



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Système d'Information et de Connaissances (S.I.C)

Thème

Développement d'un Système de Dénombrement de Cellules Sanguines

Réalisé par :

- Mme Bouklab Radja

Présenté le 10 janvier 2013 devant le jury composé de :

- Mr BENMAMMAR Badr (Président)
- Mr CHOUITI Sidi Mohammed (Encadreur)
- Mr BENAMAR Abdelkrim (Examineur)
- Mr BENZIAN Yaghmorasan (Examineur)

Année universitaire: 2011-2012

Remerciement

Au-delà de la formalité d'usage, c'est avec un grand plaisir que je remercie:

Allah le tout puissant de m'avoir donné la force d'achever ce travail ;

Je remercie en particulier ma cher mère qui ma accorder toutes leur attention et patience que j'ai inspiré, tout au long de mes études.

*Mon encadreur, **Mr chouiti Sidi Mohamed** pour ses conseils, ses orientations et la qualité des commentaires et suggestions dont il m'a fait part ;*

Chacun des membres du jury pour m'avoir fait l'insigne honneur d'accepter de juger mon travail ;

Je ne saurai oublier de remercier toutes les personnes qui m'ont aidé par leur soutien permanent de prés ou de loin.

DÉDICACES

Je dédie ce modeste travail et ma profonde gratitude à ma cher mère qui a toujours été la pour moi, a l'éducation qu'il ma prodigué avec tous les moyens, et au prix de tous les sacrifices qu'elle ma consentis a mon égard,

En guise de reconnaissance, je dédie cet humble travail à ceux qui me sont chers ;

*A ma très cher sœur **Ikram** et mon petit frère **Marwan***

*A Mon très cher époux **Mohamed** qui ne cesse de me soutenir et m'encourager. Je ne le remercierai jamais assez pour tout ce qu'il a fait pour moi*

*A mon ange **Chiraz** que dieu me la protège*

*A tout ma famille surtout ma tante **Noria***

A tous les enseignants qui ont contribué à ma formation

A tous ceux qui occupent une place dans mon cœur.

Résumé

Ce projet s'insère dans le cadre du projet National de recherche SARCY (Système Adaptable pour la Reconnaissance des cellules CYtologiques). Mon objectif est de développer un outil pour le dénombrement de cellules cytologiques. Pour cela, j'ai utilisé l'approche Mean shift pour segmenter les images et en déduire la détection des contours des cellules cytologiques.

Pour la phase d'apprentissage et de classification de ces cellules sanguines, essentiellement des globules blancs et des globules rouges, j'ai implémenté un algorithme basé sur la méthode de boosting qui repose sur le principe des itérations. Le langage de programmation utilisé est java. Pour des raisons de probabilités sur différent machines liée au domaine médicale.

Mot clés : cellules cytologiques, boosting, Mean shift.

Abstract

This project is a part of the SARCY's project; my goal is to develop a tool to focuses on the recognition of blood cells. For this, I use the Mean shift approach to segment images and deduce the edge detection.

For the learning phase and classification of blood cells, mainly white cells and red blood cells, I design and implement algorithms based on boosting method based on the principal of iteration. I have programmed in java. At rate of probability about different machine tie to medical domain.

Key words: blood cells, boosting, Mean shift.

ملخص

ان مشروعى هذا هو جزء من مشروع SARCY الهدف الرئيسى منه هو تطوير نموذج لمعرفة خلايا الدم و ذلك باستعمال تقنية Mean Shift لقطع و تحديد اجزاء من الصور.

اما بالنسبة لمرحلة تصنيف خلايا الدم (خلية دم حمراء و خلية دم بيضاء) فاستعملت تقنية boosting التى تعتمد على منهج التكرار و للحصول على احصاء جيد للخلايا اعتمدت على تقنيات JAVA لتطوير النظام

الكلمات الرئيسية : خلايا الدم , Mean shift , boosting.

