



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
ECOLE NATIONALE SUPERIEURE D'INFORMATIQUE

## **Classes préparatoires intégrées (CPI)**

**2ème année**

### **Projet Pluridisciplinaire**

Segmentation des images par  
région et extraction des  
caractéristiques.

#### **Sujet n°8**

#### **Equipe n°16 :**

- Bentaiba Abderrahmane (CE)
- Hamani Sid Ahmed
- Ould Said Takieddine
- Melbous Abdelmadjid
- Boukari Imene
- Ait Habouche Manel

#### **Encadré par :**

- Mme Kacet
- Mme Daoudi

**Année universitaire : 2015 - 2016**

## TABLE DES MATIERES

Introduction générale.....	4
<b>Partie I : Etat de l'art.....</b>	<b>6</b>
<b>Chapitre 1 : Généralités sur l'imagerie .....</b>	<b>6</b>
1. La notion d'image : .....	6
2. La création des images : .....	6
3. Image numérique : .....	6
4. Conclusion : .....	7
<b>Chapitre 2 : Segmentation des images : .....</b>	<b>8</b>
1. Définition : .....	8
2. Applications : .....	8
3. Différentes méthodes et procédés de segmentation : .....	9
4. Conclusion : .....	11
<b>Chapitre 3 : Analyse de régions et extraction des caractéristiques .....</b>	<b>12</b>
<b>Partie II : Conception de la solution .....</b>	<b>15</b>
<b>Chapitre 1 : 1<sup>ère</sup> méthode de segmentation (Seuillage).....</b>	<b>15</b>
1. Définition : .....	15
2. Pseudo-algorithme général : .....	16
3. Analyse détaillée : .....	16
4. Calcul du seuil : .....	17
a. Méthode d'Otsu : .....	17
b. Méthode de Li (Cross-Entropie) : .....	18
c. Méthode de Yen (Corrélation entropique) : .....	19
d. Méthode d'IsoData : .....	20
<b>Chapitre 2 : 2<sup>ème</sup> méthode de segmentation (K-Means).....</b>	<b>21</b>
1. Principe du K-Means : .....	21
2. Pseudo Algorithme : .....	21
3. Analyse détaillée : .....	21
4. Entrées / Sorties : .....	22
5. Classes et méthodes : .....	22
6. Simulation du K-Means : .....	23
<b>Chapitre 3 : Détection de contour « Marching Squares ».....</b>	<b>24</b>
1. Introduction : .....	24
2. Principe : .....	24
3. Pseudo-algorithme : .....	24

4. Analyse détaillée :	24
5. Entrées et sorties :	25
<b>Chapitre 4 : Filtres et opérations morphologiques</b>	26
1. Introduction :	26
2. Prétraitement :	26
a. Principe du filtrage :	26
b. Rôle des filtres :	26
c. Matrice de convolution :	26
d. Filtre médian :	27
e. Filtre gaussien :	27
f. Filtre bilatéral :	28
g. Filtre de rehaussement de contraste :	28
3. Post-traitement :	29
a. Opérations morphologiques :	29
b. Conception détaillée des opérations morphologiques :	29
- L'élément structurant :	29
- Entrées et sorties des algorithmes :	29
- Pseudo-algorithmes :	30
<b>Chapitre 5 : Extraction des caractéristiques</b>	32
1. Introduction :	32
2. La classe Regionprops :	32
3. Les entrées et sorties de la classe :	34
<b>Partie III : Réalisation</b>	36
<b>Chapitre 1 : Langages d'implémentation et outils</b>	36
1. Le langage Python :	36
2. L'éditeur PyCharm :	36
3. PyQt :	36
4. Les bibliothèques libres :	37
Scipy :	37
Numpy :	37
Matplotlib :	37
OpenCV :	37
Anaconda :	37
<b>Chapitre 2 : Interface Utilisateur</b>	38
<b>Chapitre 3 : Tests et Résultats</b>	39
1. Acquisition d'image :	39

2. Prétraitement : .....	39
3. Segmentation des images :.....	40
4. Operations morphologiques :.....	41
5. Suivi des contours : .....	41
6. Analyse des régions :.....	42
Gestion du projet .....	44
Conclusion générale .....	45
Bibliographie et webographie.....	46
Table des figures .....	47

## INTRODUCTION GENERALE

L'extraction de l'information caractéristique contenue dans une image est plus connue sous le nom de « description structurelle », celle-ci n'est rien d'autre que le résultat de l'analyse d'une image, et l'objectif principal de notre projet. Elle permet une description des entités contenues dans l'image, et fait appel, essentiellement, à la segmentation.

Grâce à la segmentation, nous avons pu tirer cette information structurelle que l'œil peut effectuer naturellement. Pour cela, il existe plusieurs approches permettant une telle description. Parmi ces approches, nous nous intéresserons à l'approche région.

La segmentation par région, comme son nom l'indique, a pour objectif de partitionner l'image en plusieurs régions, présentant une homogénéité selon un certain critère, on va donc tenter d'associer à chaque pixel de l'image un label en s'appuyant sur l'information portée (Niveau de gris ou couleur). Le domaine étant vaste, il existe un très grand nombre de méthodes permettant d'effectuer une segmentation par région. Toutes ces méthodes visent à l'extraction des indices visuels. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. Etant donnée une image, il existe toujours plusieurs segmentations possibles. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu.

A travers l'accomplissement de ce projet, nous nous proposons de répondre à ces problématiques :

- Quelles sont les méthodes de segmentation par région les plus connues ? Quels sont les avantages et inconvénients de chacune d'elles ?
- Quelles sont les étapes à suivre pour obtenir de bons résultats avec une segmentation par région ?
- Comment obtenir une image étiquetée à la fin, contenant toutes les caractéristiques dont on a besoin ?

En ce qui concerne les finalités de ce projet, on en distingue plusieurs :

- Initiation au milieu professionnel à travers la réalisation d'un projet informatique dans des délais rappelant les conditions de travail en entreprise.
- Mettre en application des notions techniques vues en cours durant toute la durée du cycle préparatoire.
- Segmenter une image selon l'approche région, et extraire ses caractéristiques.

Afin d'arriver à ces finalités, nous aurons besoin de réaliser les objectifs que nous nous sommes fixés en se basant sur le cahier des charges :

- Choisir deux méthodes de segmentation par région et les implémenter.
- Comprendre et implémenter un processus de détection de contour.
- Sélectionner les filtres et opérations de traitements de l'image pour de meilleurs résultats.
- Analyser les régions et extraire les caractéristiques dont on a besoin.
- Concevoir et implémenter une interface moderne et intuitive permettant d'exploiter les fonctionnalités des algorithmes développés.

## Partie I : Etat de l'art

*Avant d'entamer la présentation de notre solution, il est nécessaire de présenter un état de l'art des méthodes de segmentation par région, où l'on abordera quelques notions théoriques qui nous seront utiles par la suite, notamment pour appuyer notre choix des deux méthodes de segmentation qui représentent le noyau de notre conception.*

## Partie I : ETAT DE L'ART

### Chapitre 1 : GENERALITES SUR L'IMAGERIE

#### 1. La notion d'image :

Au sens étymologique, le mot « image », découlant du latin imago, désigne la représentation visuelle d'un objet par différents moyens ou supports (Dessin, photographie, peinture, sculpture...). L'image d'un objet peut alors être interprétée comme un ensemble de points sur une surface plane.

#### 2. La création des images :

Un moyen simple pour créer une image consiste en l'utilisation de capteurs comme les caméras et appareils photographiques. On distingue deux types de capteurs pour l'acquisition d'images :

- **Les capteurs passifs** : qui captent directement le rayonnement lumineux émis par une source de la scène.
- **Les capteurs actifs** : qui émettent un signal (Par exemple une onde électromagnétique ou ultrasonore irradiant la scène).

#### 3. Image numérique :

Une image numérisée correspond à une image acquise, puis traitée de façon à être stockée sous forme binaire. Le stockage de l'image numérique ainsi obtenue peut s'effectuer sous différents formats (JPEG, BMP, TIFF, PNG, GIF...). Ils correspondent à des mécanismes de compression différents. Cette image restituée est composée d'éléments communément appelés « pixels » (« Picture » + « Élément »).

**Pixel** : L'ordinateur gère les images en tant que des tableaux numériques. Ainsi, il est nécessaire de représenter des images en tableaux de deux dimensions. Un point sur la grille 2-D est appelé un pixel qui représente l'éclairement énergétique à la position de la grille correspondante. La position du pixel est donnée dans la notation commune pour les matrices.

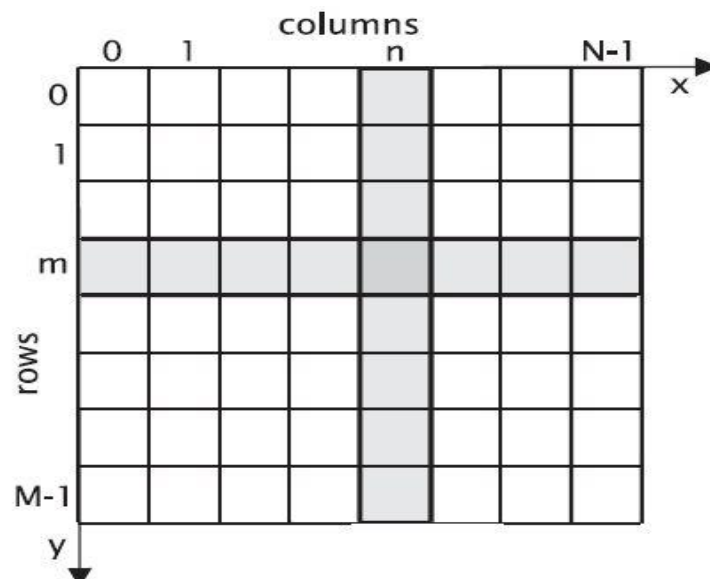


Figure I.1 - La représentation d'une image numérique par un tableau 2D avec la précision d'un pixel de coordonnées (n,m).

Dans notre projet, nous avons traité seulement 3 types d'images numériques :




Type d'image	Description	Codage du pixel	Exemple
<b>Image en couleur</b>	Une image en couleur correspond à la synthèse additive de 3 images en rouge, vert et bleu.	3xN bits	
<b>Image en niveau de gris</b>	Dans ce type d'image, on associe à chaque pixel les valeurs de 0 (Noir) à 255 (Blanc).	8 bits	
<b>Image binaire</b>	C'est une image composée de deux couleurs. On peut donc la coder avec des 0 (Noir) et des 1 (Blanc).	1 bit	

Tableau I.1 - Tableau qui regroupe les différents types d'images que nous avons traité avec leurs codages.

#### 4. Conclusion :

Il est nécessaire de comprendre d'où vient et comment faire une image ainsi que sa structure et ses différents types pour qu'on puisse la manipuler facilement.

Nous avons constaté que l'ordinateur traite l'image comme une matrice à deux dimensions, ce qui va nous permettre de manipuler tout ce qui est valeur, type et codage de chaque pixel de l'image.



## Chapitre 2 : SEGMENTATION DES IMAGES :

La phase de numérisation effectuée, vient ensuite une phase plus complexe d'interprétation de l'image. Elle a pour but d'extraire des informations sur les objets et de décrire leurs caractéristiques.

### 1. Définition :

La segmentation d'une image consiste à la partitionner en plusieurs régions homogènes selon un critère prédéfini. Son but est d'extraire les informations essentielles de l'image.

D'une manière rigoureuse, segmenter une image  $I$  revient à la partitionner en  $n$  sous-ensembles  $R_1, R_2, R_3, \dots, R_n$  tels que :

$$\begin{cases} I = \bigcup_{i=1}^{i=n} R_i \\ R_i \cap R_j = \emptyset \quad \forall i \neq j \end{cases}$$

On peut classer les différentes méthodes de segmentation en deux principales catégories comme illustré ci-dessous.

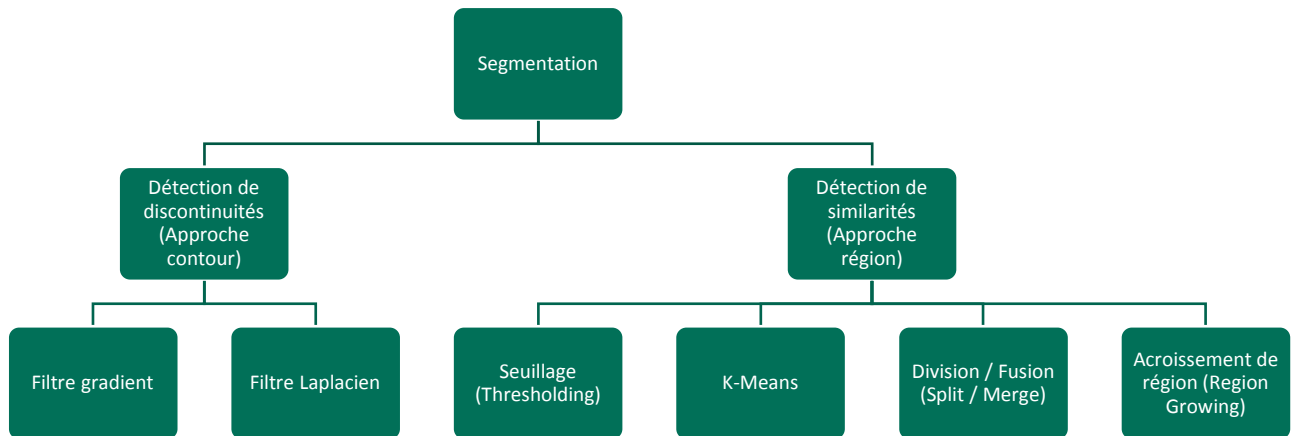


Figure I.2 - Schéma représentant les différentes méthodes de segmentation.

