

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'ELECTRONIQUE

**Mémoire de Fin d'Études  
de MASTER ACADIMIQUE**

Domaine : Sciences et Technologies  
Filière : Génie électrique

Spécialité : Électronique biomédicale

Présenté par  
MEZIANI ROZA  
BOUNIF KATIA

**Thème**

**Segmentation d'images couleur par seuillage  
d'histogrammes**

Mémoire soutenu le 13/07/2017

Président

Rapporteur

Examineur

devant le jury d'examen compose de :

Mr:LAZRIM

Mr:HAMMOUCHE. K

Mr:OUALOUCHE .F

Mr:HAMMOUCHE Kamal

Promotion: 2016/2017

# *Remerciement*

Nos remerciements les plus profonds vont à Monsieur **Hammouche Kamal** de nous avoir proposé ce thème et pour son assistance précieuse et quotidienne.

Nous tenons aussi à remercier les membres du jury qui, par leurs connaissances, ont acceptés de juger notre travail.

Nous tenons à remercier toute personne ayant participé de près ou de loin au bon déroulement de notre travail.

Un immense merci à tous nos amis, qui par leur soutien moral nous ont aidé à finir notre travail malgré les obstacles rencontrés.

Finalement un grand merci à l'ensemble des enseignants du département électronique et du département d'automatique qui nous ont énormément transmis tout au long de notre cursus.



# DEDICACES

*A cœur vaillant rien d'impossible*

*A conscience tranquille tout est accessible*

*Quand il y a la soif d'apprendre*

*Tout vient à point à qui sait attendre*

*Quand il y a le souci de réaliser un dessein*

*Tout devient facile pour arriver à nos fins*

*Malgré les obstacles qui s'opposent*

*En dépit des difficultés qui s'interposent*

*Les études sont avant tout*

*Notre unique et seul atout.*

***A mes très chers parents***

*Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices.*

*Que ce modeste travail, soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait d'incroyable pour moi.*

*Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.*

***A mon très chers frère, a mes très chers sœurs***

*Aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous.*

*Je vous dirais tout simplement, un grand merci, je vous aime.*

***A mes très chers ami(e)s***

*En témoignage de l'amitié sincère qui nous a lié et des bons moments passés ensemble.*

***A tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.***

***Je vous dédie ce travail...***

***R.MEZIANI***



## Liste des figures

Figure 1.1 : Segmentation d'une tumeur du cerveau

Figure 1.2 : Approches de la segmentation d'images

Figure 1.3 : Image segmentée par l'approche contour

Figure 1.4 : Segmentation d'image par division /fusion

Figure 1.5 : Histogrammes d'images en niveaux de gris

Figure 1.6 : Décomposition d'une image couleur en composantes RVB

Figure 2. 1: Taxinomie des méthodes d'optimisation métaheuristique

Figure 2.2 : Essaim de poissons, essaim d'oiseaux

Figure 2.3 : Lucioles

Figure 2.4 : Principe générale de l'algorithme des lucioles

Figure 2.5 : Œuf coucou dans un nouveau nid

Figure 2.6 : Principe d'un algorithme évolutionnaire

Figure 2.7 : Algorithme génétique

Figure 2.8 : Structure de l'algorithme génétique

Figure 2.9 : Algorithme de l'évolution différentielle

Figure 3.1 : Image test 1

Figure 3.2 : Image test 2

Figure 3.3 : Image test 3

Figure 3.4 : Résultats de la segmentation de l'image 1 par l'algorithme DE

Figure 3.5 : Résultats de la segmentation de l'image 2 par l'algorithme DE

Figure 3.6 : Résultats segmentation de l'image 3 par l'algorithme DE

### **Liste des tableaux**

Tableau 3.1 : Paramètres de l'algorithme DE

Tableau 3.2 : Résultats du seuillage par l'algorithme DE

Tableau 3.3 : Résultats du seuillage des images tests basés sur la recherche exhaustive

## **Sommaire**

|                                    |          |
|------------------------------------|----------|
| <b>Introduction générale .....</b> | <b>1</b> |
|------------------------------------|----------|

### **Chapitre I :**

|                                                                              |           |
|------------------------------------------------------------------------------|-----------|
| <b>1.1.Introduction .....</b>                                                | <b>2</b>  |
| <b>1.2.Définition de la segmentation d'image .....</b>                       | <b>2</b>  |
| <b>1.3. Domaines d'utilisation de la segmentation d'image .....</b>          | <b>3</b>  |
| <b>1.4.Techniques de segmentation .....</b>                                  | <b>3</b>  |
| <b>1.4.1 Approche contour .....</b>                                          | <b>5</b>  |
| <b>1.4.2. Approche région .....</b>                                          | <b>5</b>  |
| <b>1.4.2.1.. Croissance des régions .....</b>                                | <b>5</b>  |
| <b>1.4.2.2. Division des régions .....</b>                                   | <b>6</b>  |
| <b>1.4.2.3. Division /fusion .....</b>                                       | <b>6</b>  |
| <b>1.4.2.4. Classification.....</b>                                          | <b>6</b>  |
| <b>1.5. Segmentation d'image par seuillage d'histogrammes .....</b>          | <b>7</b>  |
| <b>1.5.1. Définition du seuillage .....</b>                                  | <b>7</b>  |
| <b>1.5.2. Seuillage global .....</b>                                         | <b>8</b>  |
| <b>1.5.2.1. Méthode d'Otsu .....</b>                                         | <b>10</b> |
| <b>1.6. Segmentation d'images couleur par seuillage d'histogrammes .....</b> | <b>11</b> |
| <b>1.7. Complexité algorithmique .....</b>                                   | <b>14</b> |
| <b>1.8. Conclusion .....</b>                                                 | <b>15</b> |

## **Chapitre II :**

|                                                                  |    |
|------------------------------------------------------------------|----|
| 2.1. Introduction .....                                          | 16 |
| 2.2. Problème d'optimisation.....                                | 16 |
| 2.3. Les métaheuristiques d'optimisation .....                   | 16 |
| 2.3.1. Les métaheuristiques à solution unique .....              | 17 |
| 2.3.2. Les Métaheuristiques à solutions multiples .....          | 18 |
| 2.3.2.1. Les algorithmes inspirés de la nature .....             | 18 |
| a. Méthode d'optimisation par essaim particulier .....           | 18 |
| b. L'algorithme des essaims de lucioles (Firfly algorithm) ..... | 19 |
| c. Algorithme de la recherche Coucou (CS) .....                  | 22 |
| d. Autres algorithmes inspirés de la nature .....                | 24 |
| 2.3.2.2. Les algorithmes évolutionnaires .....                   | 24 |
| 2.3.2.3. Algorithmes génétiques .....                            | 25 |
| 2.3.2.4. Algorithme de l'évolution différentielle (DE) .....     | 26 |
| 2.3.3. Conclusion.....                                           | 29 |

## **Chapitre III :**

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| 3.1. Introduction .....                                                         | 30 |
| 3.2. Méthode de segmentation adoptée pour la segmentation d'images couleur..... | 30 |
| 3.3. Tests et résultats .....                                                   | 31 |
| 3.4. Comparaison des résultats.....                                             | 36 |
| 4. Conclusion .....                                                             | 37 |
| 5. Conclusion générale .....                                                    | 38 |

# Introduction générale

---

La segmentation est une étape nécessaire dans le processus de traitement et d'analyse d'images. Elle correspond au partitionnement de l'image en un ensemble de classes ou de régions en se basant sur des critères spectraux ou spatiaux (comme l'information couleur, les relations de voisinage entre les pixels dans l'image...). Ces dernières années ont été marquées par la publication d'un grand nombre d'articles dans cette thématique. Les domaines d'application sont très variées : la télédétection, les applications militaires, l'imagerie médicale etc.

Le seuillage (thresholding en Anglais) représente un outil largement utilisé dans la segmentation d'images pour extraire des objets de leurs fonds en fonction d'un seuil. . Tout problème de seuillage consiste alors à rechercher la valeur du seuil. La plus part des méthodes de seuillage déterminent le seuil en optimisant une fonction objective, elles sont basées sur l'histogramme monodimensionnel de l'image. Cependant, ces méthodes ne tiennent pas en compte des relations spatiales entre les pixels.

Nous proposons dans ce travail d'étudier une méthode de seuillage d'histogrammes global qui est la methode d'Otsu, et de l'appliquer dans la recherche de plusieurs. Cependant, toutes ces extensions nécessitent un temps de calcul très élevé. Pour remédier a ce problème nous proposons de déterminer les seuils en utilisant la méthode d'optimisation métaheuristique a évolution différentielle ( DE).

Nous avons reparti notre travail en trois chapitres :

Le premier chapitre donne des notions sur les techniques de segmentation d'images.

Le deuxième chapitre est consacré à la présentation d'un état de l'art sur les techniques d'optimisation métaheuristique.

Dans le chapitre trois nous présenteront les tests et résultats de l'algorithme DE dans le seuillage d'histogrammes (monodimensionnels).

Une conclusion générale clôturera le mémoire.

## 1.1. Introduction

Le traitement d'image est un processus comprenant plusieurs étapes telles que l'acquisition de l'image, le prétraitement, la segmentation et l'extraction des informations.

La segmentation est l'étape fondamentale car elle conditionne le bon déroulement des étapes ultérieures. Elle a pour objectif d'extraire les objets contenus dans l'image.

Dans ce chapitre, nous présenterons une brève revue sur les différentes approches de la segmentation d'images et nous porterons l'accent sur les méthodes de seuillage d'histogrammes.

## 1.2. Définition de la segmentation d'image

La segmentation est définie comme étant un processus de partitionnement de l'image en régions homogènes ou chacune d'elles regroupe un ensemble de pixels présentant des propriétés communes (niveau de gris, couleur, texture). D'un point de vue algorithmique, la segmentation consiste à attribuer à chaque pixel de l'image un label d'appartenance à une région donnée. Elle est définie mathématiquement par Zucker de la manière suivante [Zucker, 1976] :

Segmenter une image  $I$  en  $N$  régions, revient à la partitionner en  $N$  sous ensembles  $R_1, R_2, \dots, R_N$  tels que :

$$1. I = \bigcup_i R_i \tag{I.1}$$

2.  $R_i$  est constituée de pixels connexes  $\forall i$ .

3.  $P(R_i) = \text{vrai} \forall i$ .

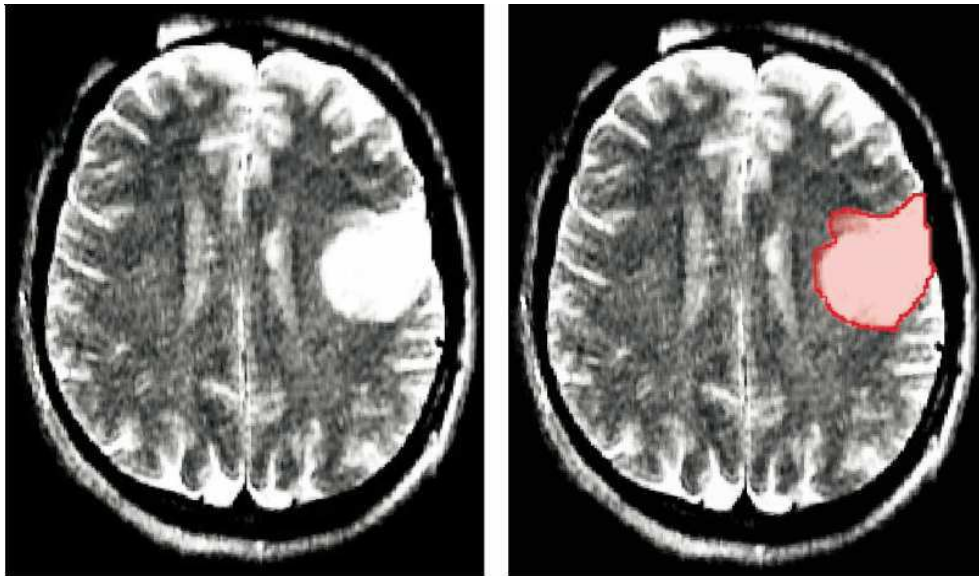
4.  $P(R_i \cup R_j) = \text{faux}$  pour tous  $i, j$ ,  $R_i$  et  $R_j$  étant adjacentes dans  $I$ .

La première condition indique que chaque pixel de l'image doit appartenir à une région  $R_i$  et l'union de toutes les régions forme l'image entière. La deuxième condition est relative à la structure des régions, elle définit une région comme un ensemble de pixels qui doivent être connexes. La troisième condition exprime que chaque région doit respecter un prédicat d'uniformité. La dernière condition implique que la non réalisation de ce même prédicat pour la réunion de deux régions adjacentes.

### 1.3. Domaines d'utilisation de la segmentation d'image

La segmentation d'image est très utilisée dans différents domaines. On peut citer a titre d'exemples la biométrie (reconnaissance des visages, des empreintes, de l'iris, etc....).