Sommaire

Introduction générale :	8
Chapitre 1 : Introduction aux images numérique	10
I. Introduction	10
II. Image numérique	10
II.1 Définition d'une image	10
II.2 Définition d'image numérique	10
II.3. Caractéristiques d'une image numérique	11
II .3.1 Pixel	11
II .3.2 La taille d'une image	12
II .3.3 Bruit	12
II .3.4 Histogramme	12
II .3.5 Contours et textures	13
II .3.6 Luminance et contraste	13
II .3.7 Niveau de gris	14
II .3.8 Les couleurs	14
II.4 Types d'image numérique	15
Il existe Deux sortes d'images numériques :	15
II.4.1 Image matricielles « ou images bitmap »	15
II.4.2 Images vectorielles	15
II.5 La qualité de l'image numérique	16
III. La composition des cellules sanguines	16
III.1. Le plasma	17
III. 2. Erythrocytes (globules rouges)	17
III.3. Les plaquettes	18
III.4. Leucocyte (globules blancs)	18
III.4.1 Les neutrophiles	19
III.4.2 Les éosinophiles	20
III.4.3 Les basophiles	21
III.4.4 Les lymphocytes	21
III.4.5 Les monocytes	22
IV. Conclusion	23
Chapitre 2 : acquisition et prétraitement d'image numérique	24

I.	Introduction.....	24
II.	Acquisition des images couleurs.....	24
III.	Prétraitement de l'image.....	25
	III.1.1 Seuillage.....	26
	III.1.2 Egalisation d'histogramme.....	27
	III.1.3 Rehaussement de contraste.....	28
	III.1.4 Minimum et maximum de niveaux de gris.....	28
	III.1.5 Squelettisation.....	28
	III.2 Restauration d'image par filtrage.....	29
	III.2.1 Définition Filtrage.....	29
	III.2.2 Convolution.....	29
	III.2.3 Type de filtrage.....	30
	III.3 Détection de contour.....	33
	III.3.1 Opérateur de laplacien.....	36
	III.3.2 Opérateur de SOBEL.....	38
	III.3.4 Opérateur de ROBERT.....	39
	III.3.5 Opérateur de PREWITT.....	40
	III.4 Morphologie mathématique.....	41
	III.4.1 Erosion.....	41
	III.4.2 Dilatation.....	41
IV.	Conclusion.....	42
	Chapitre 3 : Segmentation et Méthode de classification.....	43
I.	Introduction.....	43
II.	Segmentation.....	43
	II.1. segmentation par région :.....	44
	II.2. segmentation par classification (seuillage).....	44
	III.3. Segmentation par détection de contours.....	44
	II.3.1. Algorithme Mean Shift.....	45
	II.3.2. Principe de l'algorithme.....	45
	II.3.3. La construction de Mean shift.....	52
III.	Méthode classification.....	62
	III.1. Définition.....	62
	III.2. Apprentissage.....	62

III.3. Apprentissage automatique	62
III.3.1 Définition	62
III.4. Types d'apprentissage	63
III.4.1. L'apprentissage supervisé	63
III.4.2. Apprentissage non-supervisé.....	64
III.4.3. Apprentissage semi-supervisé	64
III.4.4. L'apprentissage partiellement supervisé	65
III.4.5. Apprentissage par renforcement.....	65
III.5. Méthodologie utilisée.....	66
III.5.1. Introduction	66
III.5.2. Le classifieur linéaire	66
III.5.3. Les arbres de décision	67
III.6. Le Boosting	69
III.6.1. Définition	69
III.6.2 Principe général.....	70
IV. Conclusion	74
I. Introduction.....	75
II. Le langage support	75
III. Processus de développement.....	75
III.1 Détection de contour :	77
III.2 La phase de reconnaissance d'objet :	77
III.3 Création des annotations :	78
III.4 Reconnaissance des globules	81
III.5 Classification.....	82
V. Conclusion :	95
Conclusion et perspectives :	96

Introduction générale :

Les progrès des moyens informatiques et des techniques de traitement ont permis d'ouvrir une voie de développement très prometteuse. L'amélioration et la multiplication des moyens de diffusion des données informatiques a pris également une grande part à la multiplication de ces techniques. Les outils de communication utilisent de plus en plus les images, ce qui a entraîné un accroissement des recherches dans les domaines du traitement et de l'analyse d'images. Ces techniques ont été développées avec succès dans des domaines divers et très variés (images industrielles, de robotique, biologiques, aériennes, etc.).

Dans le domaine médical, l'utilisation des images est à présent considérée comme indispensable pour des raisons de facilité de stockage et aussi pour disposer de techniques d'analyse des images automatiques ou semi-automatiques grâce à des systèmes informatiques. Ces systèmes permettent de proposer aux praticiens une nouvelle façon de travailler en obtenant de nouvelles mesures par des techniques quantitatives et objectives.