### 2. Applications :

La segmentation des images est utilisée dans beaucoup de domaines, parmi ses applications, on peut citer :

- Le domaine médical (Localiser les tumeurs, étudier les maladies neurologiques et autres pathologies...).
- La détection des objets (Dans des images satellites par exemple).
- La vidéo-surveillance.
- La biométrie (Reconnaissance faciale, d'iris ou d'empreinte digitale).

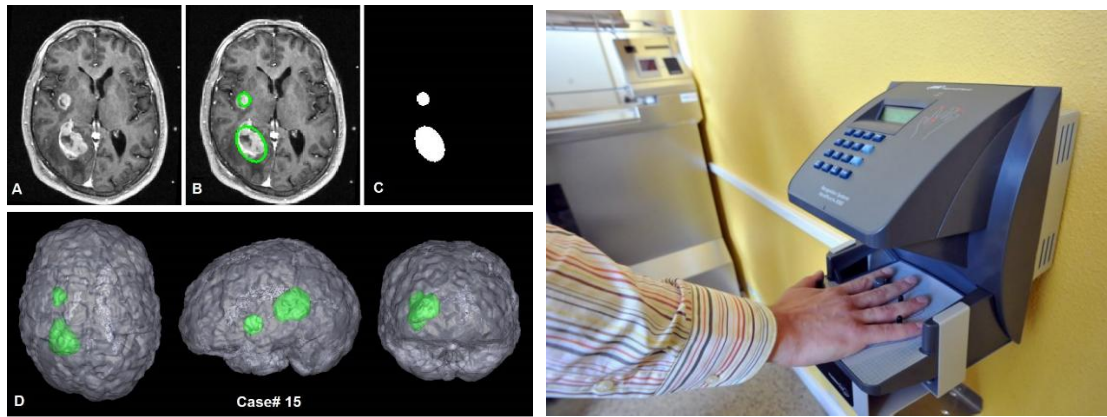


Figure I.3 - Localisation de tumeurs (A gauche) et reconnaissance d'empreintes digitales (A droite).

### 3. Différentes méthodes et procédés de segmentation :

La segmentation d'images est un domaine très vaste où l'on trouve une multitude de méthodes et d'algorithmes répartis en deux grandes catégories : Détection de discontinuités et détection de similarités. Etant donné que notre projet est essentiellement axé sur la segmentation par région, nous allons nous focaliser dessus et l'approche contour ne sera définie que brièvement.

#### a. Détection de similarités (Approche région) :

Cette approche consiste à créer des régions similaires selon un critère d'homogénéité (En termes de couleur, d'intensité...).

On distingue deux catégories :

- Les méthodes qui tiennent compte de la disposition spatiale des pixels (Exemples : Croissance de régions, division / fusion).
- Et celles qui n'en tiennent pas compte (Exemples : Classification, seuillage).

Méthodes	Principe	Avantages	Inconvénients
<b>Seuillage</b>	Diviser une image (En niveaux de gris) en plusieurs classes en utilisant des histogrammes qui doivent avoir des pics distincts.	- Méthode simple et rapide.	- Dans certains algorithmes de seuillage, il faut connaître le nombre de classes.
<b>K-Means</b>	Choisir « k » centres de gravité aléatoirement et affecter à chaque pixel la classe la plus proche (En intensité), puis recalculer les centres de gravité jusqu'à ce que leur valeur reste stable.	- Lorsque le paramètre « k » est petit, l'algorithme est rapide.	- Il faut déterminer le nombre de classes (Paramètre « k »).

Tableau I.2 - Méthodes de segmentation par région ne tenant pas compte de la disposition spatiale des pixels.

Méthodes	Principe	Avantages	Inconvénients
<b>Division / Fusion</b>	<p><b>Division :</b> Diviser l'image successivement en 4 blocs. Ces blocs sont représentés dans une structure d'arbre (QuadTree) de graphe (Graphe d'adjacence).</p> <p><b>Fusion :</b> Calculer le critère de similarité de chaque bloc et de fusionner les blocs voisins en parcourant le graphe d'adjacence.</p>	<ul style="list-style-type: none"> <li>- Méthode à la fois globale et locale : globale lors de la division et locale lors de la fusion.</li> <li>- Moins sensible au bruit que l'accroissement de région.</li> </ul>	<ul style="list-style-type: none"> <li>- Complexité des algorithmes : Manipulation de structures de données.</li> <li>- Découpage en régions non fidèle à l'image originale.</li> </ul>
<b>Accroissement de région</b>	Choisir des « seeds », un ensemble de points (Pixels) à partir desquels on pourra construire les régions en respectant un certain critère d'homogénéité.	<ul style="list-style-type: none"> <li>- Facile à mettre en œuvre (Lorsqu'on a les coordonnées des germes).</li> </ul>	<ul style="list-style-type: none"> <li>- L'influence du choix des germes initiaux.</li> <li>- Sensibilité au bruit.</li> </ul>

Tableau I.3 - Méthodes de segmentation par région tenant compte de la disposition spatiale des pixels.

La segmentation par région est suivie d'une étape très importante permettant une meilleure extraction des objets :

- **Analyse morphologique :**

Cette analyse morphologique se traduit par un ensemble d'opérations que l'on effectue sur l'image binaire obtenue après la segmentation. Elle vise à éliminer le bruit et régulariser les formes des objets de l'image. Les opérateurs morphologiques principaux sont

- **Erosion :** Elle permet d'éliminer les pixels isolés et de réduire la taille de l'image.
- **Dilatation :** Elle permet d'éliminer les trous dans les objets constituant l'image en dilatant les formes.
- **Ouverture :** C'est un opérateur composé, il correspond à une érosion suivie par une dilatation. L'ouverture permet d'adoucir les bords des formes tout en conservant leur taille.
- **Fermeture :** C'est un opérateur composé, il correspond à une dilatation suivie par une érosion. La fermeture permet d'éliminer les trous tout en conservant la taille des formes.

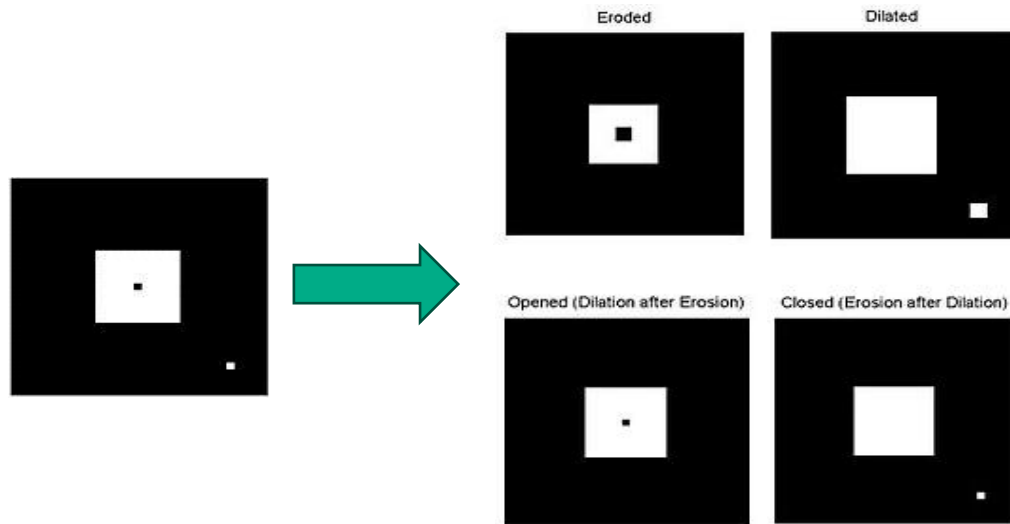


Figure I.4 - Application d'opérateurs morphologiques sur une image binaire.

**b. Détection des discontinuités (Approche contour) :**

Un contour est un ensemble de pixels entourant un objet, il correspond à un changement brusque d'intensité. Les algorithmes de cette approche ne cherchent donc plus à trouver les régions elles-mêmes mais plutôt les frontières les délimitant.

Il existe un grand nombre de méthodes pour détecter les contours, parmi les plus utilisées :

- **Filtre gradient :**

On utilise les dérivées premières (Puisque les contours représentent une variation d'intensité). L'opérateur gradient agit comme un filtre passe-haut d'où la forte sensibilité au bruit, c'est pour cette raison qu'une opération de lissage est intégrée dans les filtres de Prewitt, Sobel, Roberts....

Chacun de ces filtres est défini par sa matrice de convolution que l'on doit appliquer à l'image qu'on veut filtrer (Elle peut être verticale, horizontale, diagonale...).

- **Filtre Laplacien :**

Dans ce type de filtres, on utilise les dérivées secondes pour mettre en évidence les contours. Il suffit donc d'appliquer la matrice de convolution du filtre sur l'image.

Remarque : Dans la plupart des méthodes de détection de contours, il est nécessaire de fermer les contours (Grâce à une opération morphologique par exemple) pour pouvoir délimiter les objets.

#### 4. Conclusion :

Nous avons vu tout au long de ce chapitre une panoplie de méthodes de segmentation, et chacune d'elle possède des avantages et des inconvénients. De ce fait, il nous est possible d'affirmer qu'il n'existe pas d'algorithme parfait, il faut donc prendre en considération plusieurs paramètres tels que la nature de l'image, le contexte informatique....

### Chapitre 3 : ANALYSE DE REGIONS ET EXTRACTION DES CARACTERISTIQUES

Les opérations et algorithmes morphologiques, ainsi que les procédés de segmentation, permettent de séparer les objets d'intérêt dans l'image.

Il s'agit maintenant de décrire géométriquement ces objets (Description de forme, caractéristiques géométriques d'un objet, approximation de forme).

Parmi ces caractéristiques, on peut citer :

- **La surface (Ou aire) :** C'est-à-dire le nombre de pixels qui la composent.
- **Le périmètre :** La longueur de la frontière de la région ou le nombre de pixels formant cette frontière.
- **Le diamètre équivalent :** La longueur du diamètre du cercle ayant la même surface que la région étudiée.
- **La longueur de l'axe principal d'inertie :** C'est la longueur de la ligne la plus longue traversant l'objet.
- **L'angle de l'axe principal d'inertie :** C'est l'orientation de cet axe.

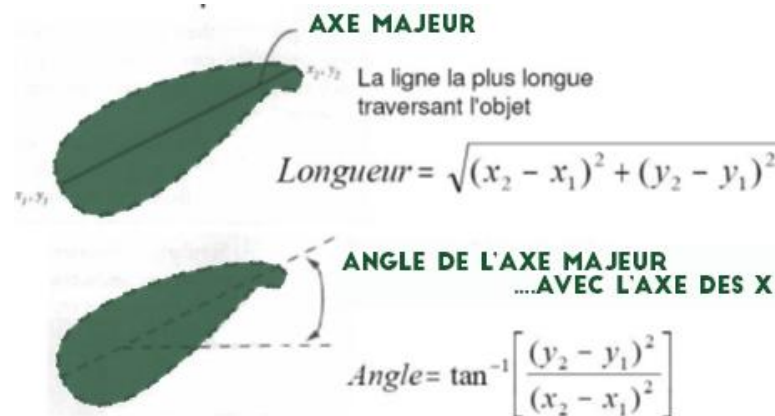


Figure I.5 - Axe majeur d'inertie et orientation.

- **La longueur du second axe d'inertie :** C'est la ligne la plus longue au travers de l'objet qui est perpendiculaire à l'axe majeur.

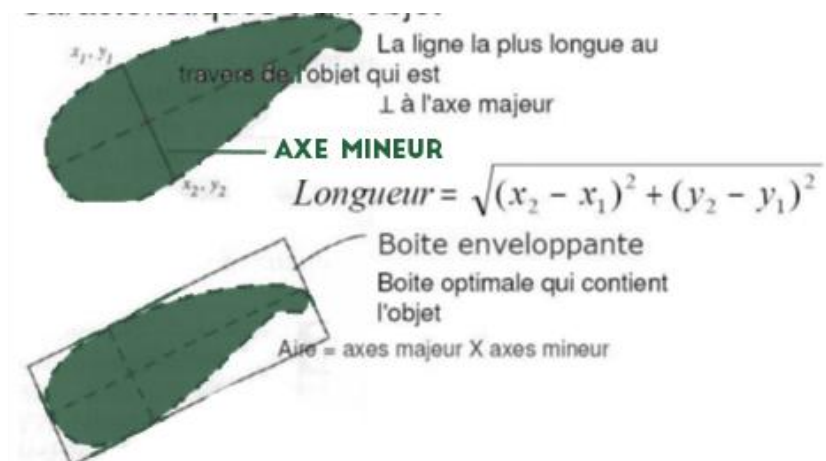


Figure I.6 - Axe mineur d'inertie.

- **Les coordonnées du centre de gravité :** Pour le calculer, il suffit d'utiliser les coordonnées en x et y de chaque point appartenant à une région et le nombre de points total de la région. Calculer la somme des coordonnées en x pour chaque point et diviser par le nombre total de points de la région. Même opération pour les y. Les coordonnées trouvées représentent le centre de gravité de l'objet.