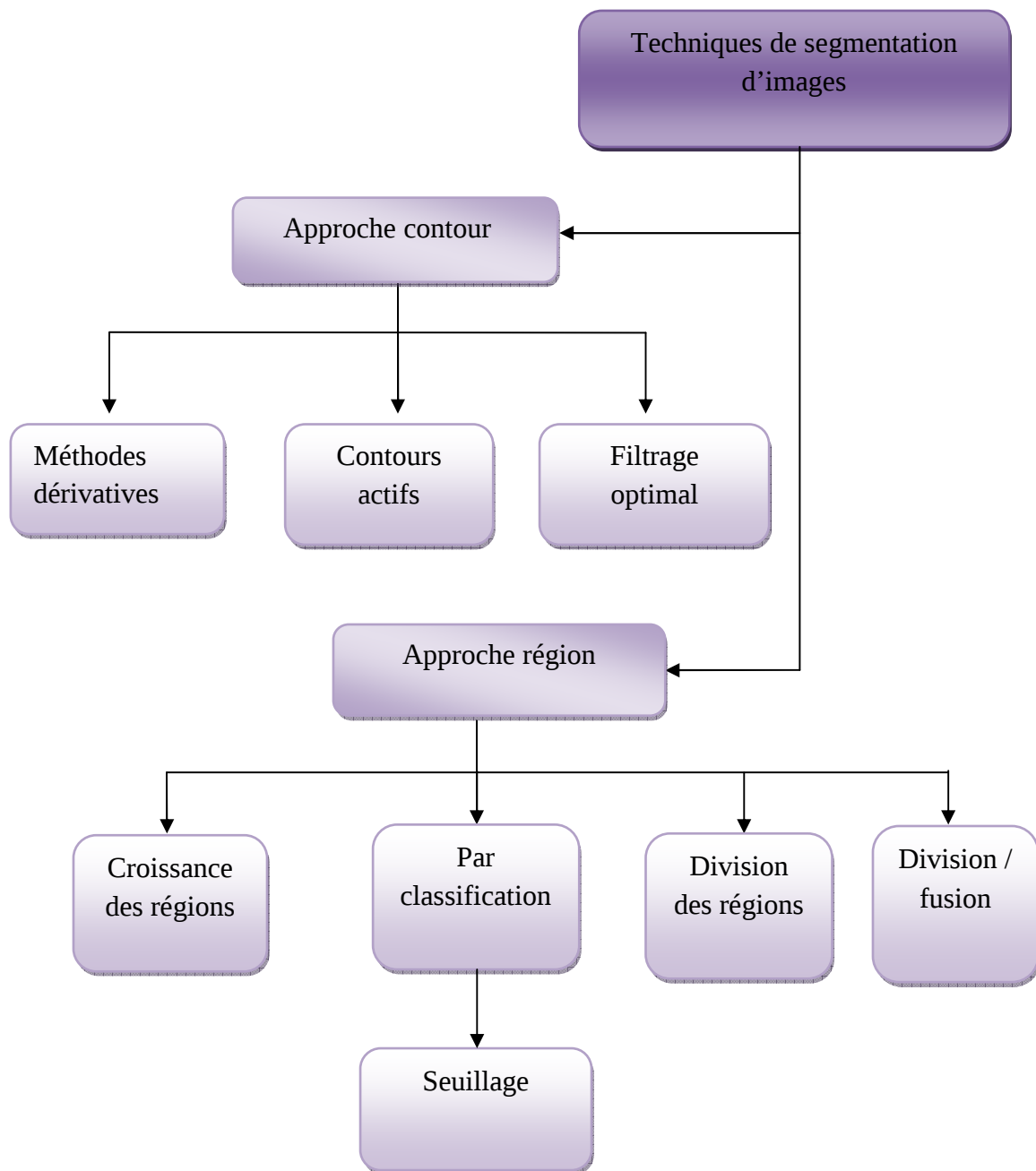
Dans le domaine médical la segmentation est très sollicitée pour la détection des tumeurs du cerveau par exemple (Fig.1.1).



**Fig.1.1** : segmentation d'une tumeur du cerveau

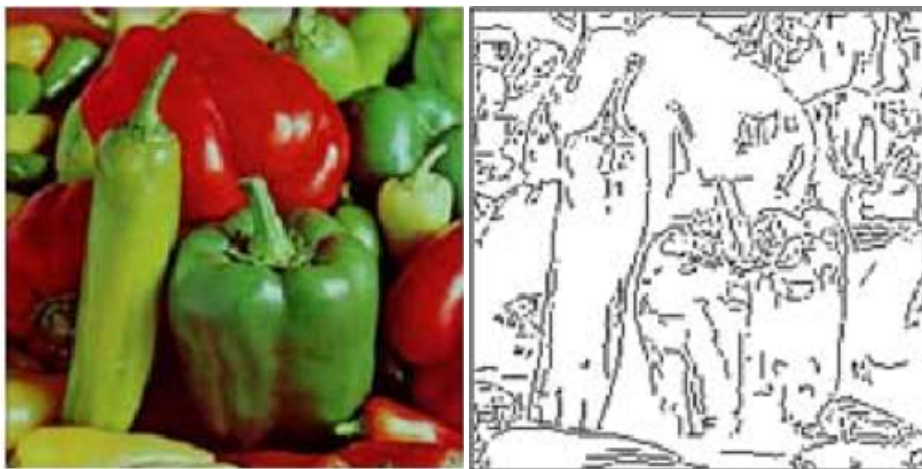
### 1.4. Techniques de segmentation

Une bonne méthode de segmentation est celle qui permettra d'arriver à une meilleure interprétation de l'image segmentée. On distingue deux grandes approches de segmentation : contour et région. La figure (1.2) donne un résumé sue ces différentes approches.

**Fig. 1.2 :** Approches de la segmentation d'images

### 1.4.1 Approche contour

De façon générale, un contour est défini comme étant la frontière entre deux régions. La détection du contour est donc équivalente à la détection de la discontinuité entre les régions en cherchant les zones de variations significatives d'intensité lumineuse ou de couleur dans l'image. La figure (1.3) montre un exemple d'image segmentée par l'approche contour. Il existe plusieurs méthodes de segmentation basées sur l'approche contour et qu'on peut regrouper en trois catégories : les méthodes dérivatives, par filtrage optimal et les contours actifs [2].



**Fig1.3** : image segmentée par l'approche contour

### 1.4.2. Approche région

L'approche région cherche à regrouper les pixels ayant les mêmes caractéristiques en régions homogènes. Elle se caractérise par la mesure d'uniformité des régions construites dans l'image. Ces régions sont construites en évaluant la similarité entre les pixels et ceux d'une même région. Le problème fondamental associé à cette approche est lié à la définition du critère d'homogénéité utilisé dans le processus de segmentation de l'image. On distingue alors quatre types de méthodes.

#### 1.4.2.1. Croissance des régions

Aussi appelée agrégation de pixels, ce type de méthodes consiste à choisir initialement un ensemble de pixels comme des germes ou noyaux des régions à partir desquels la croissance des régions s'effectuera. La croissance des régions s'effectue par agrégation des nouveaux pixels aux germes initiaux. Elle consiste à fusionner successivement aux régions

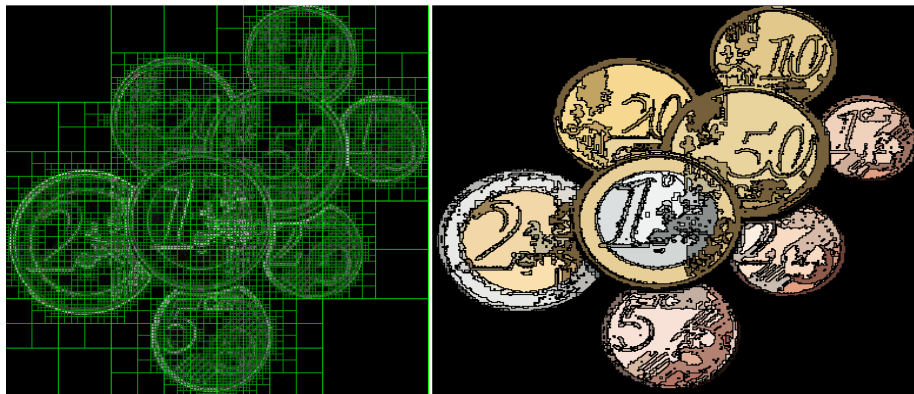
les pixels voisins qui sont similaires à condition que les régions fusionnées continuent à vérifier le critère d'uniformité. Un critère d'uniformité ou d'homogénéité est défini par exemple par la variance des niveaux de gris.

#### 1.4.2.2. Division des régions

Ces méthodes opèrent en découpant l'image principale en régions de tailles plus petites suivant un critère d'hétérogénéité. Le découpage s'arrête quand il n'y a plus de régions inhomogènes. La division de l'image est généralement représentée par une structure géométrique bien particulière telle que la structure à arbre quaternaire "quadtree".

#### 1.4.2.3. Division/fusion

La segmentation par division/fusion regroupe les deux types d'approches précédents. Elle permet de produire automatiquement une partition initiale de l'image en petites régions, qui vont ensuite, se croître en se fusionnant. La partition initiale (Split) est réalisée en divisant récursivement l'image en régions de tailles identiques si un certain critère d'homogénéité n'est pas vérifié. La phase de regroupement (Merge) consiste à regrouper les régions ainsi produites en régions plus importantes et qui satisfont un autre critère d'homogénéité [3]. La figure (1.4) nous donne les deux étapes qui sont la division en régions et la fusion d'une image couleur.



**Fig. 1.4 :** Segmentation d'image couleur par division/ fusion.

#### 1.4.2.4. Classification

La classification est une opération de base d'analyse des données, elle a pour but de regrouper les pixels caractérisés par un ensemble d'attributs en classes. Les classes sont produites de telle sorte que les pixels d'une même classe soient les plus similaires possibles et les pixels de

deux classes distinctes soient les plus différents possibles. Une région est alors formée de pixels connexes appartenant à une même classe.

La classification peut se faire de deux manières : la première, dite supervisée, suppose l'existence de certains pixels (prototypes) dont l'appartenance aux classes est connue a priori. Cette approche nécessite une image de référence ou l'intervention d'un expert, elle est souvent utilisée pour tester la technique de classification elle-même ou la pertinence des attributs.

Dans le cas où l'on n'a aucune connaissance sur l'appartenance des pixels aux classes, on parle alors de classification non supervisée ou clustering. Comme méthodes de classification non supervisée on peut citer l'algorithme K-means l'algorithme FCM (algorithme Fuzzy C-means), ainsi que l'algorithme d'estimation-Maximisation (EM).

Lorsque les objets qui composent l'image peuvent être distingués par leurs niveaux de gris seulement, la segmentation par classification des pixels peut être alors abordée par des techniques de seuillage.

### 1.5. Segmentation d'image par seuillage d'histogrammes

La segmentation par seuillage constitue un cas particulier de la segmentation par classification. C'est une technique de segmentation très populaire à cause de sa facilité de mise en œuvre et sa rapidité. Elle permet de répartir les pixels en classes en fonction de leurs niveaux de gris. Les classes sont alors délimitées par des niveaux de gris appelés seuils.

#### 1.5.1. Définition du seuillage

Soient  $\{0, 1, 2, \dots, L-1\}$  l'ensemble des niveaux de gris d'une image  $I$  composée de  $N$  pixels,  $l(x,y)$  la luminance (niveau de gris) d'un pixel de coordonnées  $(x,y)$ .  $L-1$  étant le niveau de gris maximal, souvent  $L=256$ .

La segmentation par seuillage est une opération qui consiste à répartir les pixels en  $K$  classes  $(C_1, C_2, \dots, C_k)$  à partir d'un ensemble de seuils  $T=\{t_1, t_2, \dots, t_{K-1}\}$ . Par convenance, on utilise deux autres seuils  $t_0=0$  et  $t_K=L-1$ . Un pixel de niveau de gris  $l(x,y)$  est affecté à la classe  $C_k$  si  $t_k \leq l(x,y) < t_{k+1}$  avec  $k=0, 1, 2, \dots, K-1$ .

Pour une image comportant des objets clairs sur un fond sombre, le seuillage se résume à choisir un seul seuil  $t$  et à affecter à chaque pixel de l'image le label objet( $C_1$ ) ou fond( $C_2$ ) selon le test suivant :

Si  $l(x, y) < t$  alors le pixel est étiqueté « fond », sinon le pixel est étiqueté « objet »

Dans ce cas, on parle du seuillage tout court (bilevel thresholding) et l'image segmentée ainsi obtenue est appelée image binaire.

Si par contre le nombre de seuils recherchés est supérieur à 1, on parle de multi seuillage ou multi-level thresholding.

Lorsque les seuils  $t_k$  sont indépendants de la position des pixels, le seuillage est dit global.

Dans certains cas, (non uniformité de l'éclairage de l'image, présence de différents objets ayant des dynamiques de luminances différentes), la binarisation au moyen d'un même seul seuil pour chaque point de l'image reste inadéquate. Dans ce cas, les méthodes de binarisation dites locales ou adaptatives sont mieux adaptées. Celles-ci consistent à définir un seuil  $t(x, y)$  pour chaque pixel. Le calcul du seuil  $t(x, y)$  local est déterminé en prenant en compte les niveaux de gris des pixels voisins. On s'intéresse dans notre travail au seuillage global.

### 1.5.2 Seuillage global

Les méthodes de seuillage global reposent sur l'exploitation de l'histogramme de toute l'image. L'histogramme est une fonction discrète notée  $p(i)$ , qui caractérise la distribution des niveaux de gris. Il est défini par la probabilité d'apparition du niveau de gris  $i$  ( $i=0,1,\dots, L-1$ ) tel que :

$$p(i) = \frac{h(i)}{N} \quad (1.2)$$

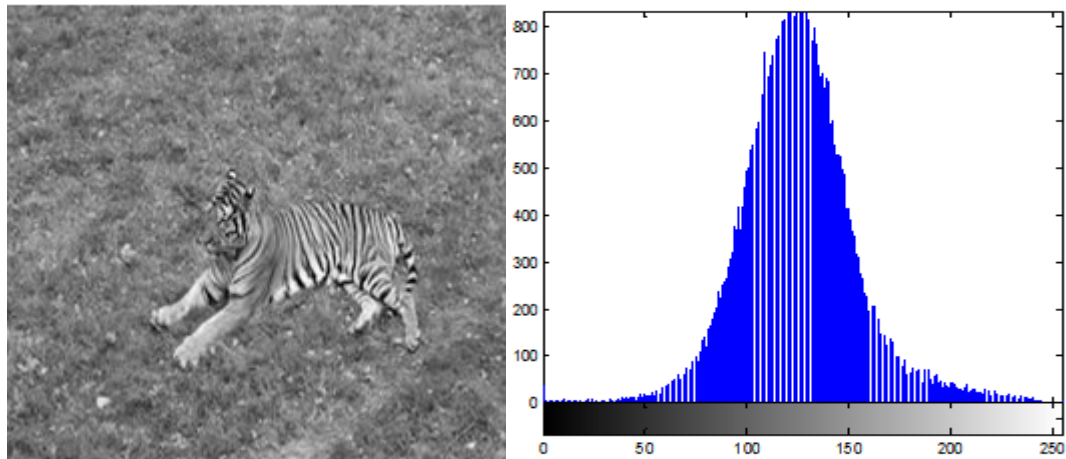
$h(i)$  étant le nombre de pixels ayant le niveau de gris  $i$  et  $N$  le nombre total de pixels dans l'image.

A partir de l'allure de l'histogramme, on peut déduire le nombre de classes ainsi que la position relative des seuils (Fig. 1.5). Un histogramme est unimodal s'il est formé d'un seul pic représentant les pixels de l'objet ou ceux du fond. Il est bimodal s'il est caractérisé par deux modes séparés par une vallée, il indique l'existence d'un objet sur un fond. Un histogramme multimodal comportant plus de deux modes, séparés par les vallées, indique la présence de plusieurs classes dans une image (chaque mode correspond à une classe).