Suivant les techniques utilisées, les examens d'imagerie médicale permettent d'obtenir des informations sur l'anatomie des organes (leur taille, leur volume, leur localisation, etc.)

Mon projet s'insère dans le cadre du projet SARCY (Système Adaptable pour la Reconnaissance des cellules CYtologiques).

Mon objectif est de développer un système de dénombrement des cellules sanguines c'est-à-dire : réaliser un outil de comptage regroupant l'identification, la quantification de globules rouge et blanc afin de pouvoir automatiser la partie de dénombrement.

Mon mémoire est organisé comme suit:

Le chapitre 1 : une introduction aux images numériques.

Le chapitre 2 : j'ai présenté comment acquérir une telle image est faire un prétraitement (filtrage,...) avant l'analyse par la segmentation.

Le chapitre 3 : présente le traitement d'image par segmentation en utilisant la méthode Mean shift. La classification et la quantification est mise en oeuvre par l'algorithme boosting, son rôle sera principalement classifier les globules selon leur nature rouge ou blanc. Très puissante, cette méthode arrive à discriminer des groupes là où d'autres méthodes, comme les méthodes linéaires, échouent.

Le chapitre 4 : présente le prototype ainsi que les différentes approches implémentées.

Conclusion et perspectives : cette partie présente les différentes analyses qui ont été menées, les résultats obtenus, ainsi que les limites de mon travail, et les perspectives que nous envisageons.

Chapitre 1 : Introduction aux images numérique

I. Introduction

Le système de vision artificielle capte une scène tridimensionnelle et la traduit en une ou plusieurs images, parmi ces applications imagerie médicale.

II. Image numérique

Avant définir qu'est ce que une image numérique on doit tout d'abord définir ce qui une image.

II.1 Définition d'une image

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain. Elle peut être décrite sous la forme d'une fonction $I(x,y)$ de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation. [1]

II.2 Définition d'image numérique

Image Numérique désigne toute image (dessin, icône, photographie...) acquise, créée, traitée et stockée sous forme binaire. Elle est décrite comme une fonction $I(x, y)$ de brillance analogique continu définie dans un domaine borné, telle que x et y sont les coordonner partielle d'un point de l'image et I est une fonction d'intensité lumineuse (niveau de gris) et de couleur :

$$I : \mathbb{R}^2 \longrightarrow \mathbb{R}$$

$$(x, y) \longrightarrow I(x, y) \quad (1.1)$$

Avec \mathbb{R} : ensemble des réelles

Elle est acquise par des convertisseurs analogique-numérique situés dans des dispositifs comme les scanners, les appareils photo ou les caméscopes numériques...

Et crée directement par des programmes informatiques, grâce à une souris, des tablettes graphiques ou par de la modélisation 3D (ce que l'on appelle, par abus de langage, les « images de synthèse ») ;

Et traitée grâce à des outils informatiques, de façon à la transformer, à en modifier la taille, les couleurs, d'y ajouter ou d'en supprimer des éléments, d'y appliquer des filtres variés, etc. ;

Et stockée sur un support informatique (disquette, disque dur, CD-ROM...). [15]

II.3. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

II .3.1 Pixel

Contraction de l'expression anglaise " Picture Elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :



La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B). [16]

II .3.2 La taille d'une image

La taille de l'image se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image. Et pour calculé la taille d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre des cases de la matrice, soit la hauteur de celui-ci que multiplie sa largeur. La taille de l'image est alors le nombre des pixels que multiplie la taille (en octet) de chacun de ces éléments.

Exemple : pour une image de 240 X 420

Nombre de pixels :

$$240 \times 420 = 100800$$

Taille de chaque pixel : 24 bits / 8 = 3 octets

Le poids de l'image est ainsi égal à :

$$100800 \times 3 = 302.400 \text{ égal } 302.400/1024 = 295 \text{ Ko}$$

II .3.3 Bruit

Le bruit d'image est la présence d'informations parasites qui s'ajoutent de façon aléatoire aux détails de la scène photographiée numériquement. Il est plus particulièrement visible dans les zones peu éclairées, où le rapport signal/bruit est faible, mais aussi dans les parties uniformes telles qu'un ciel bleu. Il a pour conséquence la perte de netteté dans les détails.