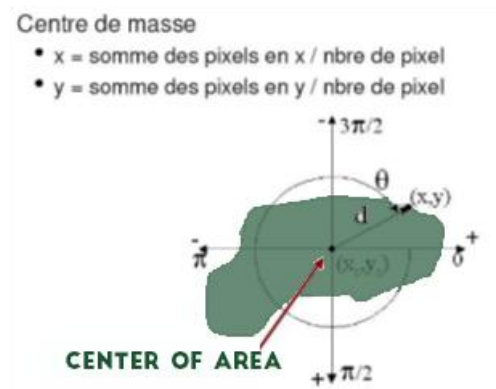


Figure I.7 - Centre de gravité d'une région.

### Conclusion :

On constate enfin que l'objectif du processus de traitement d'images est de proposer à l'utilisateur un certain nombre d'indices et d'informations sur les objets qui ont été segmentés.

## Partie II : Conception de la solution

*Nous commencerons par présenter les deux méthodes de segmentation sur lesquelles s'est porté notre choix, à savoir : la segmentation par seuillage et la segmentation de K-Means. En premier lieu et pour chacune des méthodes, nous présenterons le principe ainsi qu'une brève définition, par la suite nous verrons le pseudo-algorithme et l'analyse détaillée de ce dernier.*

*Nous aborderons brièvement la méthode Marching Squares utilisée pour la détection des contours, conformément au travail demandé dans le cahier des charges. En outre, le pseudo-algorithme que nous détaillerons par l'analyse.*

*Etant donné que l'obtention de bons résultats lors de la segmentation des images est fortement liée aux filtres et opérations morphologiques appliqués avant et après la segmentation, il est donc nécessaire d'aborder : Les prétraitements, dont les filtres que nous présenterons un par un, mais aussi les post-traitements qui représentent les opérations morphologiques telles que l'érosion, la dilatation, l'ouverture ou encore la fermeture.*

*Pour finir, nous exposerons la partie extraction des caractéristiques, où toutes les méthodes utilisées seront décrites.*

## Partie II : CONCEPTION DE LA SOLUTION

Avant d'entamer cette partie, le digramme des classes utilisées est présenté ci-dessous : (Les classes seront détaillées plus loin)

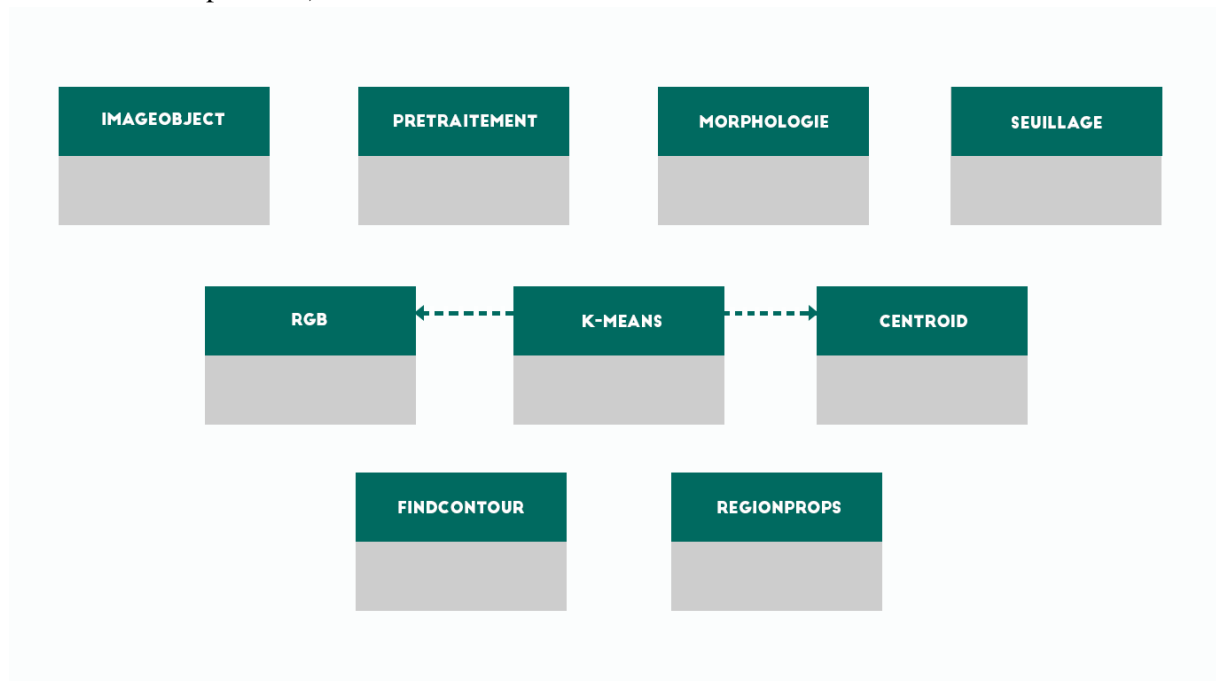


Figure II.1 - Diagramme des classes.

### Chapitre 1 : 1<sup>ERE</sup> METHODE DE SEGMENTATION (SEUILLAGE)

#### 1. Définition :

Le seuillage est une technique de segmentation très simple, elle consiste à diviser l'image en  $n$  régions (Classes) correspondant à un intervalle de niveau de gris selon un ou plusieurs seuils, et ce à partir de l'histogramme.

L'objectif de notre application est de « binariser » une image en niveau de gris. La binarisation est un cas particulier du seuillage ; Il s'agit d'attribuer la valeur 255 (Blanc) aux pixels pertinents (Qui représentent les objets de l'image) et la valeur 0 (Noir) aux autres pixels (Qui représentent le fond de l'image). En d'autres termes :

$$g(x,y) = \begin{cases} 0 & \text{si } l(x,y) < t \\ 255 & \text{si } l(x,y) \geq t \end{cases}$$

Où :

$l(x,y)$  est le niveau de gris (Luminance) d'un pixel de coordonnées  $(x,y)$ .

$t$  est le seuil appliqué à l'image.

Il existe beaucoup de méthodes permettant de calculer ce seuil automatiquement, et le principe reste le même : Trouver le seuil permettant d'optimiser une fonction critère, c'est-à-dire trouver la valeur permettant de binariser au mieux l'image à partir d'une fonction définie par chaque méthode. Nous avons choisi 4 de ces méthodes :





Figure II.2 - Les méthodes de seuillage utilisées.

## 2. Pseudo-algorithme général :

D'une façon générale, la binarisation suit les étapes suivantes :

### Algorithme Général

- Convertir l'image RGB en niveau de gris.
- Calculer l'histogramme.
- Calculer le seuil grâce à l'une des 4 méthodes choisies.
- Appliquer le seuil sur l'image.

## 3. Analyse détaillée :

- Conversion d'une image RGB en niveau de gris :

Les images étant souvent en mode RGB (Red / Green / Blue), il faut les convertir en niveau de gris pour appliquer le seuillage.

La conversion se fait en remplaçant, pour chaque pixel de l'image, les trois valeurs représentant les niveaux de rouge, vert et bleu, en une seule valeur représentant la luminosité (Le niveau de gris) selon la formule suivante :

$$Gris = 0.2989 \times Rouge + 0.5870 \times Vert + 0.1140 \times Bleu$$

- Calcul de l'histogramme :

L'histogramme est calculé en parcourant toute l'image et en incrémentant le nombre de pixels pour chaque intensité (Allant de 0 à 255) s'il y a correspondance.

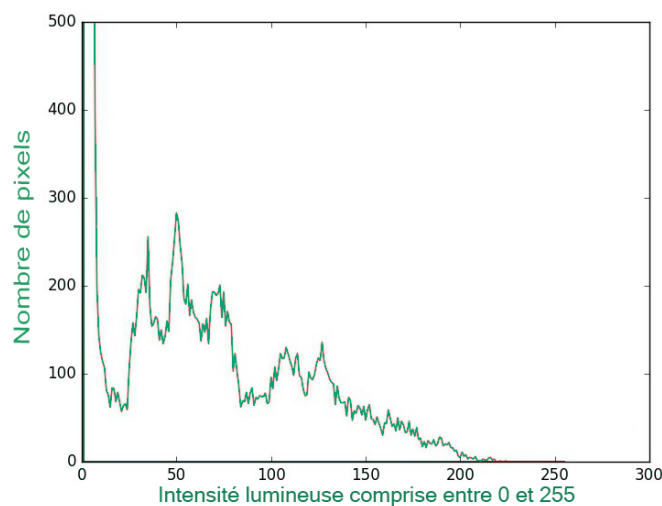


Figure II.3 - Histogramme d'une image.

- Calcul du seuil :

Le seuil sera calculé grâce aux 4 méthodes que l'on a choisies (Otsu, Li, Yen et IsoData). Nous allons expliquer chacune de ces méthodes plus loin.

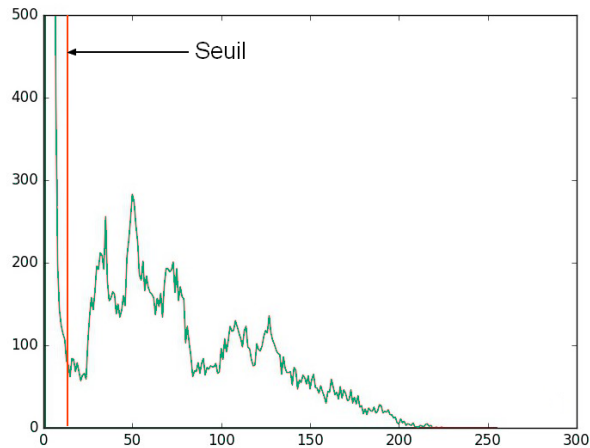


Figure II.4 - Seuil calculé à partir de l'histogramme.

- Application du seuil sur l'image :  
Une fois le seuil  $t$  calculé, on modifie la valeur de l'intensité de chaque pixel de l'image selon ce  $t$ .
  - Si l'intensité est inférieure au seuil  $t$ , on affecte au pixel la valeur 0.
  - Sinon, on lui affecte la valeur 255.

#### 4. Calcul du seuil :

##### a. Méthode d'Otsu :

##### - Principe :

L'algorithme d'Otsu est basé sur la maximisation de la variance interclasse à partir de l'histogramme de l'image (C'est-à-dire trouver parmi toutes les valeurs possibles le seuil qui va rendre la variance interclasse maximale).

La variance interclasse est une notion en statistique et en probabilité qui exprime le degré de séparation de deux classes (Ici, les deux classes concernées sont :  $C_1$  la classe du fond et  $C_2$  la classe des objets), son expression est donnée par :

$$\sigma^2(t) = \omega_1(t) \cdot \omega_2(t) \cdot [\mu_2(t) - \mu_1(t)]^2$$

Où :

$t$  est le seuil.

$\omega_1, \omega_2$  sont les fréquences des deux classes.

$\mu_1, \mu_2$  sont les moyennes des deux classes.

##### - Pseudo-algorithme :

L'algorithme d'Otsu suit les étapes suivantes :

### Algorithme Otsu

- Parcourir tous les seuils possibles  $t$  de 0 jusqu'à l'intensité maximale de l'image.
  - Calculer  $\omega(t)$  et  $\mu(t)$ .
  - Calculer  $\sigma^2(t)$ .
- Le seuil désiré correspond à la variance interclasse  $\sigma^2(t)$  maximale.

#### - Analyse détaillée :

- Parcours de tous les seuils possibles et calcul de la variance interclasse correspondante :  
Les valeurs possibles du seuil  $t$  varient entre 0 et l'intensité maximale de l'image  $L$  (Dans le pire des cas, ils varieront entre 0 et 255). Pour chacune de ces valeurs, il faudra :
  - Calculer les fréquences des deux classes  $C_1$  et  $C_2$  séparées par le seuil  $t$ .  
Les fréquences sont calculées comme suit :

$$\omega_1 = \sum_{i=0}^{i=t-1} \frac{n_i}{N}$$

$$\omega_2 = \sum_{i=t}^{i=L} \frac{n_i}{N}$$

Où :

$N$  est le nombre total de pixels.

$n_i$  est le nombre de pixels d'intensité  $i$ .

- Calculer les moyennes des deux classes  $C_1$  et  $C_2$  séparées par le seuil  $t$ .  
Les moyennes sont calculées comme suit :

$$\mu_1 = \sum_{i=0}^{i=t-1} i \cdot \frac{n_i}{N \cdot \omega_1}$$

$$\mu_2 = \sum_{i=t}^{i=L} i \cdot \frac{n_i}{N \cdot \omega_2}$$

- Calculer la variance interclasse correspondant au seuil  $t$ .
- Trouver le seuil qui maximise la variance interclasse :  
Le seuil optimal est obtenu en utilisant la fonction *argmax* qui cherche la valeur maximale parmi toutes les variances calculées.

#### b. Méthode de Li (Cross-Entropie) :

##### - Principe :

La méthode de Li consiste à minimiser la cross-entropie entre l'image originale et sa version binaire, c'est-à-dire trouver le seuil qui fera en sorte que la perte d'information soit la moins grande possible.

L'entropie est un concept mesurant l'incertitude d'une information ; Plus l'entropie d'une image est élevée, plus cette image est « aléatoire », une entropie nulle caractérise donc une image uniforme qui possède une seule couleur.

