première composante DisplayJAIWithPixelInfo.

```
public DisplayJAIWithPixelInfo getDisplayJAComponent1()    {  
    return dj1;
```

### 6. getDisplayJAComponent2

Cette méthode retourne le deuxième composant DisplayJAIWithPixelInfo.

```
public DisplayJAIWithPixelInfo getDisplayJAComponent2()    {  
    return dj2;    }
```

## 7. adjustmentValueChanged

Cette méthode sera appelée lorsque l'une des barres de défilement des instances de DisplayJAIWithPixelInfo est modifiée. La méthode ajuste la barre de défilement de l'autre

```
public void adjustmentValueChanged(AdjustmentEvent e) {  
    if (e.getSource() == jsp1.getHorizontalScrollBar()) {  
        jsp2.getHorizontalScrollBar().setValue(e.getValue());    }  
    if (e.getSource() == jsp1.getVerticalScrollBar())    {  
        jsp2.getVerticalScrollBar().setValue(e.getValue());    }  
    if (e.getSource() == jsp2.getHorizontalScrollBar())    {  
        jsp1.getHorizontalScrollBar().setValue(e.getValue());    }  
    if (e.getSource() == jsp2.getVerticalScrollBar())    {  
        jsp1.getVerticalScrollBar().setValue(e.getValue());    } } }
```

## 4. Classe « FuzzyCMeansImageClusteringApp »

Cette application affiche une utilisation de l'algorithme FuzzyCMeansImageClustering. L'image originale et en cluster sont présentées, côte à côte et synchronisé. Les données d'entrée sont lues à partir d'un fichier et à la fin de la tâche des différents résultats de clustering commandés peuvent être affichés, ainsi que sauvegardé.

### Paramètre:

1. ifile : le nom du fichier d'entrée.

### Méthodes :

#### 5. actionPerformed()

Cette méthode sera appelée quand un événement d'action se produit. Dans cette classe, il soit signifie que l'utilisateur a cliqué sur le bouton Démarrer.

```

public void actionPerformed(ActionEvent e)    {
    if (e.getSource() == start)        {
        monitor.start();
        int nClusters = numClustersSlider.getValue();
        float fuzziness = fuzzinessValues[fuzzinessSlider.getValue()];
        int maxIter = maxIterationsValues[maxIterationsSlider.getValue()];
        task = new FuzzyCMeansImageClustering(input,nClusters,maxIter,fuzziness,0);
        progressBar.setMaximum(100);
        task.start();        start.setEnabled(false);
        start.setText("Clustering, please wait...");
        numClustersSlider.setEnabled(false);        maxIterationsSlider.setEnabled(false);
        fuzzinessSlider.setEnabled(false);        epsilonSlider.setEnabled(false);    }
    else if (e.getSource() == monitor)    {.
        display.setImage2(task.getRankedMFImage(0));
        int percentage = (int)(100*task.getPosition()/task.getSize());
        progressBar.setValue(percentage);
        infoLabel.setText("Step "+task.getPosition()+" of "+task.getSize());
        if (task.isFinished())        {        start.setEnabled(true);
            start.setText("Start clustering");        numClustersSlider.setEnabled(true);
            maxIterationsSlider.setEnabled(true);        fuzzinessSlider.setEnabled(true);
            epsilonSlider.setEnabled(true);        monitor.stop();
            try        {
                ImageIO.write(task.getRankedMFImage(0),"PNG",new File("clustered.png"));
            } catch (IOException e1)        {e1.printStackTrace();        }        }    }
    if(e.getSource()== mnuOpenFile) {        fileChooser.showOpenDialog(this);
        f = fileChooser.getSelectedFile();        fl = f+"";        setVisible(false);
        new FuzzyCMeansImageClusteringApp(fl);    }
    if(e.getSource()== mnuHelp) {
        JOptionPane.showMessageDialog(this, "This applications shows a usage of the
        FuzzyCMeansImageClustering Algorithm.\nThe original and clustered image are shown,
        side-by-side and synchronized.\nInput data is read from a file and at the end of the
        task the different\nordered clustering results can be displayed as well as saved.",
        "About The Application", JOptionPane.INFORMATION_MESSAGE);    }
    if(e.getSource()== mnuAbout) {
        JOptionPane.showMessageDialog(this, "Program Designed & Developed By:\n\nAvinash
        Kumar Sharma\nReg No. 06MCA027\n3rd Year MCA 'B' Batch", "About The Developer",
        JOptionPane.INFORMATION_MESSAGE);    }
    if(e.getSource()== mnuExit)    {    int result= JOptionPane.showConfirmDialog(this,
    "Vous etez sur de fermer l'application ?", "Confirmer Fermeture",
    JOptionPane.YES_NO_OPTION);
        if(result== JOptionPane.YES_OPTION)    {
            JOptionPane.showMessageDialog(this, "Thanks for using this program.",
            "Appreciation", JOptionPane.INFORMATION_MESSAGE);        System.exit(0);    }    } }

```



## 6. windowClosing():

Cette méthode permet de fermé l'application.

```
public void windowClosing(WindowEvent w)
{
    JOptionPane.showMessageDialog(this, "Thanks for using this program.", "From the
Developer", JOptionPane.INFORMATION_MESSAGE);
    dispose();    }
```

## 7. Principal():

Le point d'entrée de l'application, qui aura besoin d'un nom de fichier d'image d'entrée et un nom de sortie du fichier image.

```
public static void main(String[] args)
{
    new FuzzyCMeansImageClusteringApp(fl);    }
```