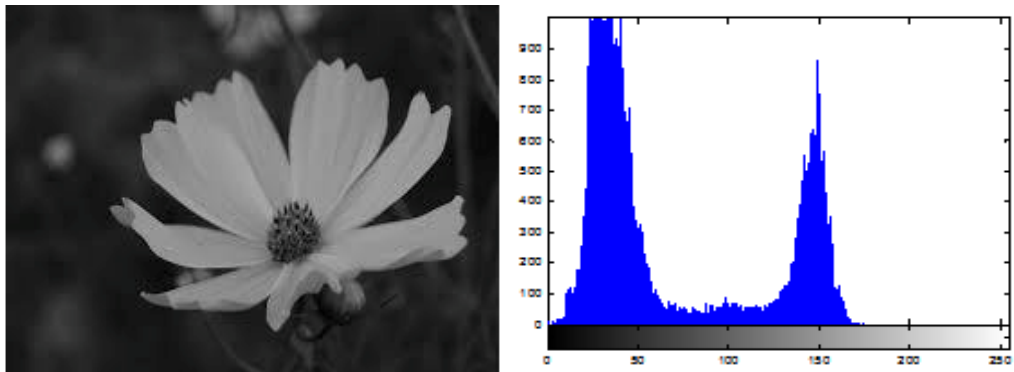
Les seuils qui séparent les classes doivent être localisés dans les vallées qui déterminent les modes de l'histogramme.

En pratique, il est rare de trouver un histogramme qui présente des modes bien distincts. En effet, la plupart des images présentent des histogrammes bruités caractérisés par des modes non discernables. La question qui se pose donc est “comment calculer les seuils  $t_k$  ? “

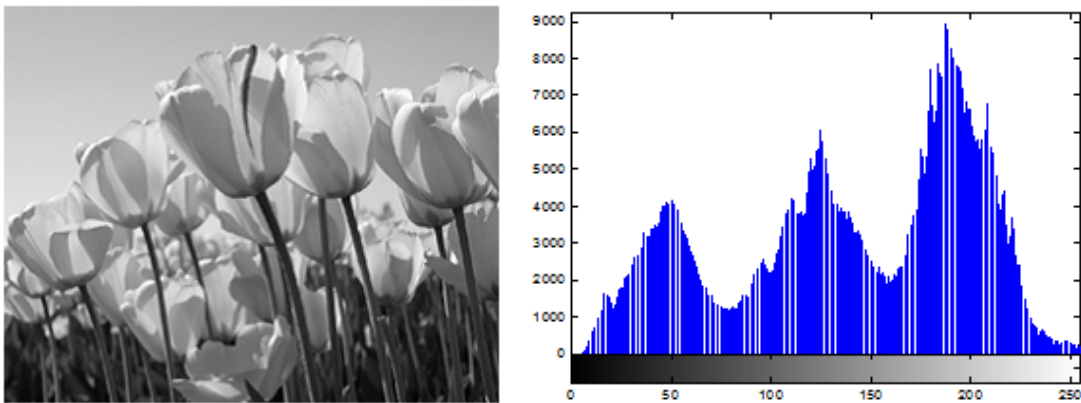
Plusieurs techniques ont été proposées. La plus part d'entre elles permettent de trouver les seuils optimaux sans tenir compte d'aucune hypothèse sur la forme de l'histogramme. Elles sont généralement basées sur l'optimisation d'une fonction objective. La fonction objective la plus connue est celle proposée par Otsu [79].



(a)



(b)



(c)

**Fig. 1.5** Histogrammes d'images en niveau de gris. (a): Unimodal, (b): Bimodal et (c): Multimodal.

#### 1.5.2.1. Méthode d'Otsu

Elle est considérée comme la méthode de référence dans le domaine du seuillage

d'histogrammes. Dans cette méthode, le seuillage consiste à séparer les pixels d'une image en deux classes  $C_1$  (fond),  $C_2$  (objet) à partir d'un seuil  $t$ . La classe « fond » regroupe tous les pixels ayant un niveau de gris inférieur au seuil  $t$  alors que la classe « objet » contient tous les pixels de niveau de gris supérieur à  $t$ .

$$C_1 = \{0, 1, \dots, t\} \text{ et } C_2 = \{t + 1 \dots L - 1\}$$

Le calcul du seuil  $t$  est effectué en maximisant la variance inter-classes ou bien en minimisant la variance intra-classes.

Soit  $\sigma_w^2(t)$  la variance intra-classes,  $\sigma_B^2(t)$  la variance inter-classes telles que :

$$\sigma_B^2(t) = P_1 P_2 (\mu_2 - \mu_1)^2 \text{ et } \sigma_w^2(t) = \sum_{i=0}^t P_1 (i - \mu_1)^2 + \sum_{i=t+1}^{L-1} P_2 (i - \mu_2)^2$$

Soit  $\sigma_T^2$  la variance totale de l'image telle que:  $\sigma_T^2 = \sum_{i=1}^{L-1} P_i (i - \mu)^2$ . Elle est liée aux deux autres variances comme suit:

$$\sigma_T^2 = \sigma_B^2(t) + \sigma_w^2(t) \quad (I.3)$$

$\mu_1(t)$  et  $\mu_2(t)$  représentent les niveaux de gris moyens des classes  $C_1$  et  $C_2$  respectivement:

$$\mu_1(t) = \sum_{i=1}^t i \cdot \frac{P_i}{P_1}, \quad \mu_2(t) = \sum_{i=t+1}^{L-1} i \cdot \frac{P_i}{P_2}, \quad \mu = \sum_{i=1}^{L-1} i \cdot P_i \quad (I.4)$$

$P_1(t)$  et  $P_2(t)$  représentent respectivement les probabilités a priori des classes  $C_1$  et  $C_2$  tels que :

$$P_1(t) = \sum_{i=1}^t P_i, \quad P_2(t) = \sum_{i=t+1}^{L-1} P_i \quad (I.5)$$

avec  $P_1(t) + P_2(t) = 1$ .

La méthode d'Otsu consiste déterminer le seuil optimal  $t^*$  en maximisant la variance interclasse:

$$t^* = \text{Arg max } \sigma_B^2(t) \quad (I.6)$$

Dans le cas du multi seuillage, la méthode d'Otsu peut être étendue au calcul d'un ensemble de seuils  $T = (t_1, t_2, \dots, t_{k-1})$  en maximisant la variance interclasse

$$T^* = \text{Argmax } J(T) \quad (I.7)$$

$$\text{où } J(T) = \sum_{k=1}^K P_k (\mu_k - \mu)^2 \quad (\text{I.8})$$

avec  $P_k$  et  $\mu_k$  la probabilité et la moyenne de la classe  $C_k$  tels que :

$$P_k = \sum_{i=t_k-1}^{t_k-1} P_i \text{ et } \mu_k = \sum_{i=t_k-1}^{t_k-1} i P_i \quad (\text{I.9})$$

D'autres méthodes de seuillages, basées sur l'entropie de l'histogramme, ont été proposées. On parle alors de seuillage entropique. Parmi ces méthodes, on peut citer les méthodes de Kapur, de cross entropie, d'entropie de Renyi etc. [2].

### 1.6. Segmentation d'images couleur par seuillage d'histogrammes

Les méthodes de seuillages ont été initialement développées pour des images en niveaux de gris. Récemment, ces méthodes ont été étendues aux images couleur.

Contrairement aux images en niveaux de gris, une image couleur  $I$  est composée de  $N$  pixels ou chaque pixel est caractérisé par trois composantes au lieu d'une seule (niveau de gris).

Les composantes les plus utilisées sont le rouge (R), le vert (V) et le bleu (B). Dans ce cas on parle d'images RVB. D'autres composantes couleur ont été proposées (voir annexe).

La performance d'une procédure d'une segmentation d'images dépend du choix de l'espace couleur. Plusieurs auteurs ont tenté de déterminer les espaces couleurs qui sont les mieux appropriés à leurs problèmes spécifiques de segmentation d'image couleurs.

La figure 1.6 montre une image couleur décomposée en trois images ou chaque image représente l'une des trois composantes R, V et B.

L'histogramme de chacune de ces images composantes présente des allures différentes.

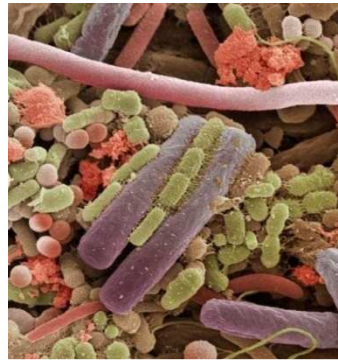
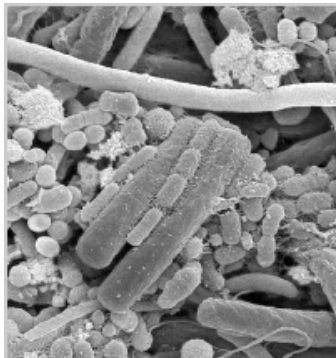
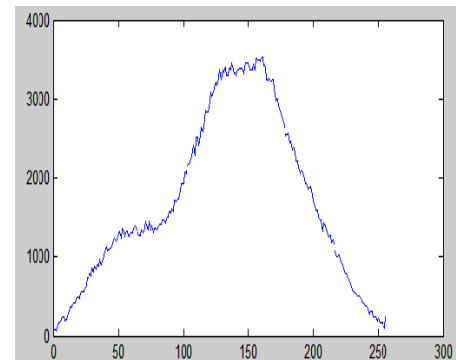


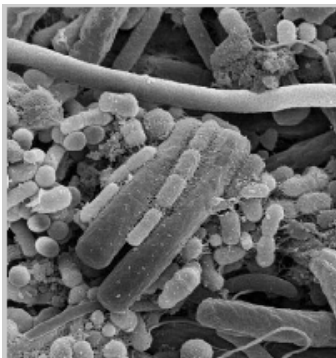
Image originale



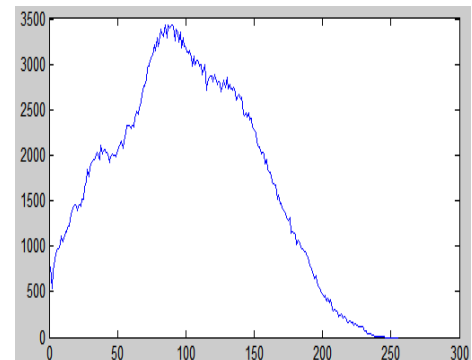
Composante R



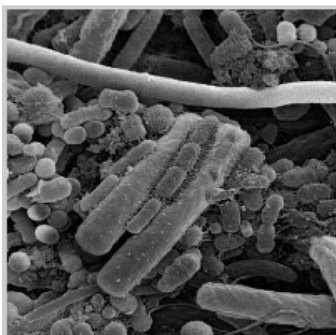
Histogramme de la composante R



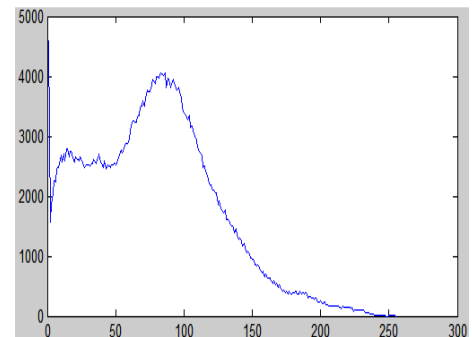
Composante V



Histogramme de la composante V



Composante B



Histogramme de la composante B

**Fig. 1.6 :** Décomposition d'une image couleur en composantes R, V, B ainsi que leurs histogrammes

La segmentation d'images couleur consiste à partitionner l'image en régions ayant des couleurs homogènes ou les pixels d'une même région partagent des propriétés colorimétriques similaires. La segmentation d'une image couleur basée sur des procédures de classification peut être abordée de deux manières. La première consiste à regrouper les pixels en classes. La seconde consiste à détecter les modes de l'histogramme multidimensionnel  $p(r, v, b)$  où

$$p(r, v, b) = \frac{h(r, v, b)}{N}. \quad (\text{I.10})$$

$h(r, v, b)$  étant le nombre de pixels ayant un niveau de rouge  $r$ , un niveau de vert  $v$  et un niveau de bleu  $b$  avec  $r, v, b = 1, \dots, L - 1$ .

Cette méthode peut être complexe. Une alternative consiste à analyser les histogrammes marginaux  $p(r), p(v), p(b)$  en effectuant un seuillage de chaque histogramme en  $K$  classes.

Soient  $T^R = \{t_1^R, t_2^R, \dots, t_{K-1}^R\}$ ,  $T^V = \{t_1^V, t_2^V, \dots, t_{K-1}^V\}$  et  $T^B = \{t_1^B, t_2^B, \dots, t_{K-1}^B\}$  les seuils de chaque composante  $R, V, B$ .

Un pixel de coordonnées  $(x, y)$  de couleur  $r(x, y), v(x, y)$  et  $b(x, y)$  est affecté à la classe :

$$C_k^R \text{ si } t_k^R \leq r(x, y) \leq t_{k-1}^R \quad k = 0, 1, \dots, K - 1 \quad (\text{I.11})$$

$$C_k^V \text{ si } t_k^V \leq v(x, y) \leq t_{k-1}^V \quad k = 0, 1, \dots, K - 1 \quad (\text{I.12})$$

$$C_k^B \text{ si } t_k^B \leq b(x, y) \leq t_{k-1}^B \quad k = 0, 1, \dots, K - 1 \quad (\text{I.13})$$

$C_k^R, C_k^V$  et  $C_k^B$  sont respectivement les classes des composantes rouge, vert et bleu.

La construction de l'image couleur segmentée peut être obtenue par recombinaison des trois composantes segmentées. Le nombre de classes de l'image couleur segmentée sera donc au maximum égal à  $(3^K - 1)$ .