L'expression de la cross-entropie des classes  $C_1$  (Fond de l'image) et  $C_2$  (Objets de l'image) est donnée par :

$$H(t) = H_1(t) + H_2(t) = \sum_{i=0}^{i=t-1} i.n_i . \log \frac{i}{\mu_1} + \sum_{i=t}^{i=L} i.n_i . \log \frac{i}{\mu_2}$$

Où :

$t$  est le seuil,  $L$  est l'intensité maximale de l'image.

$\mu_1, \mu_2$  sont les moyennes des deux classes.

$n_i$  est le nombre de pixels ayant l'intensité  $i$ .

- Pseudo-algorithme :

L'algorithme de Li suit les étapes suivantes :

**Algorithme Li**

- Parcourir tous les seuils possibles  $t$  de 0 jusqu'à l'intensité maximale de l'image.
  - Calculer  $\mu(t)$ .
  - Calculer  $H(t)$ .
- Le seuil désiré correspond à la cross-entropie minimale.

- Analyse détaillée :

- Parcours de tous les seuils possibles et calcul de la cross-entropie correspondante :  
Les valeurs possibles du seuil  $t$  varient entre 0 et l'intensité maximale de l'image  $L$ . Pour chacune de ces valeurs, il faudra :
  - Calculer les moyennes  $\mu_1, \mu_2$  des deux classes  $C_1$  et  $C_2$  séparées par le seuil  $t$ .
  - Calculer la cross-entropie  $H(t)$  de chaque seuil.
- Trouver le seuil qui minimise la cross-entropie :  
Le seuil optimal correspond à la plus petite valeur de toutes celles calculées précédemment, on utilisera donc la fonction *argmin*.

c. Méthode de Yen (Corrélation entropique) :

- Principe :

Cette méthode repose sur la maximisation de la corrélation entropique entre les classes  $C_1$  et  $C_2$ , c'est-à-dire trouver le seuil pour lequel l'image segmentée soit la plus fidèle possible à l'image originale.

La corrélation entropique associée aux deux classes est donnée par les relations suivantes :

$$R(t) = R_1(t) + R_2(t) = -\ln \sum_{i=0}^{i=t-1} \left(\frac{p_i}{P_1}\right)^2 - \ln \sum_{i=t}^{i=L} \left(\frac{p_i}{P_2}\right)^2$$

Où :

$t$  est le seuil.

$P_1, P_2$  sont les fréquences des deux classes.

$p_i$  est la fréquence de l'intensité  $i$ .

- Pseudo-algorithme :

L'algorithme de Yen suit les étapes suivantes :

**Algorithme Yen**

- Parcourir tous les seuils possibles  $t$  de 0 jusqu'à l'intensité maximale de l'image.
  - Calculer  $P_1, P_2$ .
  - Calculer  $R(t)$ .
- Le seuil désiré correspond à la corrélation entropique.

- Analyse détaillée :

- Parcours de tous les seuils possibles et calcul de la corrélation entropique correspondante : Les valeurs possibles du seuil  $t$  varient entre 0 et l'intensité maximale de l'image  $L$ . Pour chacune de ces valeurs, il faudra :
  - Calculer les fréquences des deux classes  $P_1$  et  $P_2$  séparées par le seuil  $t$ .

$$P_1 = \sum_{i=0}^{i=t-1} p_i$$

$$P_2 = \sum_{i=t}^{i=L} p_i$$

- Calculer la corrélation entropique  $R(t)$  de chaque seuil.
- Trouver le seuil qui maximise la corrélation entropique :  
Le seuil correspond à la plus grande valeur de toutes celles calculées précédemment.

d. Méthode d'IsoData :

- Principe :

Cette méthode permet de trouver un seuil à partir de l'histogramme en le séparant de façon itérative en deux classes jusqu'à convergence, c'est-à-dire jusqu'à ce que le seuil reste constant.

- Pseudo-algorithme :

L'algorithme d'IsoData suit les étapes suivantes :

**Algorithme IsoData**

- Initialiser le seuil  $t$  à l'intensité moyenne de l'image.
- Tant que  $t_i$  est différent de  $t_{i-1}$  :
  - Diviser l'histogramme en deux classes selon la valeur de  $t$ .
  - Calculer les intensités moyennes  $\mu_1, \mu_2$  des deux classes  $C_1$  et  $C_2$ .
  - Calculer la nouvelle valeur du seuil  $t$  avec :
$$t = \frac{\mu_1 + \mu_2}{2}$$
  - Comparer  $t_i$  et  $t_{i-1}$ , si elles sont égales, l'algorithme s'arrête et le seuil retourné est  $t_i$ .

## Chapitre 2 : 2<sup>EME</sup> METHODE DE SEGMENTATION (K-MEANS)

---

### 1. Principe du K-Means :

Le but principal de cet algorithme, est de servir de méthode de segmentation d'images en séparant ses différentes régions ou classes de pixels. En appliquant les règles de base de cet algorithme, l'algorithme décèle les centres des classes dont le nombre est prédéfini. Pour ce type d'algorithme, il faut toujours définir le nombre de classes avant le début du traitement. En premier lieu, un nombre de pixels équivalent au nombre de classes est déterminé, ces pixels seront considérés comme étant les centres des classes initiales. Ensuite, chaque pixel est mis dans la classe ayant le centre le plus proche. On calcule les centres de gravité de ces nouveaux ensembles qui sont considérés à leur tour comme étant les nouveaux centres des classes et on réitère l'opération. On répète ce processus jusqu'à ce qu'il y ait stabilité des classes et de leur centre.

### 2. Pseudo Algorithme :

Les étapes de l'algorithme K-Means sont détaillées par le pseudo-algorithme :

#### Algorithme K-Means

1. Initialiser les k centres.
2. (Ré-) Affecter les pixels aux classes.
3. Calculer les nouveaux centres selon l'affectation résultante en utilisant la formule.
4. Vérifier le critère d'arrêt : Si vérifié : Fin.  
Sinon : Aller à (2).

### 3. Analyse détaillée :

- Initialisation : L'initialisation des k centres peut se faire de trois manières différentes :
  - Manuellement : L'utilisateur a la liberté de choisir manuellement les k centres initiaux dans l'image ; et selon le nombre et les valeurs des centres choisis, la segmentation est faite. Le but de cette option est d'offrir à l'utilisateur la possibilité de choisir les classes selon son expertise et sa vision personnelle.
  - Aléatoirement : Un nombre de classes k est donné en entrée puis les k centres sont choisis aléatoirement par rapport à leurs valeurs de niveau de gris qui doivent être suffisamment dissimilaires.
  - Par histogramme : Chaque pic détecté dans l'histogramme correspond à un centre d'une classe.

Pour ce projet, nous utilisons une initialisation aléatoire.

- (Ré-) Affectation : Ayant défini les centres, l'affectation ou la réaffectation des pixels s'effectue selon la distance euclidienne entre le pixel et le centre. Le pixel est donc affecté à la classe dont la distance est minimale.
- Calcul des nouveaux centres : La valeur du nouveau centre est trouvée en calculant la valeur moyenne des pixels appartenant à la classe.
- Critère d'arrêt : Le critère d'arrêt est la stabilité de tous les pixels (Ou quand le nombre d'itérations est égal au maximum qu'on a fixé à 10).

#### 4. Entrées / Sorties :

Les entrées de l'algorithme sont l'image en couleurs (RGB) à segmenter et le nombre de classes qu'on veut (Le « k ») et qu'on a fixé à deux pour ce projet. Les sorties sont l'image binaire et un tableau de coordonnées finales des centres.

#### 5. Classes et méthodes :

Nous avons défini les classes suivantes :

- Classe RGB :

Nous aurons besoin de créer les pixels RGB de l'image à segmenter pour construire les objets Centroid.

- Classe Centroid :

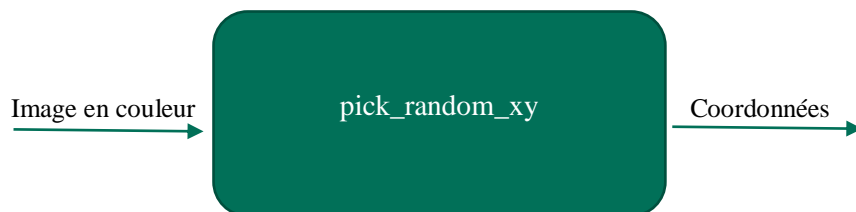
Nous aurons aussi besoin de déterminer le centre des régions obtenues à chaque étape de l'algorithme.

- Classe K-means :

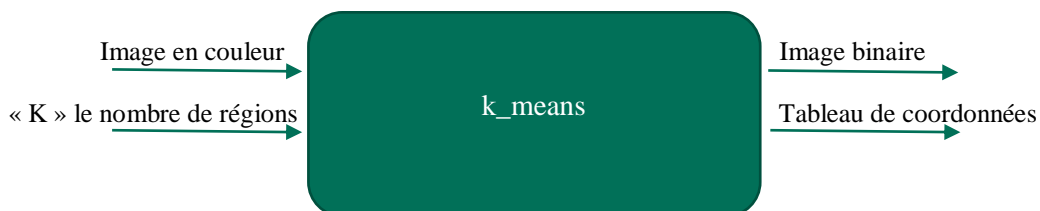
La classe contient les méthodes suivantes :



Rôle : Calcule la différence en couleur pour chaque pixel avec les pixels centres.



Rôle : Retourne des coordonnées de pixels aléatoirement.



Rôle : Effectue une segmentation de l'image par région suivant le paramètre k et renvoie un tableau de coordonnées finales des centres.

## 6. Simulation du K-Means :

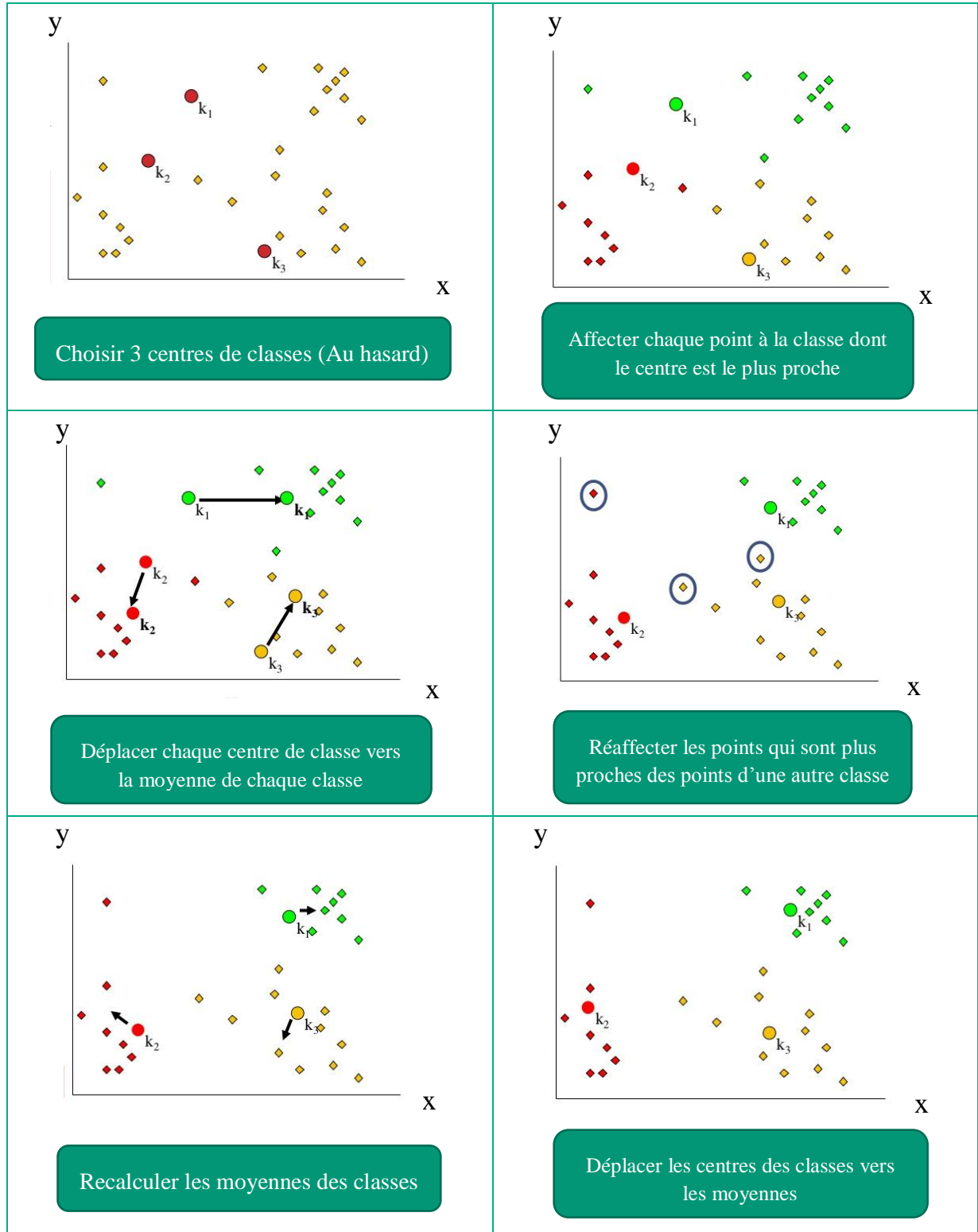


Tableau II.1 - Simulation de l'algorithme de K-Means.



## Chapitre 3 : DETECTION DE CONTOUR « MARCHING SQUARES »

### 1. Introduction :

A partir de l'image binaire, nous allons maintenant détecter et dessiner les contours des différents objets pour une meilleure visualisation. Cette partie sera assez brève puisque la détection des contours n'est pas l'objectif principal de notre projet.

### 2. Principe :

Le Marching Squares est un algorithme puissant permettant de générer les contours des objets d'une image. Pour cela, l'image est découpée en plusieurs carrés de même taille, chaque carré est composé de 4 pixels. Un déplacement est effectué en fonction de la configuration du carré et de la valeur seuil introduite. On examine donc toute l'image de façon à ce qu'on « marche » le long du contour de l'objet jusqu'à revenir au point de départ.

Remarque : Dans notre projet, l'image qu'on donne en entrée est une image binaire. Par conséquent, la valeur du seuil importe peu vu que les seules intensités possibles sont le 0 et le 1 ( $0 < \text{Seuil} < 1$ ). Donc le 0 correspondra à l'état bas, et le 1 à l'état haut.

### 3. Pseudo-algorithme :

#### Algorithme du Marching Squares

1. Découper l'image en plusieurs cellules.
2. Pour chaque cellule :
  - Déterminer le cas correspondant (Parmi les 16 cas possibles).
  - Appliquer l'interpolation linéaire pour trouver la position exacte du contour.

### 4. Analyse détaillée :

- Découper l'image en plusieurs cellules :  
Chaque cellule sera composée de 4 pixels, et les 4 valeurs de cette cellule correspondent aux intensités de chaque pixel (Puisqu'il s'agit d'une image binaire, la valeur est soit 0 soit 1).

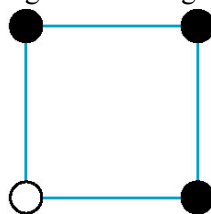


Figure II.5 - Représentation d'une cellule.

- Pour chaque cellule :
  - Déterminer le cas correspondant (Parmi les 16 cas possibles) :

Ces 16 configurations dépendent des intensités de chaque coin (Pixel) de la cellule. Le point noir représente le 0, et le point blanc représente le 1. La figure ci-dessous illustre les différents cas possibles.

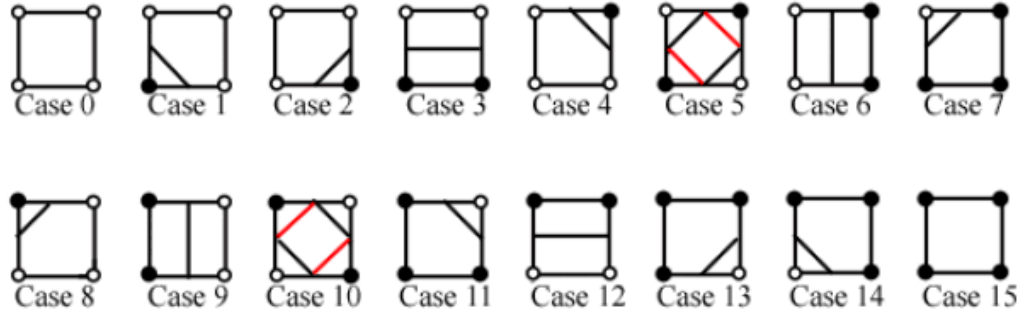


Figure II.6 - Les 16 configurations possibles.

- Appliquer l'interpolation linéaire pour trouver la position exacte du contour :

L'interpolation linéaire permet de déterminer les points exacts de l'intersection avec les arêtes des cellules. Sa formule est la suivante :

$$Px = P_{1x} + (Seuil - V(P_1)) \cdot \frac{P_{2x} - P_{1x}}{V(P_2) - V(P_1)}$$

$$Py = P_{1y} + (Seuil - V(P_1)) \cdot \frac{P_{2y} - P_{1y}}{V(P_2) - V(P_1)}$$

Où :

$Px, Py$  sont les coordonnées du point  $P$  (L'intersection entre l'arête  $[P_1, P_2]$  et la ligne du contour.

$P_{1x}, P_{1y}, P_{2x}, P_{2y}$  sont les coordonnées des points  $P_1$  et  $P_2$ .

$V(P_1), V(P_2)$  sont les poids des points  $P_1$  et  $P_2$ .

Remarque : En ce qui concerne les coefficients, puisqu'on travaille avec une image binaire, ils seront calculés comme suit :

$$Poids = \frac{Niveau\ de\ gris\ du\ pixel}{255}$$

## 5. Entrées et sorties :

En entrée, l'image binaire et le seuil à appliquer.

En sortie, un tableau contenant les coordonnées des contours.

## Chapitre 4 : FILTRES ET OPERATIONS MORPHOLOGIQUES

### 1. Introduction :

Les images telles qu'elles sont permettent rarement de parvenir à une bonne extraction des objets. C'est pour cette raison que des traitements doivent être effectués avant ou après la segmentation de l'image à analyser, et ce, afin d'assurer une meilleure distinction entre les régions.

Nous avons regroupés ces traitements en deux catégories :

- Prétraitement.
- Post-traitement.

### 2. Prétraitement :

#### a. Principe du filtrage :

Le principe du filtrage est de modifier les valeurs des pixels d'une image pour améliorer certaines de ses caractéristiques. Il en existe plusieurs, mais dans notre projet, nous en avons choisi quatre :

- Filtre médian.
- Filtre gaussien.
- Filtre bilatéral.
- Filtre de rehaussement de contraste.

#### b. Rôle des filtres :

Filtres	Rôle
<b>Filtre médian</b>	<ul style="list-style-type: none"> <li>- Réduire le bruit.</li> <li>- Lissage.</li> </ul>
<b>Filtre gaussien</b>	<ul style="list-style-type: none"> <li>- Réduire le bruit.</li> <li>- Appliquer un flou à l'image.</li> </ul>
<b>Filtre bilatéral</b>	<ul style="list-style-type: none"> <li>- Eliminer le bruit tout en conservant les contours</li> </ul>
<b>Filtre de rehaussement de contraste</b>	<ul style="list-style-type: none"> <li>- Accentuer les contours.</li> <li>- Rendre l'image moins terne.</li> </ul>

Tableau II.2 - Tableau qui résume le rôle des filtres utilisés.

#### c. Matrice de convolution :

La plupart des filtres utilisent une matrice de convolution (Egalement appelée « Noyau » ou « Kernel »). C'est un masque de taille impaire  $N$  ( $N > 1$ , par exemple  $3 \times 3$ ) que l'on applique sur l'image qu'on veut filtrer grâce à un produit de convolution.

Le produit de convolution se fait entre l'image et le masque. Prenons par exemple un masque de taille  $3 \times 3$ . Pour chaque pixel  $P$  de l'image, on multiplie sa valeur et celles des 8 pixels qui l'entourent par la valeur correspondante dans le masque, le pixel  $P$  prendra alors la somme de tous les résultats obtenus. Ci-dessous une figure qui illustre cela.

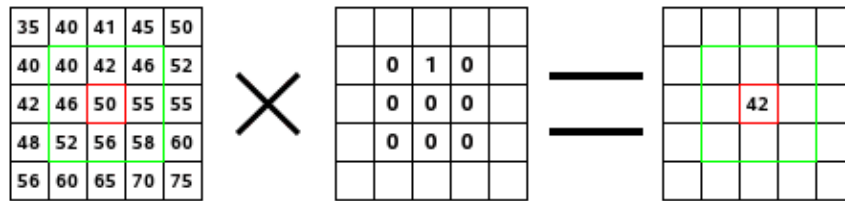


Figure II.7 - Produit de convolution (Exemple).

**NOTA :** Il faut prévoir un traitement spécial pour les pixels situés sur les bords de l'image (Les rangées horizontales et verticales des bords). On pourrait par exemple dupliquer ces rangées pour créer un voisinage à ces pixels bordants et ainsi appliquer le produit de convolution.

#### d. Filtre médian :

Le principe du filtre médian est de remplacer chaque pixel par la valeur médiane de son voisinage.

Dans l'exemple ci-dessous, on prend le voisinage du pixel  $P$  ayant la valeur 200, et on trie les 9 valeurs dans un ordre croissant, la médiane de cette liste de nombres est 82. On remplace donc la valeur du pixel  $P$  par 82. Et on fait cela pour tous les pixels de l'image.

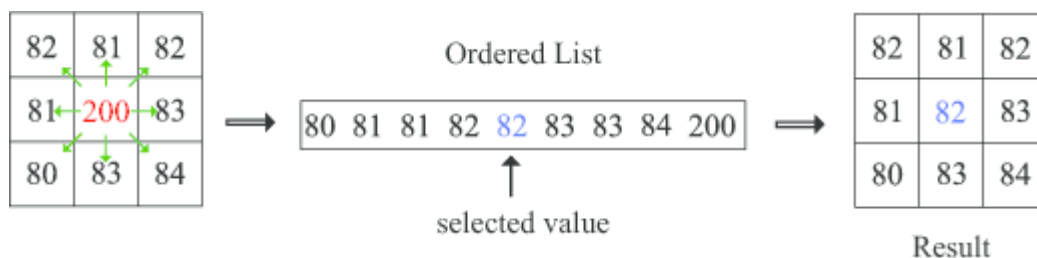


Figure II.8 - Filtre médian (Exemple).

#### e. Filtre gaussien :

Ce filtre est modélisé par une fonction appelée fonction gaussienne<sup>1</sup>.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Où :

$\sigma$  est l'écart-type. Il détermine la largeur de la cloche gaussienne (Figure ...).

$x$  et  $y$  sont des entiers appartenant à un intervalle centré (Qui dépendra de la taille du masque). Avec  $x$  qui représente la distance de l'origine sur l'axe des abscisses, et  $y$  la distance de l'origine sur l'axe des ordonnées.

<sup>1</sup> La fonction gaussienne est la loi normale (En probabilité), mais puisque dans le traitement d'images on manipule des données en deux dimensions, donc la fonction introduite ici est une fonction à deux variables.

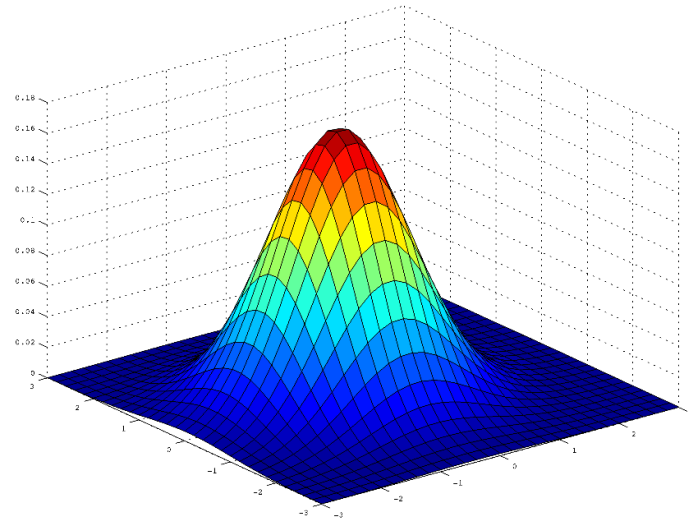


Figure II.9 - Courbe gaussienne en deux dimensions.

Les coefficients de la matrice de convolution de ce filtre sont obtenus à partir de la fonction gaussienne.

Par exemple, si  $\sigma = 0.8$ , on aura le masque suivant :

$$\begin{array}{|c|c|c|} \hline G(-1, -1) & G(0, -1) & G(1, -1) \\ \hline G(-1, 0) & G(0, 0) & G(1, 0) \\ \hline G(-1, 1) & G(0, 1) & G(1, 1) \\ \hline \end{array} \approx \frac{1}{16} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Ce masque sera par la suite appliqué à l'image grâce au produit de convolution.

Remarque : En général, si  $\sigma < 1$ , on obtiendra un masque qui sera utilisé pour réduire le bruit, et si  $\sigma > 1$ , le masque obtenu servira à flouter l'image.

#### f. Filtre bilatéral :

Ce filtre est basé sur la fonction gaussienne vue précédemment, mais, contrairement au filtre gaussien qui s'intéresse seulement à la proximité spatiale, les coefficients de celui-ci dépendent également de la ressemblance des intensités entre le pixel de l'image et celui du masque.

#### g. Filtre de rehaussement de contraste :

Il existe plusieurs méthodes de rehaussement de contraste<sup>2</sup>. Celle que nous avons utilisée dans notre projet consiste à comparer la valeur  $v$  de chaque pixel à celles des pixels avoisinants : Si  $v$  se rapproche du niveau de gris maximal du voisinage, on la remplace par cette intensité maximale, sinon, on la remplace par l'intensité minimale du voisinage.

<sup>2</sup> Le contraste d'une image désigne l'écart entre son intensité maximale et minimale.

### 3. Post-traitement :

#### a. Opérations morphologiques :

Les opérations morphologiques reposent sur la morphologie mathématique, une théorie de traitement non linéaire de l'information très utilisée dans le domaine du traitement d'images. Elle privilégie l'extraction des informations géométriques, c'est-à-dire les formes des composants de l'image.

Les opérations morphologiques dépendent d'un élément structurant, qui est une « forme de référence » sur laquelle se base la comparaison ; On déplace cet élément sur toute l'image à analyser pour mettre en évidence certaines caractéristiques.

Ces opérations ont pour but de régulariser les formes obtenues à partir de la segmentation et de supprimer les pixels superflus.

Il existe deux opérateurs de base : L'érosion et la dilatation, et deux opérateurs obtenus à partir de la combinaison des deux premiers : L'ouverture et la fermeture.