Dans le cas de la méthode d'Otsu, les seuils de chaque composante peuvent être déterminés indépendamment en maximisant :

$$J(T^R) = \sum_{k=1}^K P_k^R (\mu_k^R - \mu^R)^2, \quad (\text{I.14})$$

$$J(T^V) = \sum_{k=1}^K P_k^V (\mu_k^V - \mu^V)^2 \quad (\text{I.15})$$

$$\text{et } J(T^B) = \sum_{k=1}^K P_k^B (\mu_k^B - \mu^B)^2 \quad (\text{I.16})$$

ou d'une manière conjointe en maximisant  $J(T^R, T^V, T^B) = J(T^R) + J(T^V) + J(T^B)$ .

avec :  $P_k^{(R,V ou B)} = \sum_{i=t_{k-1}^{(R,V ou B)}}^{t_k^{(R,V ou B)}} P_i$  la probabilité de la classe  $C_k^{(R,V ou B)}$

$$\mu_k^{(R,V ou B)} = \sum_{i=t_{k-1}^{(R,V ou B)}}^{t_k^{(R,V ou B)}} i \cdot P_i \quad (I.17)$$

$\mu_k^{(R,V ou B)}$  est la valeur moyenne rouge, vert ou bleu de la classe  $C_k^{(R,V ou B)}$ .

$\mu^{(R,V ou B)}$  le niveau moyen de toute la composante rouge, vert ou bleu.

### 1.7. Complexité algorithmique

Plusieurs techniques ont été proposées pour le calcul d'un seul seuil puis étendues au problème du multiseuillage. Cependant, en pratique cette extension peut engendrer des temps de calculs prohibitifs. En effet la recherche d'un seul seuil d'une manière exhaustive est tout à fait possible. Cependant, cette recherche exhaustive devient prohibitive lorsque le nombre de seuils augmente. En effet il a été démontré que la complexité des calculs augmente exponentiellement lorsque le nombre de seuils augmente [Chang et al, 1995]. Pour  $K$  classes le nombre d'opérations nécessaires pour le calcul de  $K-1$  seuils est donné par :

$$N0 = \frac{(L+K)!}{L!K!} \quad (I.18)$$

Pour résoudre ce problème plusieurs techniques ont été proposées. Parmi lesquelles on trouve des techniques d'optimisation méta heuristiques [Horng, 2010].

### 1.8. Conclusion

Dans ce chapitre, nous avons brièvement présenté les différentes techniques de segmentation d'images. Nous nous sommes spécialement intéressés à la segmentation d'images couleur par seuillage d'histogrammes.

Les techniques de seuillage d'histogrammes sont nombreuses, plusieurs d'entre elles ont été proposées pour le calcul d'un seul seuil. En effet la recherche d'un seul seuil de manière exhaustive est tout à fait possible. Cependant cette recherche devient prohibitive lorsque le nombre de seuils augmente.

Pour résoudre ce problème plusieurs techniques ont été proposées, parmi lesquelles on trouve des techniques d'optimisation métaheuristiques.

## 2.1 Introduction

Le problème de la segmentation d'images par seuillages, s'exprime comme un problème d'optimisation, pouvant être NP-difficile et qui ne peut être résolue de manière exacte dans un temps raisonnable. Pour résoudre ce type de problème avec un temps de calcul raisonnable, de nouvelles techniques basées sur des métaheuristiques, qui sont pour la plupart inspirées de la nature ont fait l'objet de beaucoup de recherches.

Dans ce chapitre, nous introduisons un bref aperçu sur les méthodes classiques d'optimisation métaheuristiques, et nous détaillons l'algorithme à évolution différentielle (DE).

## 2.2 Problème d'optimisation

Un problème d'optimisation se définit comme la recherche, parmi un ensemble de solutions possibles  $S$  (appelé aussi espace de décision ou espace de recherche), d'une solution optimale  $x^*$  qui rendent minimale (ou maximale) une fonction mesurant la qualité de cette solution. Cette fonction est appelée fonction objectif ou fonction coût. Si l'on pose  $f : S \rightarrow R$  la fonction objectif à minimiser (respectivement à maximiser) à valeurs dans  $R$ , le problème revient alors à trouver l'optimum  $x^* \in S$  tel que  $f(x^*)$  soit minimal (respectivement maximal).

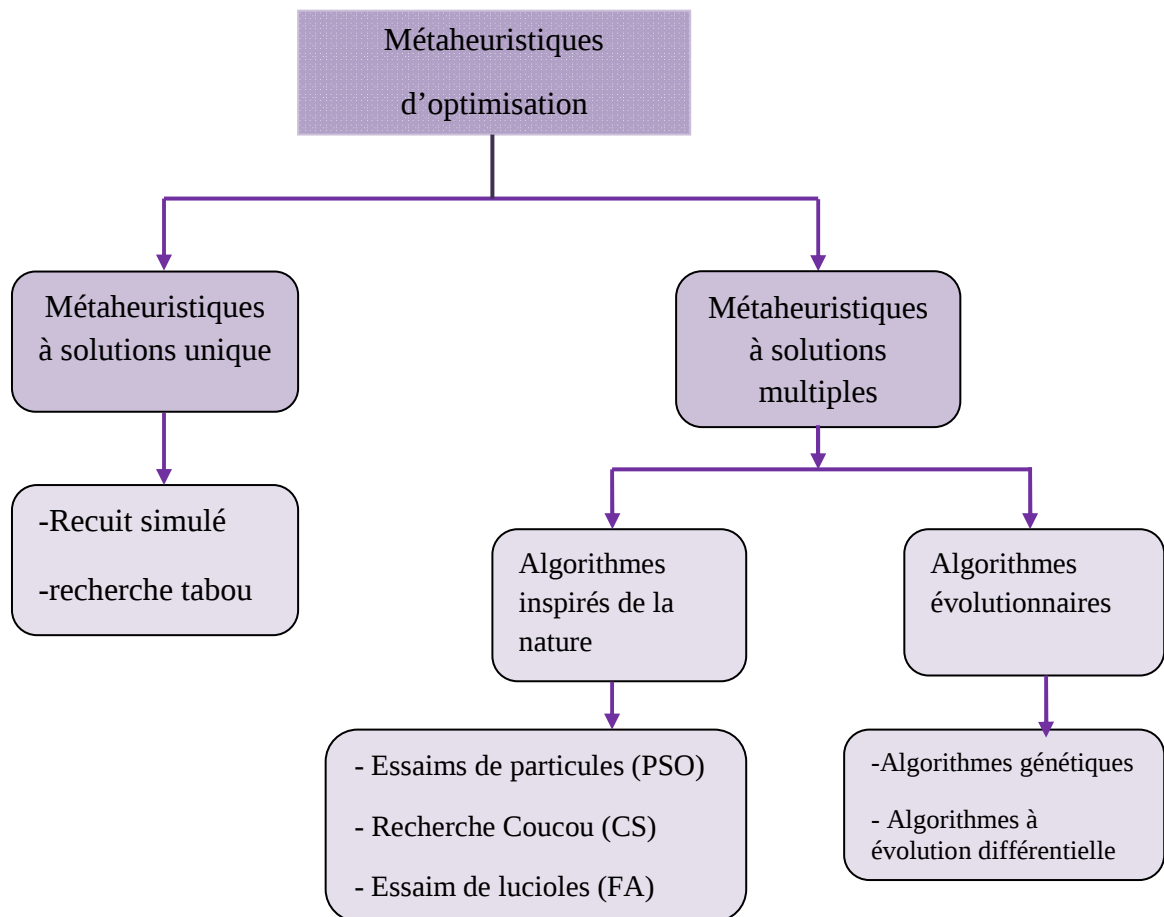
Lorsque l'on veut résoudre un problème d'optimisation, on recherche la meilleure solution possible à ce problème, c'est-à-dire l'optimum global. Cependant, il peut exister des solutions intermédiaires, qui sont également des optimums, mais uniquement pour un sous-espace restreint de l'espace de recherche : on parle alors d'optimums locaux.

## 2.3 Les métaheuristiques d'optimisation

Les métaheuristiques sont des procédures de haut niveau conçues pour résoudre des problèmes d'optimisation dits difficiles. Ce sont en général des problèmes aux données incomplètes, incertaines, bruitées ou confrontés à une capacité de calcul limitée. Les métaheuristiques ont rencontré le succès dans beaucoup de domaines. Cela découle du fait qu'elles peuvent être appliquées à tout problème pouvant être exprimé sous la forme d'un problème d'optimisation. [6]

Ces méthodes sont, pour la plupart, inspirées de la physique (recuit simulé), de la biologie (algorithmes évolutionnaires) ou de l'éthologie (essaims particuliers, colonies de fourmis).

Dans ce qui suit, nous présentons brièvement les principales métaheuristiques d'optimisation. Nous les classons en deux catégories : métaheuristiques à solution unique et métaheuristiques à population de solutions. La figure (2.1) suivante représente une taxinomie de quelques méthodes métaheuristiques existantes.



**Fig. 2.1 :** Taxinomie des méthodes d'optimisation métaheuristiques.

### 2.3.1 Les métaheuristiques à solution unique

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage.

De manière générale, les opérateurs de recherche locale s'arrêtent quand une solution localement optimale est trouvée, c'est-à-dire quand il n'existe pas de meilleure solution dans le voisinage. Mais accepter uniquement ce type de solution n'est bien sûr pas satisfaisant. Dans un cadre plus général, il serait alors intéressant de pouvoir s'échapper de ces minima locaux .Il

faut alors permettre à l'opérateur de recherche locale de faire des mouvements pour lesquels la nouvelle solution retenue sera de qualité moindre que la précédente.

Les méthodes les plus utilisées dans les problèmes de segmentation d'image sont généralement la méthode de descente, le recuit simulé et la recherche tabou [12].

### 2.3.2 Les Métaheuristiques à solutions multiples

À l'inverse des méthodes de recherche à solution unique, les métaheuristiques à population de solutions sont des méthodes qui font évoluer simultanément un ensemble d'individus (solutions) dans l'espace de recherche, où chacun profite de l'expérience du groupe, de manière directe ou indirecte. Ces méthodes sont principalement inspirées du vivant. On peut distinguer deux catégories de métaheuristiques à population : les algorithmes d'intelligence en essaim inspirés de l'éthologie et de la biologie et les algorithmes évolutionnaires inspirés de la théorie de l'évolution de C. Darwin

#### 2.3.2.1. Les algorithmes inspirés de la nature

##### a. Méthode d'optimisation par essaims particulière

La métaheuristique basée sur la méthode des essaims particuliers ("Particle Swarm Optimization", PSO) a été développée par Kennedy et al. en 1995 [Kennedy, et al, 1995]. Le principe de la méthode est tiré des comportements collectifs d'animaux comme le déplacement des bancs de poissons ou le vol des oiseaux (Fig2.2). En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une intelligence limitée et une connaissance seulement locale de sa situation dans l'essaim. Un individu de l'essaim n'a pour connaissance que la position et la vitesse de ses plus proches voisins. Chaque individu utilise donc, non seulement, sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement.

Kennedy et Eberhart se sont inspirés de ces comportements socio-psychologiques pour créer le PSO. L'algorithme de base de la P.S.O utilise une population appelée *essaim* de solutions possibles, elles-mêmes appelées *particules*. Ces particules sont placées aléatoirement dans l'espace de recherche. Chaque particule  $k$  est caractérisée par sa position  $x_k$  et sa vitesse  $v_k$ . A chaque itération, les particules se déplacent en prenant en compte leur meilleure position personnelle ( $Pbest_k$ ) mais aussi la meilleure position globale ( $Pbest_g$ ) de toute la population.

Dans les faits, la vitesse de chaque particule est mise à jour à partir de la formule suivante :

$$V_k(t+1) = w V_k(t) + C_1 (Pbest_k(t) - x_k(t)) + C_2 (x_k(t) - Pbest_g(t)) \text{ ou}$$

$V_k(t+1)$  et  $V_k(t)$  sont les vitesses de la  $k$ ème particule aux itérations  $t$  et  $t+1$ .

$Pbest_k(t)$  est la meilleure position personnelle de la  $k$ ème particule.

$Pbest_g(t)$  est la meilleure position de toutes les particules à itération  $t$ .

$x_k(t)$  est la position de la particule  $k$  à itération  $t$ .

$w$  est le coefficient d'inertie et  $C_1$  et  $C_2$  les coefficients d'accélération.

La position de chaque particule est ensuite ajustée comme suit :

$$x_k(t+1) = x_k(t) + V_k(t+1)$$

Cet algorithme a été appliqué pour seuiller les histogrammes d'une image couleur [19] [20].

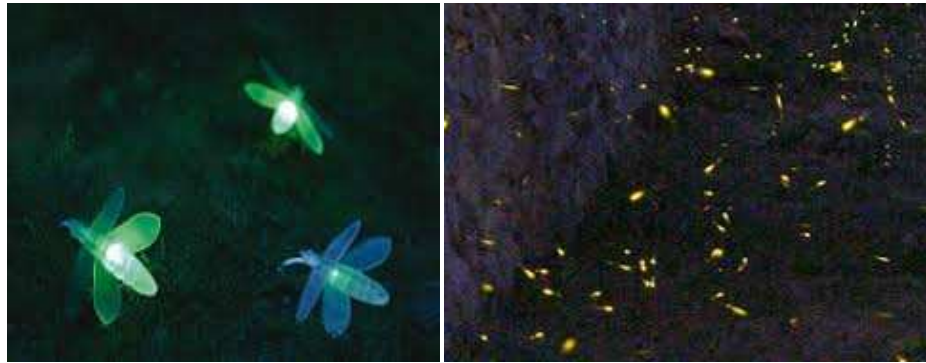
La fonction d'Otsu a été utilisée comme fonction objective.



**Fig.2.2 :** Essaim de poissons, essaim d'oiseaux

### **b. L'algorithme des essaims de lucioles (Firefly algorithm)**

Les lucioles (firefly en anglais) sont de petits coléoptères ailés, capables de produire une lumière clignotante froide pour une attraction mutuelle (Fig.2.3). Les lucioles ont un mécanisme de type condensateur, qui se décharge lentement jusqu'à ce qu'un certain seuil soit atteint, ils libèrent l'énergie sous forme de lumière. Le phénomène se répète de façon cyclique.



**Fig.2.3.** Lucioles.

L'algorithme des lucioles a été introduit par XS Yang en 2008 [14] [15] [16]. Sa source d'inspiration est basée sur l'émission de la lumière, absorption de la lumière et le comportement attractif mutuelle entre les lucioles.