- **Dilatation** : Elle permet d'éliminer les trous isolés qui sont à l'intérieur des objets et dilater les contours en tenant compte de l'élément structurant.
- **Erosion** : C'est l'opération duale de la dilatation (C'est-à-dire que dilater les pixels des objets revient à éroder les pixels du fond). Son but est d'éliminer les pixels isolés du fond et éroder le contour des objets (En fonction de l'élément structurant).
- **Fermeture** : C'est une dilatation suivie d'une érosion. La fermeture adoucit les contours, fusionne les pixels proches les uns des autres et bouche les trous de petite taille.
- **Ouverture** : C'est une érosion suivie d'une dilatation. Elle permet d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant, détacher des objets faiblement liés et lisser les contours.

Remarque : L'application successive de l'ouverture et de la fermeture permet de filtrer le bruit de l'image.

#### b. Conception détaillée des opérations morphologiques :

##### - L'élément structurant :

Dans nos algorithmes, l'élément structurant que nous avons utilisé est celui-ci :

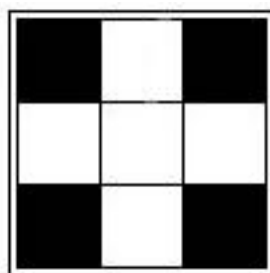


Figure II.10 - Elément structurant.

##### - Entrées et sorties des algorithmes :

En entrée, l'image binaire segmentée (Sous forme de matrice).

En sortie, l'image binaire après avoir effectué les opérations morphologiques (Sous forme de matrice également).

- Pseudo-algorithmes :

**Dilatation :**

- Parcours de l'image (Avec  $i$  l'indice de la ligne, et  $j$  l'indice de la colonne) :
  - Si la valeur du pixel courant  $P(i, j)$  est égale à 1 :
    - Si  $i > 0$  et si  $P(i - 1, j) = 0$  :  
Mettre  $P(i - 1, j)$  à 2.
    - Si  $j > 0$  et si  $P(i, j - 1) = 0$  :  
Mettre  $P(i, j - 1)$  à 2.
    - Si  $(i + 1)$  est inférieur au nombre maximal de lignes de l'image et  $P(i + 1, j) = 0$  :  
Mettre  $P(i + 1, j)$  à 2.
    - Si  $(j + 1)$  est inférieur au nombre de colonnes de la ligne  $i$  et  $P(i, j + 1) = 0$  :  
Mettre  $P(i, j + 1)$  à 2.
- Remettre tous les pixels dont la valeur a changé (Donc tous les pixels qui sont égaux à 2) à 1 (Puisqu'il s'agit d'une image binaire).

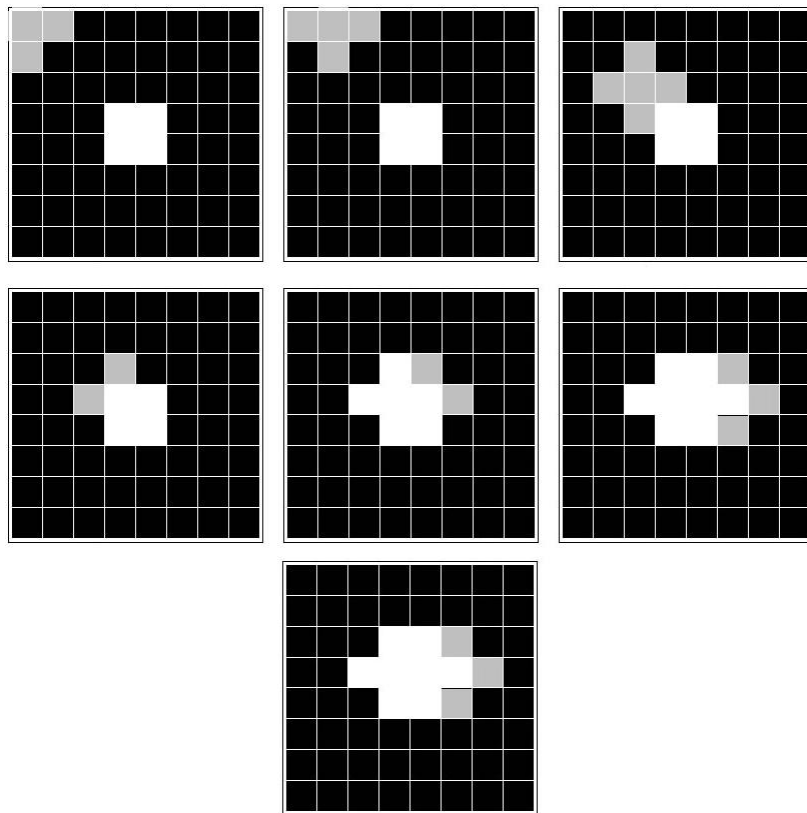


Figure II.11 - Schéma représentant les étapes de la dilatation.

**Erosion :**

Etant donné que l'érosion et la dilatation sont duales, l'algorithme de l'érosion se base sur celui de la dilatation :

- Appliquer l'opérateur logique « Non » à l'image en entrée.

- Appliquer la dilatation.
- Appliquer une deuxième fois le « Non » sur l'image obtenue à partir de la dilatation.

**Fermeture :**

La fermeture étant la succession d'une dilatation et d'une érosion, il suffit d'appliquer les deux opérateurs successivement à l'image en entrée.

**Ouverture :**

L'ouverture étant la succession d'une érosion et d'une dilatation, il suffit d'appliquer les deux opérateurs successivement à l'image en entrée.



## Chapitre 5 : EXTRACTION DES CARACTERISTIQUES

### 1. Introduction :

L'extraction des caractéristiques représente la dernière étape de notre processus de traitement d'image, effectivement l'analyse des régions ne peut se faire qu'après segmentation de l'image, proposant ainsi des composantes connexes, informant l'utilisateur sur les propriétés géométriques des objets obtenus.

Le schéma ci-dessous propose un récapitulatif des étapes qu'a subi l'image avant d'atteindre sa destination finale.

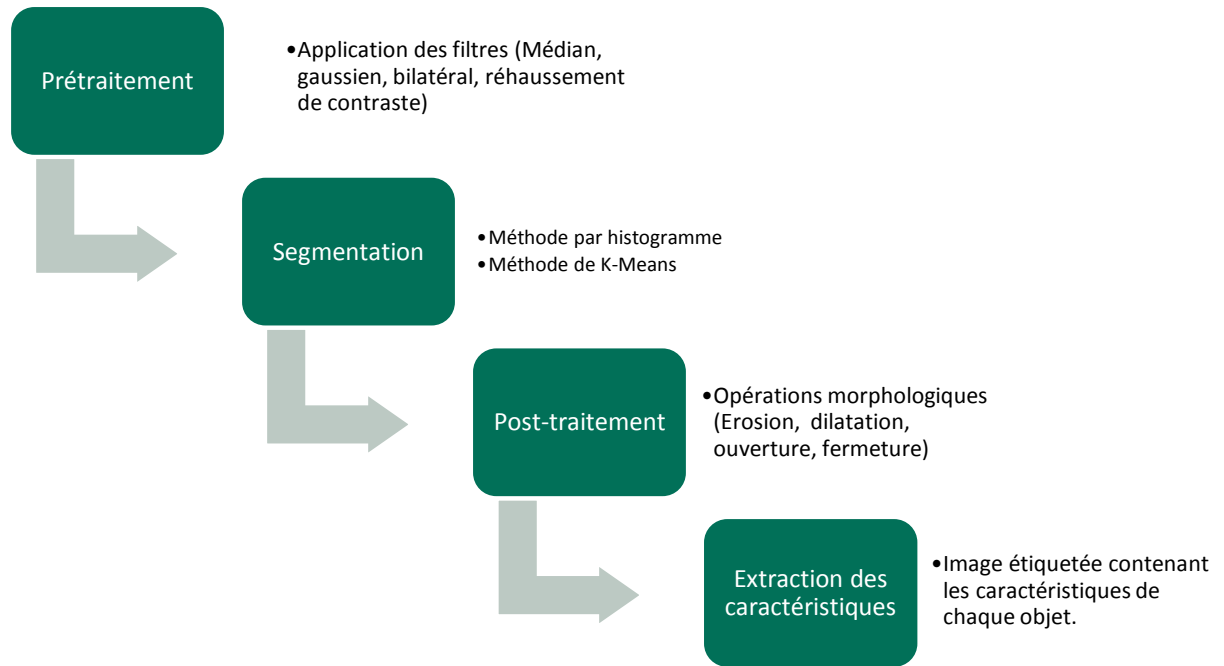


Figure II.12 - Processus de traitement de l'image.

### 2. La classe Regionprops :

La classe Regionprops offre une panoplie de méthodes qui permettent de calculer les propriétés géométriques pour les régions d'une segmentation.

Il est à noter que l'image en entrée doit être sous forme de matrice, ainsi chaque pixel représente une case, les opérations de calcul se font plus aisément. Pour cela la méthode Label est indispensable :



Rôle : Transforme l'image segmentée en une matrice contenant toutes les régions.

Conformément aux besoins de notre projet, nous nous limiterons aux méthodes citées ci-dessous :

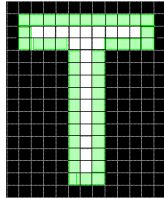

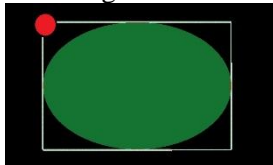
Méthode	Description
Area	<p>Renvoie la somme des pixels d'une région.</p> <p><b>NOTA :</b> Les pixels concernés sont les cellules adjointes ayant la même couleur et formant une région.</p>
Perimeter	<p>Renvoie la somme des pixels concernés par le contour.</p> <p><b>NOTA :</b> La méthode opère de la manière suivante :</p> <ul style="list-style-type: none"> <li>- Erosion de l'image.</li> <li>- Obtention du contour en effectuant la différence entre l'image originale et le résultat de l'érosion.</li> <li>- Calcul de la somme des pixels.</li> </ul>  <p><i>Figure 1.</i> Contour région.</p>
Equivalent_diameter	<p>Renvoie la longueur du diamètre équivalent (On parle toujours de la somme des pixels), qui représente le diamètre du cercle dont la surface est égale à celle de la région concernée.</p>
Centroid	<p>Renvoie les coordonnées du centre de gravité, en se basant sur la méthode Moments.</p> <p><b>NOTA :</b> Un moment est un indicateur de dispersion des pixels.</p>
Major_axis_length et Minor_axis_length	<p>Renvoie la longueur des axes d'inertie, en se basant sur la méthode Moments.</p>
Orientation	<p>Renvoie l'angle entre l'axe majeur d'inertie et l'axe des abscisses, qui représente l'angle de l'axe principal d'inertie.</p>  <p><i>Figure 2.</i> Orientation de l'axe majeur d'inertie.</p>
BoundingBox	<p>Renvoie les coordonnées du point le plus haut à gauche du plus petit rectangle pouvant contenir la région.</p>  <p><i>Figure 3.</i> BoundingBox.</p>

Tableau II.3 - Méthodes de la classe Regionprops.

Segmentation des images par région et extraction des caractéristiques.

### 3. Les entrées et sorties de la classe :

En entrée, l'image binaire.

En sortie, un tableau contenant toutes les régions existantes, accompagnées de leurs propriétés.

## Partie III : Réalisation

*Nous présenterons dans cette partie le langage de programmation choisi et un inventaire des outils et bibliothèques utilisés pour réaliser notre application. Ainsi que les tests et critiques des résultats.*

## Partie III : REALISATION

### Chapitre 1 : LANGAGES D'IMPLEMENTATION ET OUTILS

#### 1. Le langage Python :

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par **Guido van Rossum**, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au *Centrum voor Wiskunde en Informatica* aux Pays-Bas, jusqu'à se voir associer une organisation à but non lucratif particulièrement dévouée, la **Python Software Foundation**, créée en 2001.

Nous avons choisi Python comme langage de programmation pour ces raisons :

- Il est puissant, à la fois facile à apprendre et riche en possibilités.
- Il est très facile d'étendre les fonctionnalités existantes.
- Il existe ce qu'on appelle des **bibliothèques** qui aident le développeur à travailler sur des projets particuliers.



#### 2. L'éditeur PyCharm :

PyCharm est un IDE utilisé pour programmer en Python, il est développé par l'entreprise JetBrains.

Il existe en deux versions : Professional Edition qui est sous licence et Community Edition qui est gratuite.

On a utilisé la version Professional grâce à nos adresses e-mails de l'ESI qui nous ont permis d'obtenir des licences gratuitement.



#### 3. PyQt :

PyQt est parmi les deux meilleures bibliothèques utilisées pour créer des interfaces graphiques en Python, nous l'avons choisi à cause de la facilité d'intégration avec notre programme et son design qui est moderne.



#### 4. Les bibliothèques libres :

**Scipy** : La librairie Scipy intègre un ensemble de modules pour l'optimisation, l'algèbre linéaire, les statistiques, le traitement du signal et le traitement d'images.

**Numpy** : C'est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle Open Source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

**Matplotlib** : La librairie Matplotlib permet l'affichage de courbes, de graphiques, d'histogrammes et l'export des courbes dans divers formats (PDF, SVG, PNG ou BMP). Matplotlib possède également un module nommé pylab utilisant une syntaxe proche de Matlab pour l'affichage.

**OpenCV** : (**Open** Source **C**omputer **V**ision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (Libre) pour les plateformes Windows, GNU/Linux, Android et MacOS.

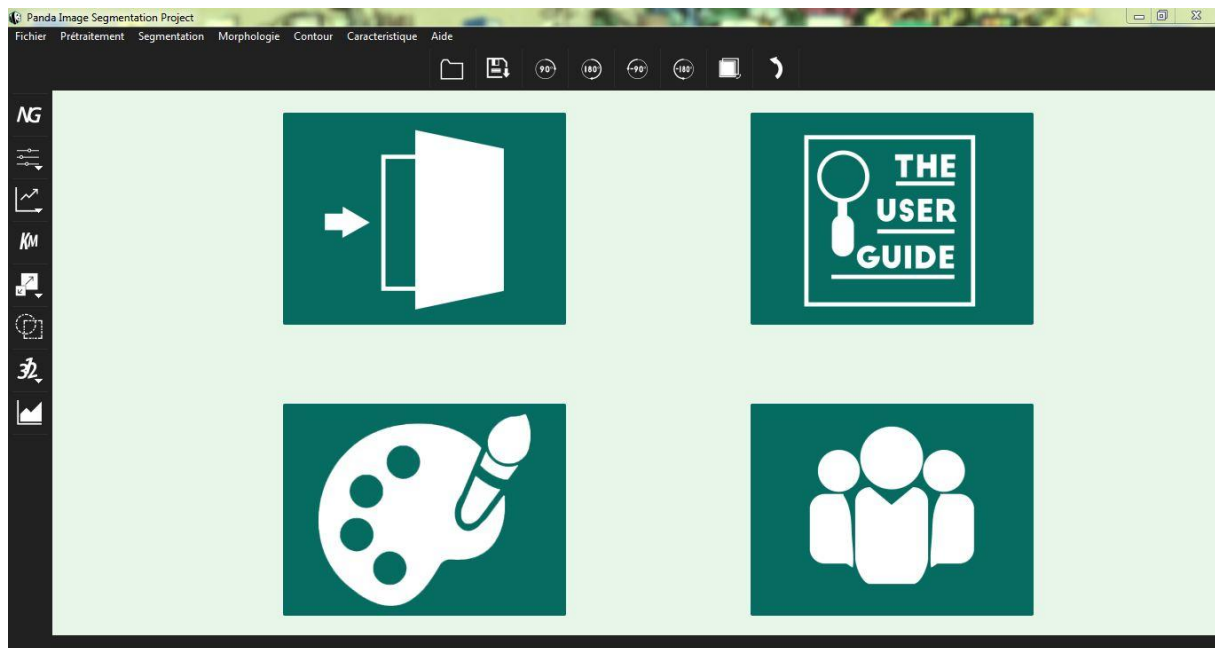
Initialement écrite en C il y a 10 ans par des chercheurs de la société Intel, OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40 000 membres actifs. C'est la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie.



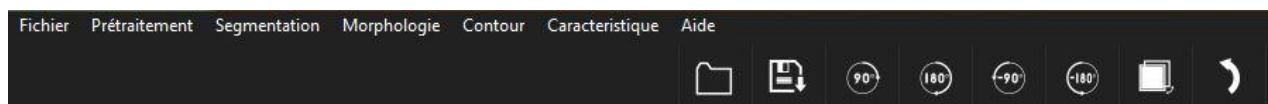
**Anaconda** : Développé par l'entreprise Continuum. Il a l'avantage d'être multiplateforme (Windows, Mac OS X et GNU/Linux) et d'intégrer une grande quantité d'outils et packages Python, notamment : IPython, Spyder, NumPy, SciPy, Matplotlib, Sympy, PIP...

## Chapitre 2 : INTERFACE UTILISATEUR

- Ecran d'accueil : L'espace de travail de l'application « Panda » vu par l'utilisateur.



- Fonctionnalités et barre d'outils horizontale :



L'application propose les fonctionnalités suivantes :

- **Fichier** : Pour ouvrir, sauvegarder une image, quitter l'application.
- **Prétraitement** : Cet onglet comprend la fonction niveau de gris, l'affichage de l'histogramme, les filtres, ainsi que les opérations de rotation et de zoom.
- **Segmentation** : L'utilisateur a le choix de faire une segmentation par seuillage (Li, Otsu, Iso Data, Yen), ou une segmentation de K-means.
- **Morphologie** : Cet onglet propose les opérations morphologiques qu'on a développé.
- **Contour** : Fonction de détection de contour.
- **Caractéristiques** : Affiche les caractéristiques de l'image sous différentes formes (Image en niveau de gris, en couleur et étiquetage)

La barre d'outils horizontale propose :

- D'ouvrir et sauvegarder une image.
- Des rotations avec différents angles.
- La binarisation de l'image.
- Barre d'outils verticale : Une autre manière d'accéder aux fonctionnalités de l'application, présentées dans cet ordre : Niveau de gris/ Filtres/ Seuillage/ K-means/ Opérations morphologique/ Détection de contour/ Extraction des caractéristiques/ Histogramme.



### Chapitre 3 : TESTS ET RESULTATS

---

Dans ce chapitre, nous proposons d'évaluer notre application en présentant nos jeux d'essai ainsi que les résultats obtenus après application des différentes méthodes implémentées.

#### 1. Acquisition d'image :



Figure III.1 - Image originale.

#### 2. Prétraitement :

En ce qui concerne les prétraitements, nous pouvons voir l'image en niveau de gris, afficher l'histogramme, supprimer les bruits en utilisant différents filtres, en plus de la modification d'orientation et du zoom.

Voici quelques exemples :



Figure III.2 - Résultat du filtre niveau de gris.



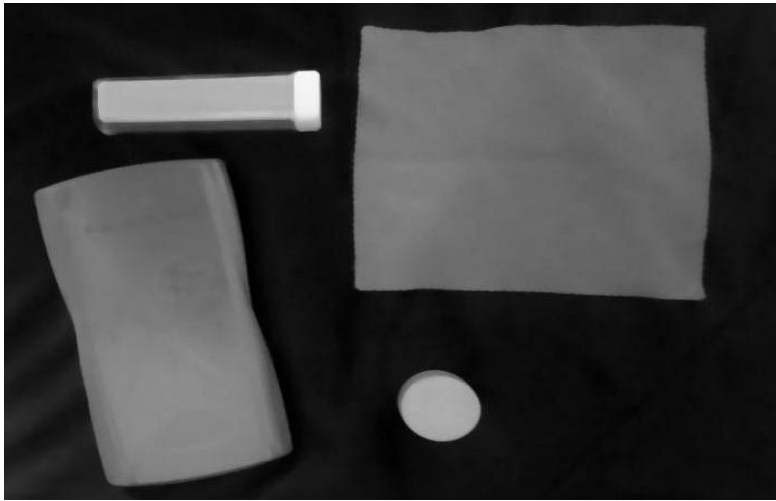


Figure III.3 - Test du filtre médian.

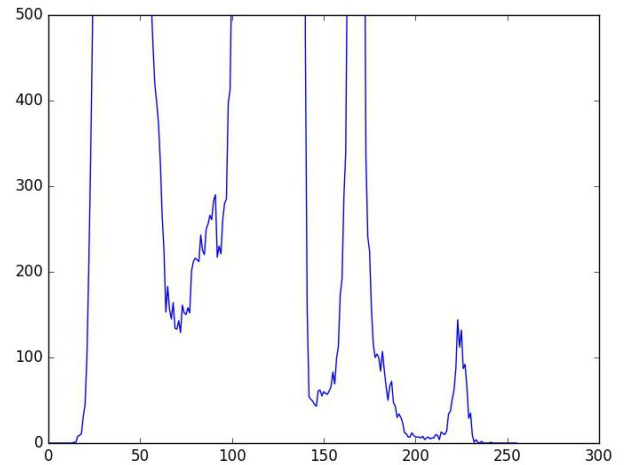


Figure III.4 - L'histogramme.

- Après application du filtre médian, les différents objets de l'image sont nettement plus homogènes.

### 3. Segmentation des images :

On présente deux méthodes de segmentation :

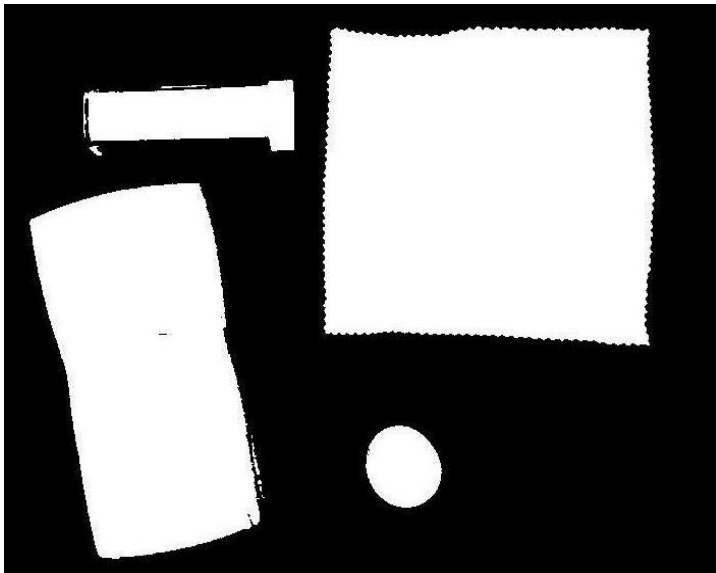


Figure III.5 - Segmentation par l'algorithme de K-Means.

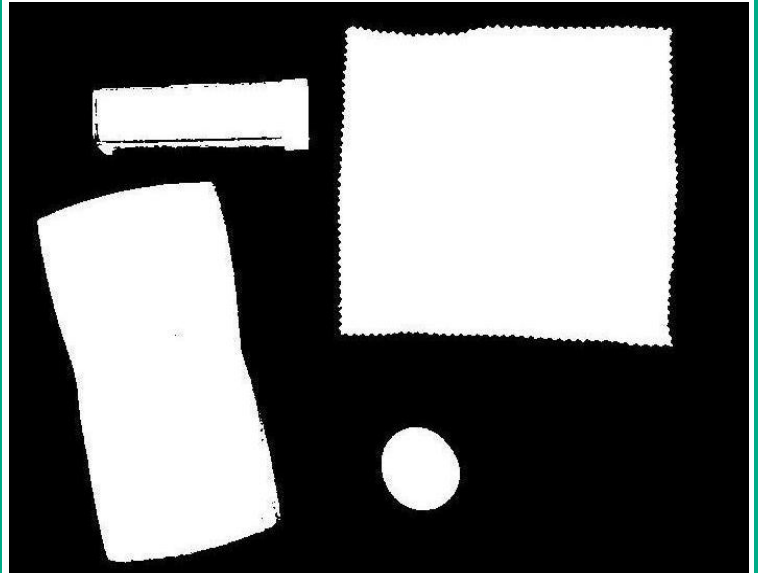


Figure III.6 - Segmentation par l'algorithme d'Otsu.

- Critiques :  
Nous pouvons remarquer que la segmentation par la méthode K-means donne de meilleurs résultats en termes de précision, mais elle présente néanmoins l'inconvénient du temps de

traitement qui est plus ou moins considérable (quelques secondes), tandis qu'une segmentation par seuillage est quasi instantanée.

En conclusion, il s'avère que chaque méthode présente des avantages et des inconvénients.

#### 4. Operations morphologiques :

Pour les opérations morphologiques, nous avons la dilatation, l'érosion, l'ouverture, la fermeture, et l'ouverture et la fermeture (l'une après l'autre). Voici un exemple :

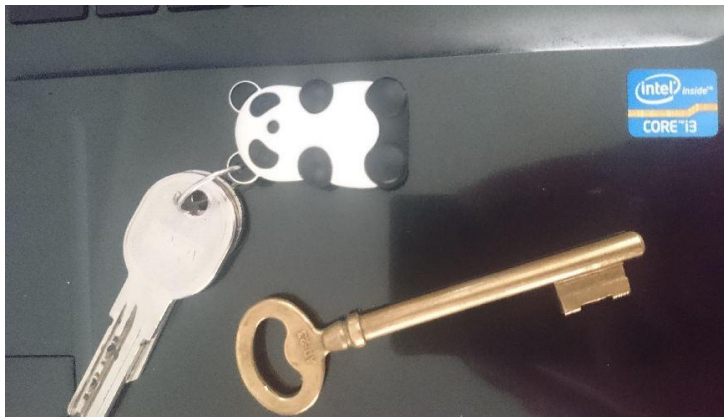


Figure III.7 - Image originale.

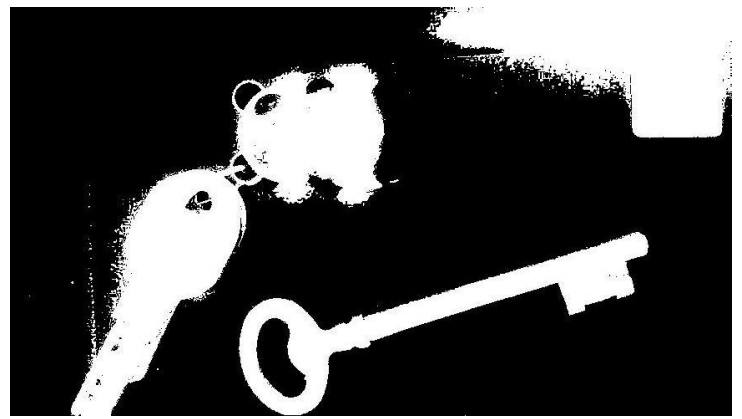


Figure III.8 - Image après la binarisation.