Trois règles sont idéalisées dans l'algorithme des lucioles :

1. Toutes les lucioles sont unisexes, cela signifie qu'une luciole est attirée par une autre, indépendamment de son sexe.
2. L'attractivité est proportionnelle à la luminosité, ainsi pour deux clignotements des lucioles, celle qui est moins lumineuse se déplacera vers la plus lumineuse. L'attractivité est proportionnelle à la luminosité et toutes les deux diminuent lorsque leur distance augmente. S'il n'y a aucune autre source lumineuse, la luciole se déplacera au hasard.
3. La luminosité d'une luciole est déterminée par la fonction objective.

L'algorithme des lucioles, a été appliqué dans [10] [11] pour segmenter une image couleur par seuillage. Dans [10] l'algorithme FA a été utilisé pour optimiser les fonctions objectives de Kapur et celle de cross entropie. Dans [11] l'algorithme des lucioles a été appliqué pour optimiser la fonction objective d'Otsu.

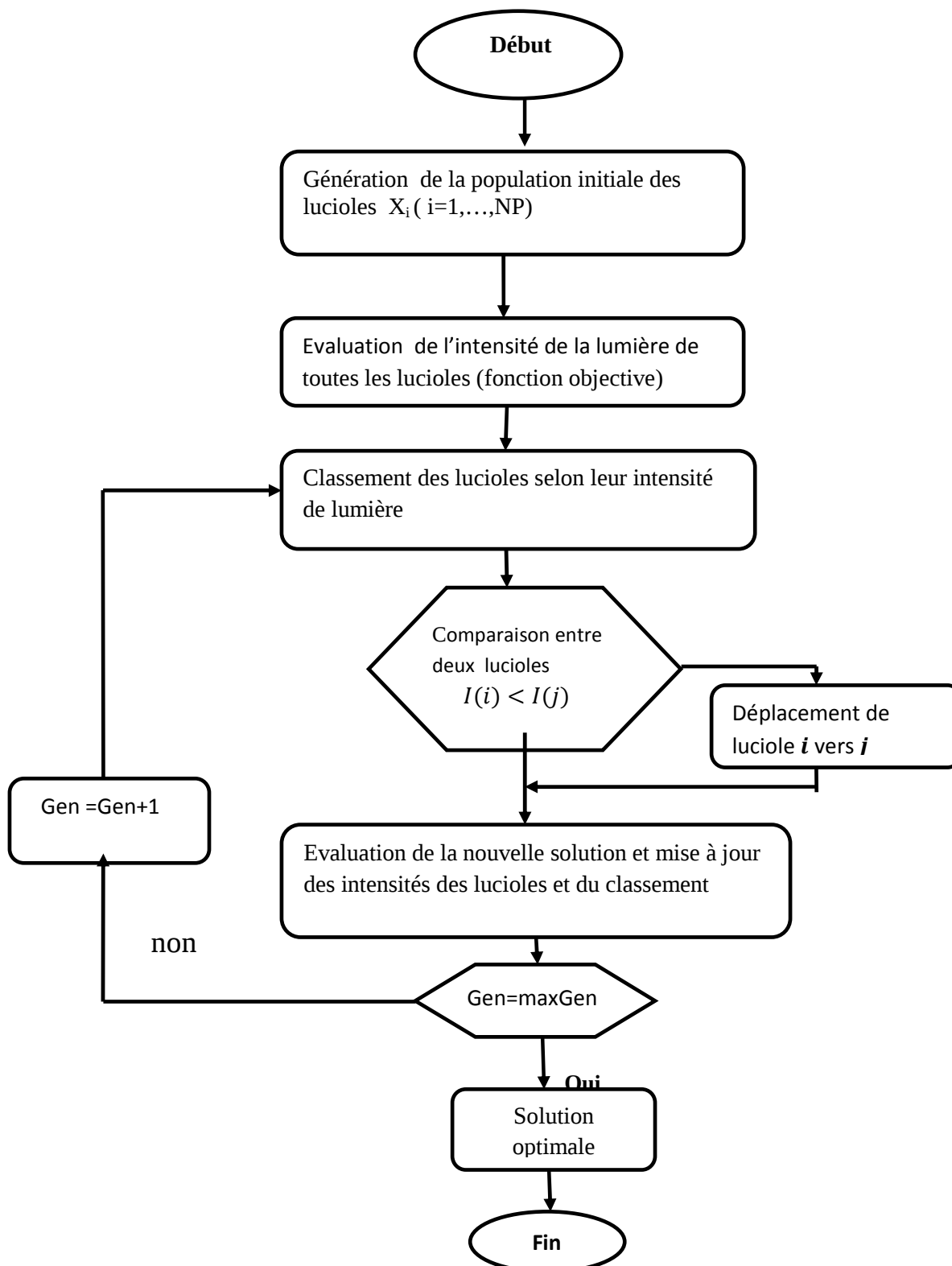


Fig2.4.principe général de l'algorithme des lucioles

**Génération de la population initiale :** La population initiale est générée aléatoirement, elle représente l'ensemble des solutions possibles.

**Classement des lucioles :** Il s'effectue par rapport à la fonction objective, et sert à déterminer le meilleur ou mauvais individu.

**Déplacement et mise à jour :** A chaque itération, la luminosité et l'attraction de chaque luciole sont calculées. Après le déplacement, la nouvelle luciole est évaluée, sa position et son intensité de lumière sont mises à jours.

### c. Algorithme de la recherche Coucou (CS)

La recherche coucou (Cuckoo Search - CS) est une métaheuristique qui s'inspire du mode de reproduction de certaines espèces de coucous. Leur stratégie de reproduction a pour particularité le fait que les femelles pondent leurs œufs dans le nid d'autres espèces (dont les œufs ressemblent) (Fig. 2.5). Ces œufs peuvent alors être couvés par des parents de substitution. Par ailleurs, quand les œufs de coucou parviennent à éclore dans le nid hôte (ils éclosent plus rapidement), les poussins coucou ont le réflexe d'éjecter les œufs de l'espèce hôte en dehors du nid et imitent même le cri des poussins hôtes dans le but d'être nourris par l'espèce hôte. Il peut cependant arriver que les œufs de coucous soient découverts ; dans ce cas, les parents de substitution les sortent du nid, ou abandonnent carrément le nid et commencent leur couvée ailleurs. Cette métaheuristique est basée sur le comportement parasite des espèces de coucous associée à une logique de déplacement de type « Levy flight » propres à certains oiseaux et certaines espèces de mouches.



**Fig.2.5 :** Œuf coucou dans un nouveau nid.

L'algorithme coucou se base sur les règles suivantes [13] :

Chaque coucou pond seulement un œuf à la fois et le place dans un nid choisi aléatoirement. Les meilleurs nids avec des œufs (solutions) de haute qualité sont conservés pour les prochaines générations.

Le nombre de nids hôtes est fixé et l'œuf pondu par un coucou peut être découvert par l'espèce hôte avec une probabilité  $Pa \in [0,1]$ . Dans ce cas, l'oiseau hôte, soit sort l'œuf du nid, soit quitte le nid et en construit un nouveau. Pour simplifier, cette dernière hypothèse peut être approchée par le remplacement d'une fraction  $Pa$  de  $n$  nids par des nouveaux.

Dans l'algorithme CS, chaque œuf dans un nid représente une solution et chaque coucou peut pondre un seul œuf (qui représente une solution). Dans ce cas, il n'y a plus de distinction entre œuf, nid ou coucou, et chaque nid correspond à un œuf qui représente aussi un coucou. Le but de l'algorithme CS est d'utiliser la nouvelle et potentiellement meilleure solution pour remplacer une moins bonne solution dans un nid.

Le pseudo-code de l'algorithme ci-dessous résume les étapes générales de l'algorithme de la recherche coucou.

### Algorithme de la recherche Coucou (CS)

#### Début

Fonction objective  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$

Générer une population initiale de  $n$  nids  $x_i$  ( $i = 1, 2, \dots, n$ )

**Tantque** ( $t < \text{MaxGeneration}$ ) ou (le critère d'arrêt) **faire**

**Pour** chaque coucou  $i$  **faire**

        Obtenir un Coucou aléatoirement par les vols de Levy

        Evaluer la fitness de chaque coucou

        Choisir un nid parmi  $n$  (soit  $j$ ) aléatoirement

**Fin pour**

**Si** ( $F_i < F_j$ ) (minimisation) **alors**

    Remplacer  $j$  par  $i$

**Fin si**

Une fraction ( $pa$ ) des mauvais nids est abandonnée et des nouveaux sont construits

Garder les meilleures solutions (ou nids avec des solutions de qualité)

Classer les solutions et trouver la meilleure actuelle

**Fin tantque**

Post-processus des résultats et visualisation

**Fin**

On trouve dans [18] [19], une utilisation de cet algorithme pour la segmentation d'images couleur par seuillage d'histogrammes. Dans [19], les fonctions d'Otsu et Kapur sont utilisées comme fonction d'objectives.

#### **d. Autres algorithmes inspirés de la nature**

Il existe plusieurs autres algorithmes dont le principe est inspiré de la nature. Parmi ces méthodes, on peut citer ceux qui ont été appliqués pour la segmentation d'images couleur par seuillages d'histogrammes.

**-Colonies de fourmis :** Cet algorithme consiste à mimer le comportement des fourmis lors de la recherche de nourriture [20].

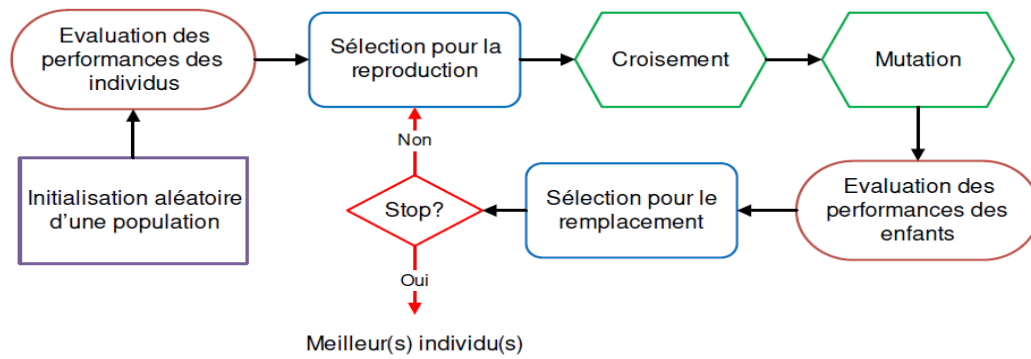
**-Recherche de nourriture bactérienne (BFO) :** Cet algorithme imite le comportement de recherche de nourriture bactérienne [21].

**-Colonies d'abeilles :** La méthode d'optimisation par colonie d'abeilles est basée sur le comportement des abeilles lors de la collecte du nectar [10].

#### **2.3.2.2. Les algorithmes évolutionnaires**

Les algorithmes évolutionnaires sont une famille d'algorithmes s'inspirant de la théorie de l'évolution « darwinienne » pour résoudre des problèmes divers. Selon la théorie du naturaliste Charles Darwin, énoncée en 1859, l'évolution des espèces est la conséquence de la conjonction de deux phénomènes, d'une part la sélection naturelle qui favorise les individus les plus adaptés à leur milieu à survivre et à se reproduire, laissant une descendance qui transmettra leurs gènes et d'autre part, la présence de variations non dirigées parmi les traits génétiques des espèces (mutations). Le terme Evolution Computation englobe une classe assez large de métaheuristiques telles que les algorithmes génétiques et l'évolution différentielle.

La figure 2.6 décrit le principe d'un algorithme évolutionnaire.



**Fig.2.6 :** principe d'un algorithme évolutionnaire.

### 2.3.2.3. Algorithmes génétiques

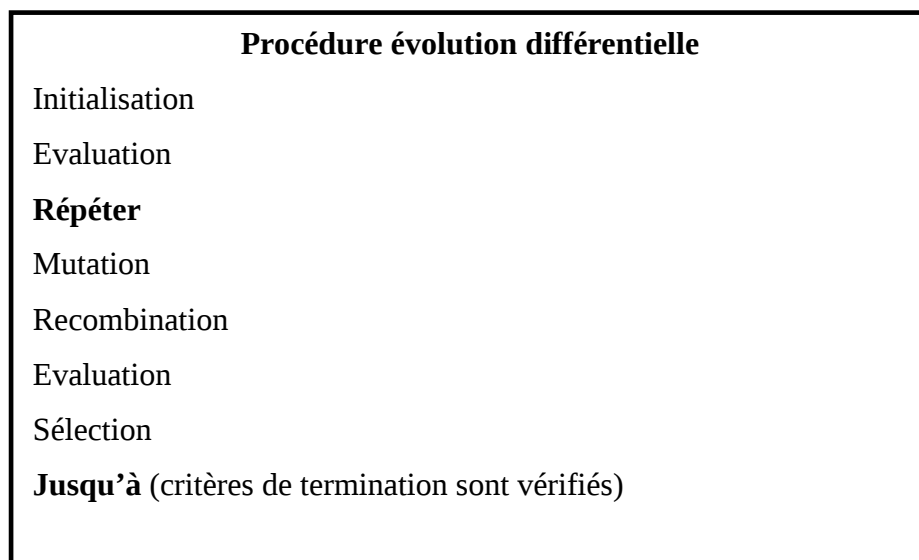
Les algorithmes génétiques (AGs) ont été développés par John Holland en 1975 [12]. Ce sont des algorithmes d'optimisation stochastiques fondés sur le mécanisme de la sélection naturelle et de la génétique [14]. Ils utilisent les principes de la survie des structures les mieux adaptées. Leur fonctionnement est extrêmement simple, on part d'une population d'individus générée d'une manière aléatoire et qui représentent des solutions potentielles du problème à résoudre. On évalue leur performance (fitness) par l'intermédiaire de la fonction objective. Sur la base de ces performances, on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples comme la sélection, le croisement et la mutation. On recommence ce cycle jusqu'à ce que l'on trouve une solution que l'on juge satisfaisante. L'algorithme de la figure(2.7) montre le principe de l'algorithme génétique.