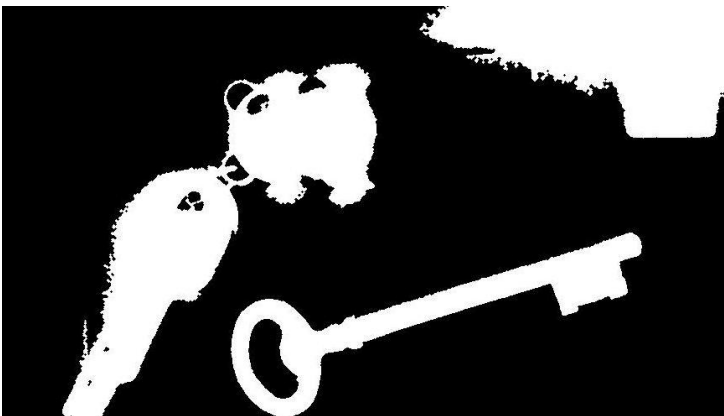


Figure III.9 - Image binaire après l'application des opérations morphologiques (Ouverture et fermeture).

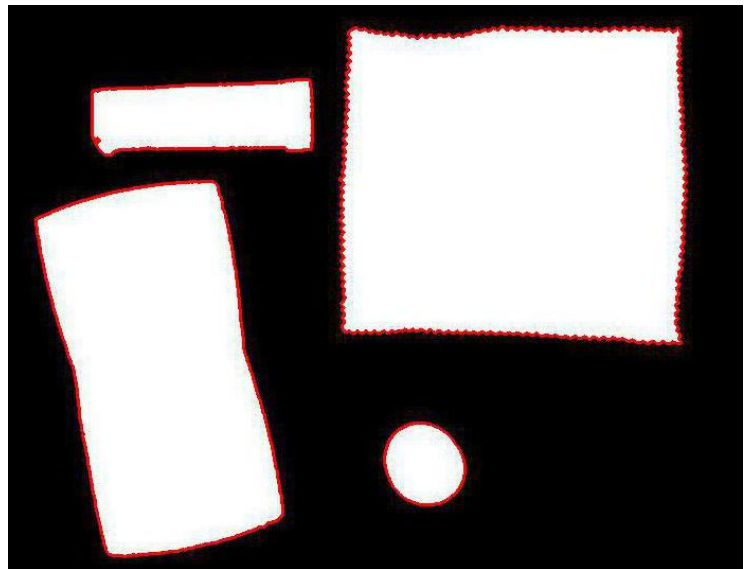
**Remarque :**

On peut remarquer qu'après l'application des opérations morphologiques, le bruit a été diminué.

#### 5. Suivi des contours :

Pour la détection des contours, on va voir un test en utilisant l'algorithme du Marching Squares :

Figure III.10 - Détection des contours par la méthode du Marching Squares.



## 6. Analyse des régions :

Nous avons mesuré les différents paramètres des régions contenues dans l'image tels que : la surface, le périmètre, le diamètre équivalent, les coordonnées du centre de gravité, ...etc.



Figure III.11 - Affichage des différents paramètres.



Figure III.12 - Affichage des différents paramètres.

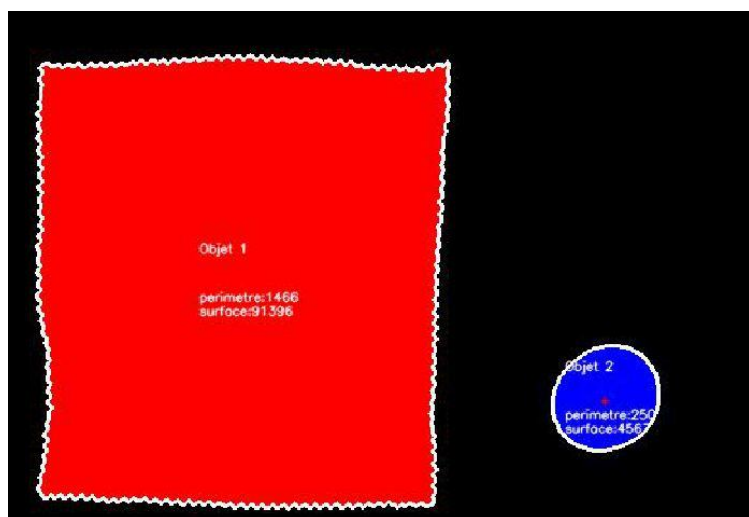


Figure III.13 - Image d'étiquettes des régions 1 & 2 avec le dessin de contour.

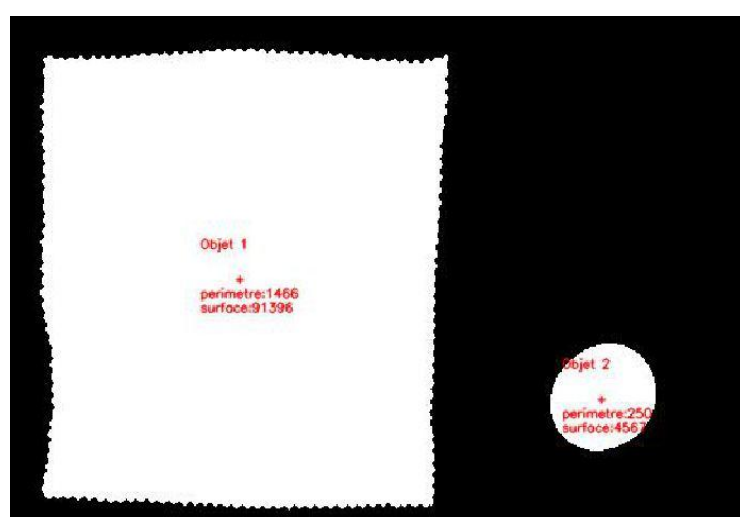


Figure III.14 - Image binaire avec les caractéristiques.

## GESTION DU PROJET

Afin d'assurer le bon déroulement de tout projet informatique, il est primordial d'établir une gestion performante qui associe organisation, travail et esprit d'équipe.

En ce qui concerne l'organisation, nous avons bien sûr suivi le planning élaboré, et modifié au fur et à mesure que l'on avançait dans la réalisation, étant donné qu'au début il n'était pas évident de planifier la durée des tâches, ni de déterminer les tâches qui apparaissaient après l'accomplissement de quelques une. Le diagramme de Gantt nous a été très utile pour assurer le suivi de toutes les tâches, ainsi qu'une bonne coordination entre elles.

Mais avant qu'on ait commencé la répartition des tâches, nous nous étions accordés le temps nécessaire pour une large documentation qui a permis à chaque membre de bien cerner le sujet. Le choix du langage de programmation a évidemment fait l'unanimité, un choix pertinent et très réfléchi.

La répartition des tâches s'est faite spontanément à partir du cahier de charge, chaque membre avait une fonction qu'il devait mener à bien, et en assumer la responsabilité, bien évidemment il pouvait solliciter l'aide des autres membres en cas de besoin. Parallèlement à l'implémentation de tous les algorithmes, nous fûmes divisés en deux : l'équipe chargée de concevoir l'interface graphique, et l'équipe qui s'occupait du rapport au fur et à mesure que l'application prenait vie.

L'un des aspects les plus importants de cette gestion est la communication. Une réunion en début de semaine était tenue entre les membres, c'était l'occasion pour chacun d'avoir un aperçu sur l'état d'avancement du projet. L'autre moyen de communication était la boîte email de l'école, en effet, c'est ce qui a assuré l'échange fluide d'informations utiles.

Le point qui a fait notre force est sans aucun doute le travail. En d'autres termes, nous avons suivi un rythme intensif dès le début, qui a été récompensé par l'aisance avec laquelle nous avons pu terminer le projet. Par ailleurs, nous avons tenu à appliquer le concept de base de la programmation orientée objet, fraîchement acquis en cours, en concevant des classes pour encapsuler toutes les méthodes utilisées lors de la conception.

Pour finir, les membres de l'équipe ont tous fournis un effort considérable pour aboutir aux objectifs fixés, sous la coordination du chef d'équipe. Une entente générale régnait au sein du groupe, et un véritable esprit d'équipe naquit.

## CONCLUSION GENERALE

La segmentation et l'analyse des images est un domaine en plein essor qui est promis à un bel avenir. En effet, les recherches scientifiques se poursuivent encore jusqu'à présent, et chaque jour, de nouvelles contributions sont apportées.

A travers ce projet, nous avons pu témoigner de la diversité des méthodes de segmentation par région, mais aussi de la difficulté d'automatiser ces procédés. De plus, nous avons constaté que certains algorithmes sont très gourmands en ressources, notamment l'algorithme K-Means.

D'autre part, outre le fait que ce projet nous ait permis de mettre en pratique les connaissances acquises durant ces deux dernières années, il nous a aidé à explorer et à développer d'autres aspects, entre autres, l'aspect relationnel et organisationnel.

En effet, sur le plan organisationnel, ce projet nous a permis de nous familiariser avec l'esprit d'entreprise, et d'apprendre à respecter un planning de travail et à mieux s'organiser.

D'un point de vue humain, cette expérience nous a appris à travailler en équipe, à consolider nos liens et à respecter les idées d'autrui. Bien entendu, nos avis n'étaient pas toujours partagés mais nous avons su trouver des compromis pour satisfaire tout le monde.

En conclusion, cette expérience s'est révélée très enrichissante dans la mesure où nous avons pu avoir un aperçu concret de la façon dont est mené un projet informatique. Nous comptons bien évidemment améliorer notre application davantage et ajouter d'autres fonctionnalités à l'avenir.

## BIBLIOGRAPHIE ET WEBOGRAPHIE

- Bergounioux, M. (2008). *Quelques méthodes mathématiques pour le traitement d'image*.  
Récupéré sur <https://cel.archives-ouvertes.fr/cel-00125868v4>
- Bernard, O. (s.d.). *Traitement d'images numériques. Analyse d'images : 2ème partie*.  
Récupéré sur [https://www.creatis.insa-lyon.fr/~bernard/files/courses/DIP-5ImageAnalysis\\_part2.pdf](https://www.creatis.insa-lyon.fr/~bernard/files/courses/DIP-5ImageAnalysis_part2.pdf)
- Echegut, R. (2010). *Méthodes de segmentation*.  
Récupéré sur <http://www.univ-orleans.fr/mapmo/membres/louchet/teaching/timo/Echegut.pdf>
- Gouiffès, M. (s.d.). *Segmentation d'images*.  
Récupéré sur <http://m.i.c.h.e.l.e.free.fr/CoursSegmentation.pdf>
- Goumeidane, A. B. (2010). *Segmentation et évaluation des images segmentées*.
- Lingrand, D. (s.d.). *Marching Squares*.  
Récupéré sur <http://users.polytech.unice.fr/~lingrand/MarchingCubes/algo.html>
- Oberlin, T. (s.d.). *Cours de traitement des images. Partie 6 : Segmentation d'images*.  
Récupéré sur <http://oberlin.perso.enseiht.fr/teaching.html>
- Rafael C. Gonzalez, R. E. (s.d.). *Digital Image Processing*. Pearson Education.
- Ravishankar Chityala, S. P. (2014). *Image Processing and Acquisition using Python*. Chapman and Hall/CRC.

## TABLE DES FIGURES

Figure I.1 - La représentation d'une image numérique par un tableau 2D avec la précision d'un pixel de coordonnées (n,m).....	6
Figure I.2 - Schéma représentant les différentes méthodes de segmentation. ....	8
Figure I.3 - Localisation de tumeurs (A gauche) et reconnaissance d'empreintes digitales (A droite). ..	9
Figure I.4 - Application d'opérateurs morphologiques sur une image binaire. ....	11
Figure I.5 - Axe majeur d'inertie et orientation.....	12
Figure I.6 - Axe mineur d'inertie. ....	12
Figure I.7 - Centre de gravité d'une région. ....	13
Figure II.1 - Les méthodes de seuillage utilisées. ....	16
Figure II.2 - Histogramme d'une image. ....	16
Figure II.3 - Seuil calculé à partir de l'histogramme. ....	17
Figure II.4 - Représentation d'une cellule. ....	24
Figure II.5 - Les 16 configurations possibles.....	25
Figure II.6 - Produit de convolution (Exemple).....	27
Figure II.7 - Filtre médian (Exemple).....	27
Figure II.8 - Courbe gaussienne en deux dimensions. ....	28
Figure II.9 - Élément structurant. ....	29
Figure II.10 - Schéma représentant les étapes de la dilatation. ....	30
Figure II.11 - Processus de traitement de l'image. ....	32
Figure III.1 - Image originale. ....	39
Figure III.2 - Résultat du filtre niveau de gris. ....	39
Figure III.3 - Test du filtre médian. ....	40
Figure III.4 - L'histogramme.....	40
Figure III.5 - Segmentation par l'algorithme de K-Means.....	40
Figure III.6 - Segmentation par l'algorithme d'Otsu. ....	40
Figure III.7 - Image originale. ....	41
Figure III.8 - Image après la binarisation. ....	41
Figure III.9 - Image binaire après l'application des opérations morphologiques (Ouverture et fermeture).....	41
Figure III.10 - Détection des contours par la méthode du Marching Squares. ....	42
Figure III.11 - Affichage des différents paramètres.....	42
Figure III.12 - Affichage des différents paramètres.....	42
Figure III.13 - Image d'étiquettes des régions 1 & 2 avec le dessin de contour.....	43
Figure III.14 - Image binaire avec les caractéristiques. ....	43