1. Initialisation de la population de solutions  $P(t)$
2. Evaluer chaque individu de  $P(t)$
3. Tant que le critère d'arrêt n'est pas satisfait faire
  - 3.1.  $t = t + 1$
  - 3.2. Sélectionner  $P(t+1)$  à partir de  $P(t)$
  - 3.3. Croisement  $P(t+1)$
  - 3.4. Muter  $P(t+1)$
  - 3.5. Evaluer  $P(t+1)$
 Fin tant que
4. Afficher la meilleure solution de la population

**Fig.2.7 :** Algorithme génétique

### 2.3.2.4. Algorithme de l'évolution différentielle (DE)

Proposé par Prince et Storn [25], l'algorithme d'évolution différentielle (DE) est une version améliorée de l'AG. Il dispose d'opérateurs de mutation, croisement et sélection. Cependant l'algorithme DE marque la différence avec l'AG en donnant plus d'importance à la mutation. Cette mutation a pour but de créer une solution « mutant » à partir des différences entre deux solutions de la population. Dans la méthode DE, la population initiale est générée par tirage aléatoire uniforme sur l'ensemble des valeurs possibles de chaque variable. Les bornes inférieures et supérieures des variables sont spécifiées par l'utilisateur selon la nature du problème. La figure (2.8) représente la structure générale d'un algorithme DE.



**Fig.2.8:** Structure d'un algorithme DE.

Après l'initialisation, l'algorithme effectue une série de transformations sur les individus, dans un processus appelé évolution.

La population contient individus. Chaque individu  $x_{i,(t)}$  est un vecteur de dimension  $D$ , où  $t$  désigne la génération :

$$x_{i,(t)} = (x_{1i,(t)}, x_{2i,(t)}, x_{Di,(t)}) \quad \text{avec } i = 1, 2, \dots, N \quad (\text{II.1})$$

A chaque génération, l'algorithme applique successivement ces trois opérations (mutation, croisement et sélection) sur chaque vecteur pour produire un vecteur d'essai :

$$u_{i,(t+1)} = (u_{1i,(t+1)}, u_{2i,(t+1)}, u_{Di,(t+1)}) \text{ avec } i = 1, 2, \dots, N \quad (\text{II.2})$$

Une opération de sélection permet de choisir les individus à conserver pour la nouvelle génération ( $t + 1$ ).

#### a - Mutation

Pour chaque vecteur courant  $x_{i,(t)}$  , on génère un vecteur mutant  $v_{i,(t+1)}$  qui peut être créé en utilisant une des stratégies de mutation suivantes :

##### - Rand/1 :

$$v_{i,(t+1)} = x_{r1,(t)} + F \cdot (x_{r2,(t)} - x_{r3,(t)}) \quad (\text{II.3})$$

##### - Best/1 :

$$v_{i,(t+1)} = x_{best,(t)} + F \cdot (x_{r1,(t)} - x_{r2,(t)}) \quad (\text{II.4})$$

##### - Current to best/1 :

$$v_{i,(t+1)} = x_{i,(t)} + F \cdot (x_{r1,(t)} - x_{r2,(t)}) + F \cdot (x_{best,(t)} - x_{i,(t)}) \quad (\text{II.5})$$

##### - Best/2 :

$$v_{i,(t+1)} = x_{best,(t)} + F \cdot (x_{r1,(t)} - x_{r2,(t)}) + F \cdot (x_{r3,(t)} - x_{r4,(t)}) \quad (II.6)$$

- **Rand/2 :**

$$v_{i,(t+1)} = x_{r1,(t)} + F \cdot (x_{r2,(t)} - x_{r3,(t)}) + F \cdot (x_{r4,(t)} - x_{r5,(t)}) \quad (II.7)$$

Les indices  $r_1, r_2, r_3, r_4$ , et  $r_5 \in \{1, 2, \dots, N\}$  sont des entiers aléatoires et tous différents. Ils sont également choisis différents de l'indice courant  $i$ .  $x_{best,(t)}$  est le meilleur individu à la Gène génération.  $F \in [0, 2]$  est une valeur constante, appelée differential weight, qui contrôle l'amplification de la variation différentielle de  $(x_{i,(t)} - x_{j,(t)})$ .

### b - Croisement

Après la mutation, une opération de croisement binaire forme le vecteur d'essai final

$u_{i,(t+1)}$ , selon le vecteur  $(x_{i,(t)})$  et le vecteur mutant correspondant  $v_{i,(t+1)}$ . L'opération de croisement est introduite pour augmenter la diversité des vecteurs de paramètres perturbés. Le nouveau vecteur  $u_{i,(t+1)}$  est donné par la formule suivante :

$$u_{ji,(t+1)} = \begin{cases} v_{ji,(t+1)} & \text{si } randb(j) \leq CR \text{ ou } j = rnbr(i) \\ x_{ji,(t)} & \text{si } randb(j) > CR \text{ et } j \neq rnbr(i) \end{cases} \quad \text{Pour tout } j \in \{1, 2, \dots, D\} \quad (II.8)$$

Où  $randb(j)$  est la  $j^{\text{ème}}$  valeur procurée un générateur de nombre aléatoire uniforme appartenant à l'intervalle  $[0, 1]$ .  $CR$  est le coefficient de croisement qui appartient à l'intervalle  $[0, 1]$  et est déterminé par l'utilisateur.  $rnbr(i)$  est un indice choisi au hasard dans l'ensemble  $\{1, 2, \dots, N\}$ .

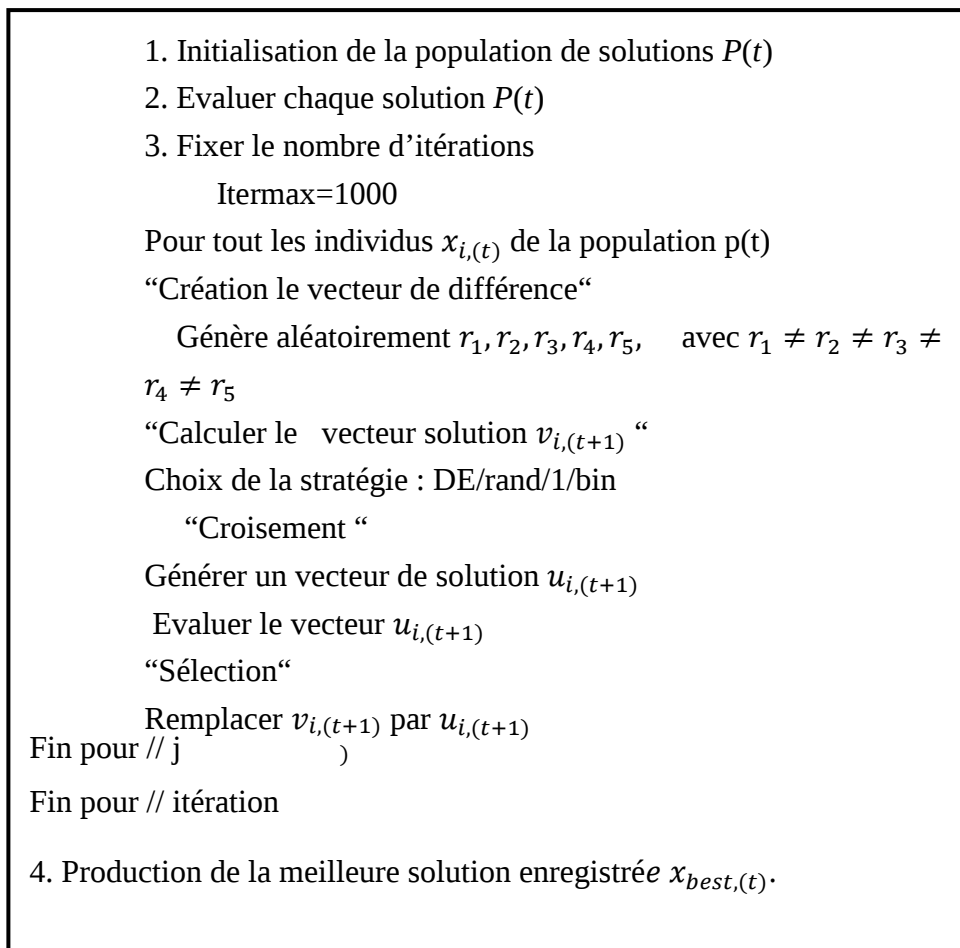
### c - Sélection

Pour décider quel vecteur, parmi  $(u_{i,(t)})$  ou  $(x_{i,(t)})$ , doit être choisi dans la génération  $(t+1)$  on doit comparer les valeurs de fonction du cout de ces deux vecteurs. En effet, on garde le

vecteur ayant la plus petite valeur de fonction du cout en cas de minimisation. Le nouveau vecteur  $x_{i,(t+1)}$  est choisi selon l'expression suivante :

$$x_{i,(t+1)} = \begin{cases} u_{i,(t+1)} & \text{si } f(u_{i,(t+1)}) < f(x_{i,t}) \\ x_{i,t} & \text{sinon} \end{cases} \quad (\text{II.9})$$

Le réglage des principaux paramètres de l'algorithme (taille de la population N, facteur de mutation F et facteur de croisement CR) contribue de façon importante à l'efficacité de la méthode. La figure (2.8) représente les étapes suivies lors de l'application de l'algorithme DE.



**Fig.2.9** : Algorithme de l'évolution différentielle.

## 2.4. Conclusion

Dans ce chapitre nous avons dressé un état de l'art quelques metaheuristiques d'optimisation qui ont été appliquées au seuillage d'histogrammes. Nous nous sommes intéressés aux metaheuristiques a population de solutions. Nous avons détaillé l'algorithme à évolution différentielle qui fait partie des algorithmes évolutionnaires, notre choix s'est porté sur le DE car sa programmation est relativement simple.

### 3.1. Introduction

Nous présenterons dans ce chapitre la méthode de segmentation d'images couleur par seuillage d'histogrammes basée sur l'algorithme DE. Les résultats obtenus par cette méthode sont évalués sur images tests biomédicales.

### 3.2. Méthode de segmentation adoptée pour la segmentation d'images couleur

La méthode de segmentation d'images couleur que nous proposons est basée sur le seuillage des histogrammes R, V et B par la méthode d'optimisation DE.

La couleur de chaque pixel de l'image originale est définie par un triplet de valeurs correspondant aux composantes R, V et B, chacune de ces composantes est généralement codée sur 8 bits, et peut prendre 256 valeurs.

La fonction objective utilisée est la fonction d'Otsu donnée par l'expression suivante (voir chapitre1):

$$J(T^R, T^V, T^B) = J(T^R) + J(T^V) + J(T^B). \quad (\text{III.1})$$

Où

$$J(T^R) = \sum_{k=1}^K P_k^R (\mu_k^R - \mu^R)^2, \quad (\text{III.2})$$

$$J(T^V) = \sum_{k=1}^K P_k^V (\mu_k^V - \mu^V)^2, \quad (\text{III.3})$$

$$J(T^B) = \sum_{k=1}^K P_k^B (\mu_k^B - \mu^B)^2 \quad (\text{III.4})$$

$T^R = \{t_1^R, t_2^R, \dots, t_{K-1}^R\}$ ,  $T^V = \{t_1^V, t_2^V, \dots, t_{K-1}^V\}$  et  $T^B = \{t_1^B, t_2^B, \dots, t_{K-1}^B\}$  étant les seuils recherchés de chaque composante R, V, B.

Le vecteur solution constitue donc une concaténation des trois vecteurs seuils  $T^R$ ,  $T^V$  et  $T^B$  tel que:

$$T = \{t_1^R, t_2^R, \dots, t_{K-1}^R, t_1^V, t_2^V, \dots, t_{K-1}^V, t_1^B, t_2^B, \dots, t_{K-1}^B\} \quad (\text{III.5})$$

Le nombre de variables est donc égal à  $3 \times (K-1)$ .

Le calcul des seuils par la méthode de DE utilise une population de NP solutions où chaque solution contient  $3 \times (K-1)$  seuils.

Début

1. Chargement de l'image couleur
2. Décomposition de l'image en composantes R, V, B
3. Calcul des histogrammes de chacune des composantes R, V, B
4. Choix des paramètres de l'algorithme DE:
  - Ns: Nombre de seuils
  - NP: Taille de la population
  - MAXiter: Nombre maximal d'itérations
  - CR: Le taux de croisement
  - F: Le facteur d'échelle
5. Calcul des seuils par la méthode de l'évolution différentielle
6. Affichage de l'image segmentée

### 3.3. Tests et résultats

Pour évaluer la méthode de segmentation d'images couleur par seuillage d'histogrammes proposée, nous l'avons appliqué sur trois images couleur médicales. La première image de taille (500×1090) est représentée sur la figure (3.1), elle contient un ensemble de bactéries de formes différentes (bâtonnets et coques), mais de même couleur. Les trois images monochromes R, V et B ainsi que leurs histogrammes sont également affichées sur la figure (3.1). Les histogrammes des images R et B ont un seul mode alors que celui de la composante V est bimodal. La deuxième image de taille (158×319) contient un ensemble de bactéries en forme de bâtonnets et de coques de formes mais avec des couleurs différentes (Fig. 3.2). Les trois images monochromes R, V et B correspondantes ainsi que leurs histogrammes sont également affichées sur la figure (3.2). Les trois histogrammes un seul mode chacun. La troisième image de taille (709×960) contient des cellules sanguines. La figure (3.3) montre cette image, ses trois composantes R, V et B ainsi que leurs histogrammes. On peut difficilement remarquer que les trois histogrammes sont bimodaux.

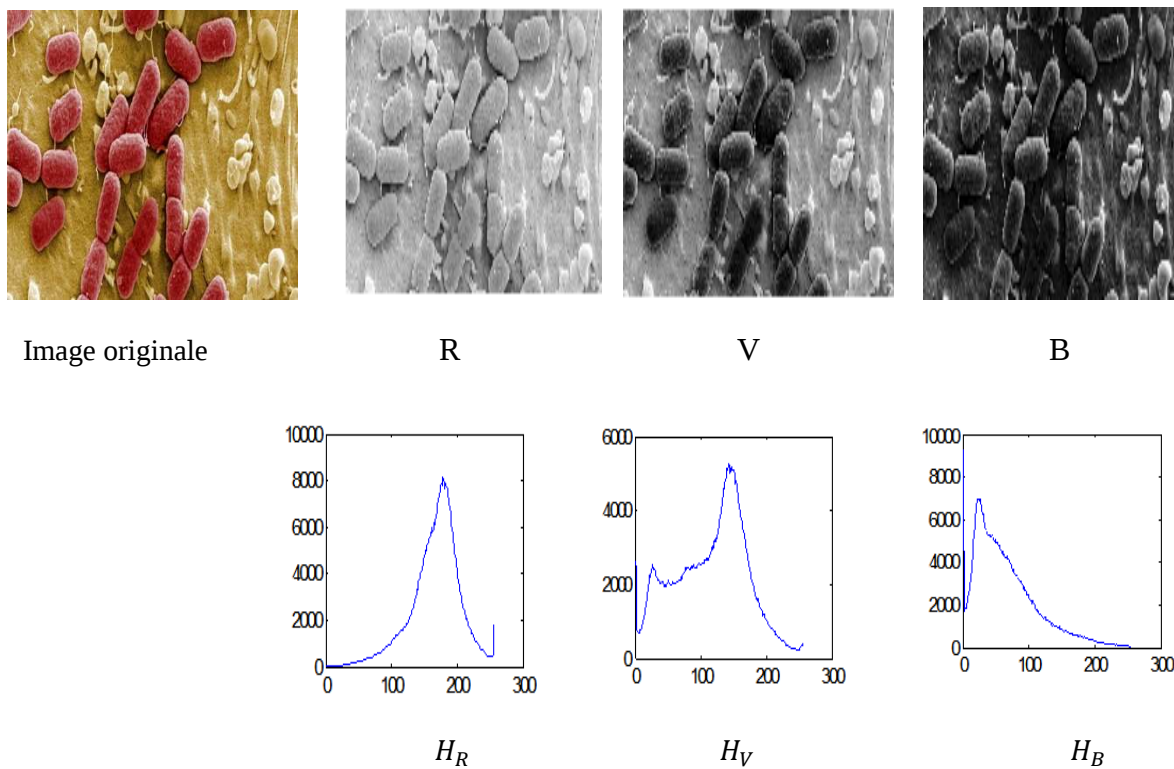


Fig. 3.1: Image Test 1.

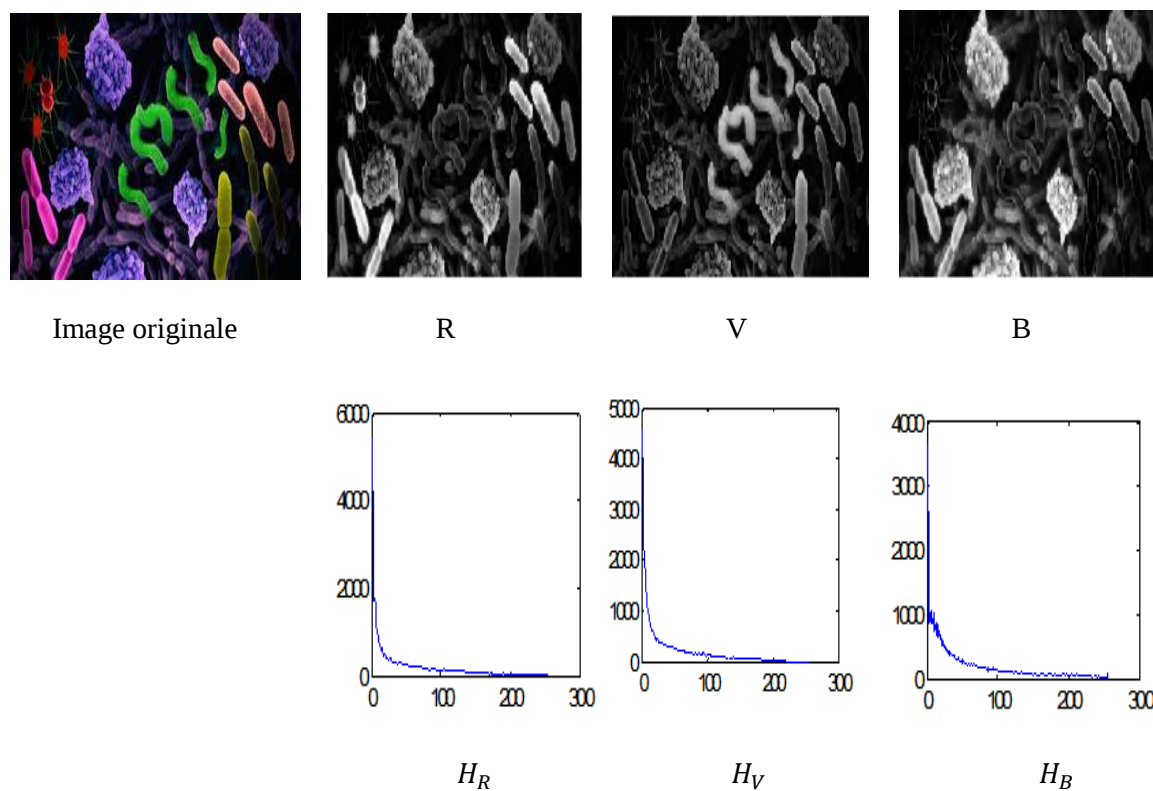
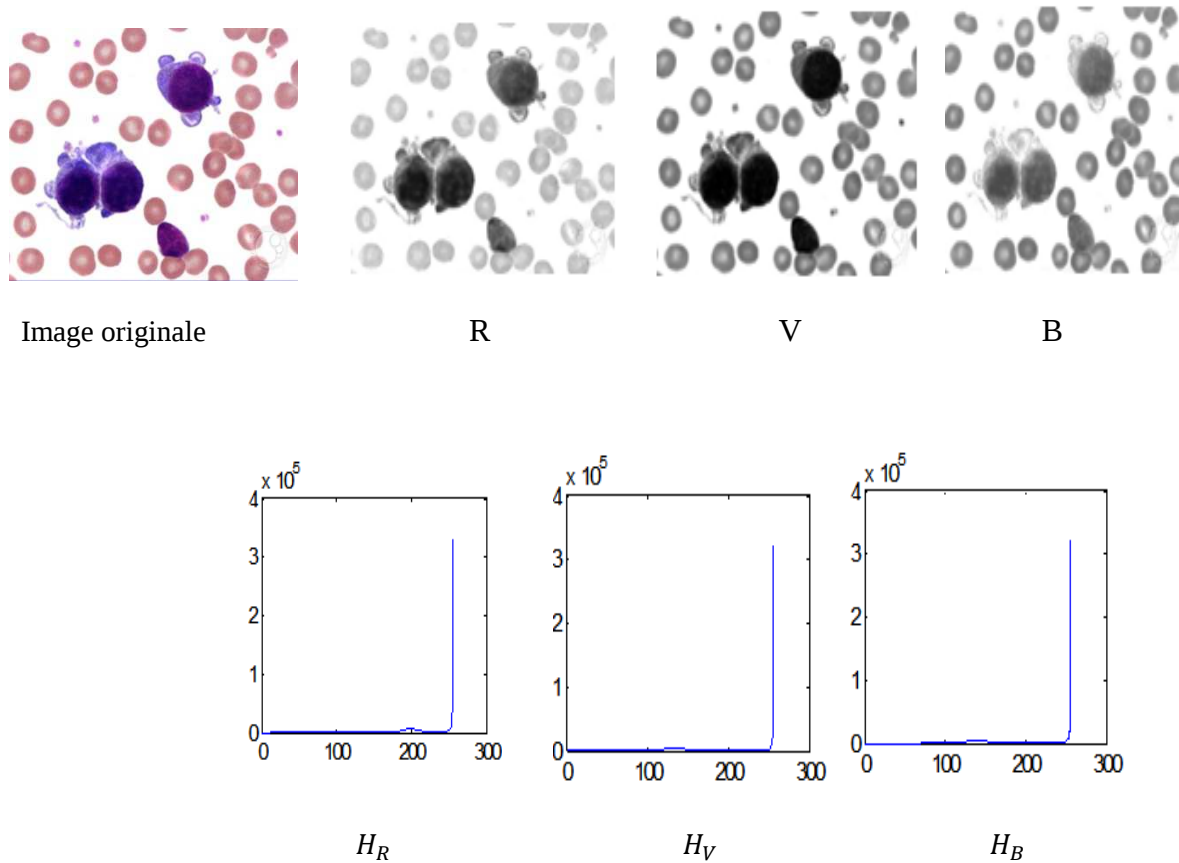


Fig.3.2: Image Test 2.

**Fig. 3.3** : Image test 3

L'algorithme DE est appliqué avec les paramètres donnés dans la table (3.1). Les valeurs de ces paramètres sont les mêmes que ceux utilisés dans [25]. Notons qu'également la stratégie (DE/rand/1/bin) est utilisée dans nos tests.

|                                |      |
|--------------------------------|------|
| Taille de la population<br>NP  | 5    |
| Nombre maximal<br>d'itérations | 1000 |
| Taux d'échelle<br>F            | 0.8  |
| Taux de croisement<br>CR       | 0.3  |

**Tableau (3.1)** : Paramètres de l'algorithme DE

Le tableau (3.2) donne les résultats du seuillage obtenus par l'algorithme DE pour un nombre de seuils allant de 1 à 4. Les figures (3.7) à (3.9) montrent les résultats obtenus par le seuillage d'histogrammes basé sur l'algorithme DE.

| Image (s) | Ns | Seuils                                                                                           | $\mathcal{J}(T^R, T^V, T^B)$ | Le temps de calculs (secondes) |
|-----------|----|--------------------------------------------------------------------------------------------------|------------------------------|--------------------------------|
| Image 1   | 1  | R : (156), (255)<br>V : (106), (255)<br>B : (85), (255)                                          | 0.00022391                   | 5.31                           |
|           | 2  | R : (130), (180)<br>V : (84), (149)<br>B : (57), (123)                                           | 0.00018113                   | 10.58                          |
|           | 3  | R : (114), (161), (197)<br>V : (65), (119), (172)<br>B : (45), (89), (147)                       | 0.00016683                   | 15.87                          |
|           | 4  | R : (108), (156), (179), (204)<br>V : (51), (95), (133), (174)<br>B : (36), (73), (111), (166)   | 0.0001614                    | 21.38                          |
| Image 2   | 1  | R : (77), (255)<br>V : (65), (220)<br>B : (89), (255)                                            | 0.00015498                   | 5.34                           |
|           | 2  | R : (47), (128)<br>V : (38), (103)<br>B : (51), (136)                                            | 0.00012944                   | 10.53                          |
|           | 3  | R : (31), (84), (155)<br>V : (28), (69), (126)<br>B : (34), (85), (157)                          | 0.00012229                   | 15.84                          |
|           | 4  | R : (22), (65), (120), (182)<br>V : (15), (38), (79), (135)<br>B : (30), (67), (120), (178)      | 0.00011955                   | 21.27                          |
| Image 3   | 1  | R : (161), (206)<br>V : (167), (180)<br>B : (183), (196)                                         | 0.00010521                   | 5.28                           |
|           | 2  | R : (136), (255)<br>V : (81), (196)<br>B : (157), (219)                                          | 8.9756e-005                  | 10.49                          |
|           | 3  | R : (88), (160), (255)<br>V : (76), (154), (217)<br>B : (125), (173), (255)                      | 8.741e-005                   | 15.80                          |
|           | 4  | R : (84), (156), (204), (234)<br>V : (74), (136), (171), (223)<br>B : (120), (149), (185), (231) | 8.66472e-005                 | 21.16                          |

Tableau 3.2: Résultats du seuillage par l'algorithme DE.



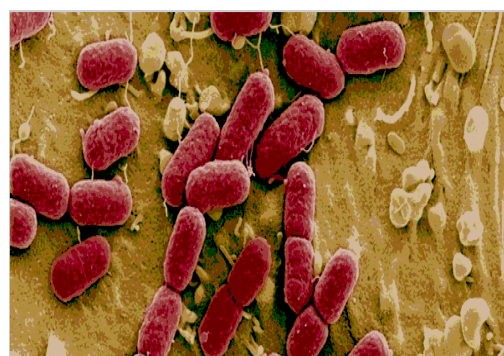
NS=1



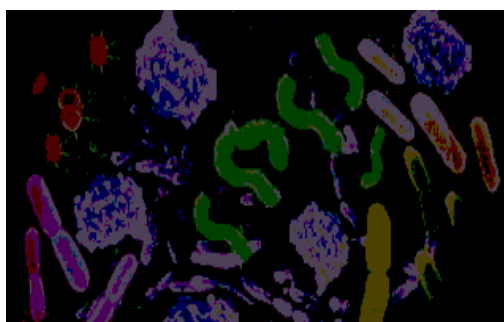
NS=2



NS=3



NS=4

**Fig .3.4 :** Résultats de la segmentation de l'image1 par l'algorithme DE

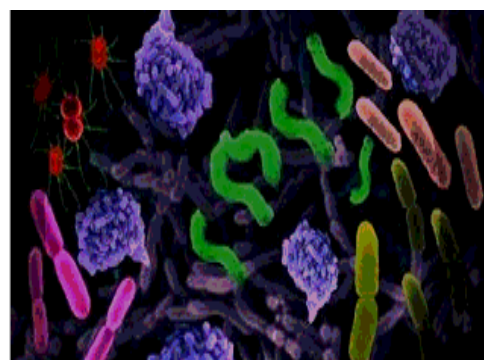
NS=1



NS=2

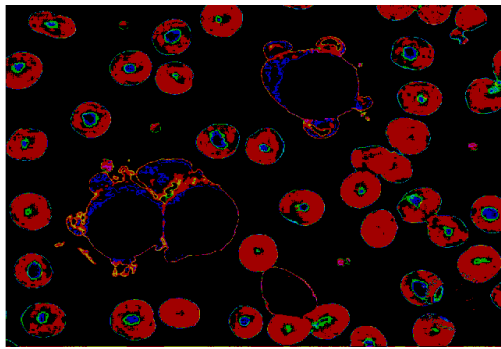


NS=3

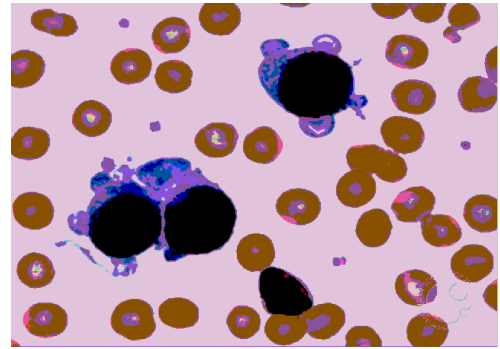


NS=4

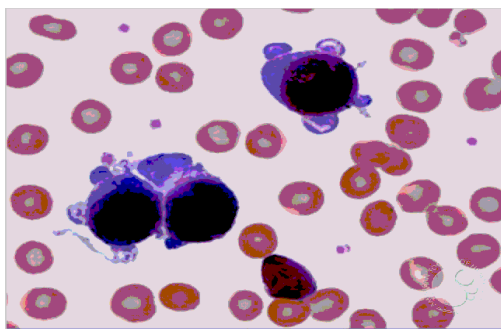
**Fig.3.5 :** Résultats de la segmentation de l'image 2 par l'algorithme DE.



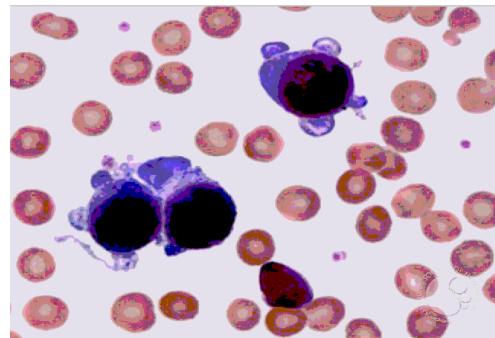
NS=1



NS=2



NS=3



NS=4

**Fig.3.6:** Résultats de la segmentation de l'image3 par l'algorithme DE.

Le tableau précédent montre que, d'une part, le temps de calcul des seuils augmente avec l'augmentation du nombre de seuils mais reste le même quelque soit la taille de l'image. D'autre part, la fonction objective diminue au fur et à mesure que le nombre de seuils augmente. En effet, on peut voir sur les figures (3.7) à (3.9) que les images segmentées deviennent, visuellement, plus proches des images originales lorsque le nombre de seuils augmente. Ceci peut s'expliquer par le fait que le nombre de classes dans l'image segmentée (nombre de couleurs) augmente.

### 3.4. Comparaison des résultats

Pour évaluer la méthode proposée d'une manière objective, nous avons comparé ses résultats avec ceux obtenus avec la méthode de recherche exhaustive. Cette technique consiste à faire varier tous les seuils entre 0 et 255. Pour chaque combinaison de seuils, on évalue la fonction objective. Finalement, on retient la meilleure solution, c'est à dire celle qui donne la meilleure fonction objective. Nous avons utilisé la fitness (fonction objective d'Ostu) et le temps de

calcul comme critères d'évaluation. Concernant le critère  $J(T^R, T^V, T^B)$ , plus il est petit, meilleur est le résultat.

Le tableau (3.3) contient les valeurs des seuils, la fonction objective ainsi le temps de calcul obtenus par la recherche exhaustive par l'algorithme DE pour un nombre de seuils égal à 3.

|       | Recherche exhaustive                                                        |                                        |       | Algorithme DE                                                               |                                        |       |
|-------|-----------------------------------------------------------------------------|----------------------------------------|-------|-----------------------------------------------------------------------------|----------------------------------------|-------|
| Image | Seuils                                                                      | $J(T^R, T^V, T^B)$<br>$\times 10^{-4}$ | T(s)  | Seuils                                                                      | $J(T^R, T^V, T^B)$<br>$\times 10^{-4}$ | T(s)  |
| 1     | R : (118), (163), (198)<br>V : (64), (118), (170)<br>B : (44), (87), (145)  | 1.6678                                 | 48.02 | R : (114), (161), (197)<br>V : (65), (119), (172)<br>B : (45), (89), (147)  | 1.6683                                 | 15.87 |
| 2     | R : (31), (84), (156)<br>V : (27), (70), (125)<br>B : (35), (86), (157)     | 1.2228                                 | 47.81 | R : (31), (84), (155)<br>V : (28), (69), (126)<br>B : (34), (85), (157)     | 1.2229                                 | 15.84 |
| 3     | R : (89), (161), (227)<br>V : (75), (154), (217)<br>B : (126), (171), (225) | 0.87403                                | 47.91 | R : (88), (160), (255)<br>V : (76), (154), (217)<br>B : (125), (173), (255) | 0.8741                                 | 15.80 |

**Tableau (3.3) :** Résultats du seuillage des images tests basé sur la recherche exhaustive.

A travers le tableau (3.3), nous pouvons que les valeurs des seuils obtenus par l'algorithme DE sont proches des valeurs optimales, obtenus par la recherche exhaustive. Les valeurs de la fonction objective sont également très proches. Cependant, pour un nombre de seuils égal à 3, l'algorithme DE est beaucoup plus rapide que la recherche exhaustive.

#### 4. Conclusion

Dans ce chapitre, nous avons présenté les résultats de la segmentation par seuillage d'histogramme basée l'algorithme DE. Nous avons montré que l'algorithme DE peut fournir des valeurs optimales avec un temps de calcul beaucoup plus réduit surtout lorsque le nombre de seuils est élevé.

## Conclusion générale

---

Dans ce mémoire, nous nous sommes intéressés à la segmentation d'images couleur par seuillage d'histogrammes basée sur une méthode d'optimisation métaheuristique en l'occurrence l'algorithme à évolution différentielle.

Nous pouvons retenir de ce travail que la segmentation d'images couleur par seuillage d'histogramme est une approche très intéressante dans la mesure où elle est formulée comme un problème d'optimisation qui est simple à implémenter. Son seul problème est le temps de calcul lors de la recherche des seuils surtout lorsque leurs nombres augmentent.

Pour accélérer les calculs, nous avons fait appel aux techniques d'optimisation métaheuristique et en particulier à l'algorithme à évolution différentielle. Cet algorithme est capable de converger vers la solution optimale avec un temps de calcul réduit. De plus, ils ne nécessitent pas beaucoup de paramètres.

Des tests effectués sur des images biomédicales ont montré l'intérêt de cette approche pour déterminer les valeurs optimales des seuils avec un temps de calcul beaucoup plus réduit comparativement avec la technique de recherche exhaustive.

La fonction objective utilisée dans ce mémoire pour la recherche des seuils est celle d'Otsu. Une des perspectives à ce travail est d'utiliser d'autres fonctions objectives telles que les fonctions basées sur l'entropie de Shanon (Kapur), cross-entropie, entropie de Tsallis, ect.... D'autres part, nous avons considéré que les histogrammes R, V et B. Il serait aussi intéressant de tester la méthode proposée en considérant d'autres espaces couleur comme l'espace Lab, Luv, ect...

Quand à l'algorithme DE, il serait intéressant de tester les autres stratégies de mutation et étudier l'influence des paramètres d'échelle F et le taux de croisement CR sur les résultats de seuillage.

## Bibliographies

- [1] Bitam Abdelmadjid « segmentation d'image couleurs par la méthode JSEG », thèse de magistère, département d'électronique, Université Tizi-Ouzou, 2006
- [2] Abdelli Ouardia « segmentation d'images par seuillage d'histogrammes bidimensionnels » thèse de magistère, département d'automatique, Université de Tizi-Ouzou, 2011.
- [3] Melle Sbili Lila « segmentation d'image par classification floue », thèse de magister, département d'automatique, Université de Tizi-Ouzou, 2015.
- [4] Amir Nakib « conception de métaheuristiques d'optimisation pour la segmentation .Application aux images biomédicales ».informatiques [Cs], Université Paris XII Val de Marne, 2007, Français
- [5] Lehamel Malha « segmentation d'image texturées a partir des attribue fractale ». Thèse de magister, département d'automatique, Université de Tizi-Ouzou.
- [6] Julien Lepagnot « conception de metaheuristiques pour l'optimisation dynamique, application à analyse de séquences d'image IRM. » Thèse de doctorat en informatique Université Paris-Est, 2011.
- [7] Menghour Kamilia « approches bio-inspirés pour la sélection d'attribue », mémoire de doctorat, département d'informatique, Université Badji Mouhktar Anaba ,2014.
- [8] M.Charlie Vanaret « hybridation d'algorithmes évolutionnaires et de méthode d'intervalles pour l'optimisation des problèmes déficèle », thèse de doctorat, (I.R.I.T), institut national polytechnique de Toulouse, 2015.
- [9] Ilhem Boussaid « Perfection de métaheuristiques pour l'optimisation continue » Université Paris-Est, 2013.Français
- [10] Lifang He, Songwei Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation, *Neurocomputing* (2017)
- [11] V. Rajinikanth, M. S. Couceiro, RGB Histogram based Color Image Segmentation Using Firefly Algorithm, (2014)

- [12] Ahmed Nasreddine Benaichouche, « conception de métaheuristiques d'optimisation pour la segmentation d'image : application aux images IRM du cerveau et aux images se tomographie par émission de posions, analyse numérique » [cs.NA].Université Paris –Est ,2004.Français.
- [13] Philippe Feyel. « Optimisation des correcteurs par les metaheuristiques. Application à la stabilisation inertielle de ligne de visée » 2015
- [14] Yang X.S.: “Firefly Algorithms for Multimodal Optimization, Stochastic Algorithms: Foundations and Applications”, SAGA 2009, Lecture Notes in Computer Science, Springer-Verlag, Berlin,
- [15] Yang X.S.:”Nature-Inspired Metaheuristic Algorithms”. Luniver Press, UK (2008)
- [16] Yang X.S.: “Firefly algorithm, stochastic test functions and design optimization”, International Journal of Bio-Inspired Computation, (2010)
- [17] Soham Sarkar , Swagatam Das , Sheli Sinha Chaudhuri , A Multilevel Color Image Thresholding Scheme based on Minimum Cross Entropy and Differential Evolution, *Pattern Recognition Letters* (2014)
- [18] S. Pare, et al., A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve, *Appl. Soft Comput. j.* (2016)
- [19] A.K. Bhandari, A. Kumar, S. Chaudhary, G.K. Singh, A Novel Color Image Multilevel thresholding based Segmentation using Nature Inspired Optimization Algorithms, *Expert Systems With Applications* (2016),
- [20] S. Dey, et al., New quantum inspired meta-heuristic techniques for multi-level colour image thresholding, *Appl. Soft Comput. J.* (2015)
- [21] WJ Tang, QH Wu et JR Saunders. Bacterial foraging algorithm for dynamic environments. Dans *Evolutionary Computation*, 2006. CEC 2006 IEEE Congress on, pages 1324–1330. IEEE, 2006
- [22] Oussama El Gerari « contribution a l'amélioration des techniques d la programmation génétique et de la programmation évolutionnaire », thèse de magistère, département informatique, Université du Littoral Cote d'Opale, 2011
- [23] Kevin M Passino. Biomimicry of bacterial foraging for distributed optimization and control. *Control Systems, IEEE*, 22(3):52–67, 2002.

[24] Hocini Lotfi « conception de métaheuristiques d'optimisation pour la segmentation des images de télédétection », thèse de magister, département d'électronique, université Tizi-Ouzou ,2012.

[25] K. Price, R. Storn, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces Report TR95 -012, International Computer Science Institute, Berkeley, California. Journal of Global Optimization vol. 11, pp. 341–359, 1997.

[26] J. Kennedy, and R. C. Eberhart, Particle swarm optimization. IEEE Int. Conf. on Neuronal networks, Piscataway (Japan). vol. 4, pp. 1942–1948, 1995.

### Les espaces couleurs

Il existe de nombreux systèmes de représentation de la couleur, chacun représente des caractéristiques particulières, ceux-ci peuvent être regroupés en quatre familles principales : les systèmes de primaires, les systèmes perceptuellement uniformes, les systèmes perceptuels et les systèmes d'axes indépendants.

### L'espace RVB

Le système RVB, qui fait partie de la famille des systèmes primaires, est le système le plus utilisé dans le domaine de l'imagerie couleur de par sa facilité d'utilisation mais aussi de par la dépendance aux matériels (carte d'acquisition, carte vidéo, camera, etc) vu que ces dernières n'utilisent que le triplet (R, V, B). Cependant, la définition de l'espace de représentation de la couleur n'étant pas unique puisqu'il dépend des primaires ainsi que du blanc de référence ; différents systèmes ont alors vu le jour. Le système de représentation RVB introduit par CIE en 1973 (Comité International d'Eclairage) reste celui de référence même s'il présente quelques inconvénients tels que :

- certaines couleurs ne sont pas représentées par addition des trois spectres.
- les composantes trichromatiques peuvent prendre des valeurs négatives.

Pour éliminer l'existence des valeurs négatives la CIE a créé un autre espace nommé XYZ.

### L'espace XYZ

L'espace XYZ devient alors le système de référence colorimétrique constitué de trois primaires X, Y et Z dites virtuelles qui se déduisent du système RVB par une simple transformation linéaire décrite par l'équation suivante :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.201 \\ 0.229 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{bmatrix} \times \begin{bmatrix} R \\ V \\ B \end{bmatrix} \quad (1)$$

La CIE a défini les coordonnées chromatiques du système (x y z) qui se calculent à partir des composants trichromatiques (X Y Z) par la relation suivantes qui forment le système (X Y Z), normalisé noté (x,y,z)

## Annexe

---

$$\begin{cases} x = \frac{X}{X+Y+Z} \\ y = \frac{Y}{X+Y+Z} \\ z = \frac{Z}{X+Y+Z} \end{cases} \quad (2)$$

Comme  $x+y+z=1$ ,  $z$  peut être déterminé à partir de  $x, y$  et  $z$ , ce qui permet de représenter la couleur dans un plan .

### L'espace $L^*u^*v$

Le problème avec l'espace XYZ est son manque d'uniformité dans la perception à savoir que deux couleurs à la même distance d'une autre ne sont pas nécessairement perceptuellement équidistantes. Pour corriger cet inconvénient la CIE a proposé le modèle  $L^*u^*v$  qui est perceptuellement uniforme c'est-à-dire que si deux couleurs A et B sont aussi proches entre elles que deux autres couleur C et D, alors la différence perçue entre A et B sera équivalente à celle perçue entre C et D. Les équations permettant de passer du système XYZ au système  $L^*u^*v$  sont :

$$\begin{cases} L^* = \begin{cases} 166 \left( \frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16 \text{ si } \frac{Y}{Y_0} > 0.008856 \\ 903.3 \left( \frac{Y}{Y_0} \right) & \text{Ailleurs} \end{cases} \\ u^* = 13L^*(u' - u'_0) \\ v^* = 13L^*(v' - v'_0) \end{cases} \quad (3)$$

$$\begin{cases} u' = \frac{4X}{X+15Y+3Z} \\ u'_0 = \frac{4X_0}{X_0+15Y_0+3Z_0} \\ v' = \frac{9X}{X+15Y+3Z} \\ v'_0 = \frac{9X_0}{X_0+15Y_0+3Z_0} \end{cases} \quad (4)$$