II .3.4 Histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RGB :

1. Un histogramme représentant la distribution de la luminance.
2. Trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouges, bleues et vertes. [2]

II .3.5 Contours et textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes. [16]

II .3.6 Luminance et contraste

La luminance est l'intensité d'une source étendue dans une direction donnée, divisée par l'aire apparente de cette source dans cette même direction.

Le contraste est une propriété intrinsèque d'une image qui désigne et quantifie la différence entre les parties claires et foncées d'une image (elle différencie les couleurs claires des couleurs foncées).

La loi de Weber-Fechner définit le contraste C_w comme :

$$C_w = \frac{L - L_{fond}}{L_{fond}}$$

L et L_{fond} désignent respectivement la luminance de l'objet et du fond directement aux abords de l'objet. Ce contraste mesure la perception que l'on a d'un objet de luminance L placé sur un fond de luminance L_{fond} .

Le contraste de Michelson C_m (compris entre 0 et 1) est défini comme :

$$C_m = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

Où L désigne la luminance et I l'intensité lumineuse. L_{min} , L_{max} sont respectivement les degrés de luminosité de deux zones voisines

II .3.7 Niveau de gris

Le niveau de gris, désigne la concentration des points. Un niveau de gris va alors varier du blanc au noir. Cela est également valable pour toutes les autres couleurs (Rouge, Vert, Bleu).

Dans le cas d'une image numérique, le niveau de gris représente l'intensité lumineuse d'un pixel, lorsque ses composantes de couleur sont identiques en intensité lumineuse. [15]

II .3.8 Les couleurs

Il existe plusieurs modes de codage informatique des couleurs, le plus utilisé pour le maniement des images est l'espace colorimétrique rouge, vert, bleu (RVB ou RGB - Red Green Blue). Cet espace est basé sur une synthèse additive des couleurs, c'est-à-dire que le mélange des trois composantes R, V, et B à leur valeur maximum donne du blanc, à l'instar de la lumière. Le mélange de ces trois couleurs à des proportions diverses permet de reproduire à l'écran une part importante du spectre visible, sans avoir à spécifier une multitude de fréquences

- Images 24 bits (ou « couleurs vraies ») : Le codage de la couleur est réalisé sur trois octets, chaque octet représentant la valeur d'une composante couleur par un entier de 0 à 255. Ces trois valeurs codent généralement la couleur dans l'espace RVB. Le nombre de couleurs différentes pouvant être ainsi représenté est de $256 \times 256 \times 256$ possibilités, soit près de 16 millions de couleurs
- Images à palettes, images en 256 couleurs (8 bits) : c'est une méthode qui donne la possibilité de définir une palette, ou table de couleurs, contenant l'ensemble des couleurs peuvent être contenues dans l'image, à chacune des quelles est associé un indice. Le nombre de bits réservé au codage de chaque indice de la palette détermine le nombre de couleurs peuvent être utilisées.
- Images en teintes (ou niveaux) de gris : en code généralement sur un octet (256 valeurs). Par convention, la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale)
- Image noir et blanc : en stockant un bit dans chaque case, il est possible de définir deux couleurs (noir ou blanc). [18]

II.4 Types d'image numérique

Il existe Deux sortes d'images numériques :

II.4.1 Image matricielles « ou images bitmap »

Elle est composée comme son nom l'indique d'une matrice (tableau) de points à plusieurs dimensions. Dans le cas des images à deux dimensions (le plus courant), les points sont appelés pixels. D'un point de vue mathématique, on considère l'image comme une fonction de dans où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme un codage.

Exemple : JPEG , GIF, TIFF, PNG

II.4.2 Images vectorielles

Une image vectorielle (ou image en mode trait), en informatique, est une image numérique composée d'objets géométriques individuels (segments de droite, polygones, arcs de cercle, etc.) définis chacun par divers attributs de forme, de position, de couleur, etc. Elle se différencie de cette manière des images matricielles, dans lesquelles on travaille sur des pixels.

Le principe est de représenter les données de l'image par des formules géométriques qui vont pouvoir être décrites d'un point de vue mathématique. Cela signifie qu'au lieu de mémoriser une mosaïque de points élémentaires, on stocke la succession d'opérations conduisant au tracé.

L'avantage de ce type d'image est la possibilité de l'agrandir indéfiniment sans perdre la qualité initiale. [16][3]

Exemple : SVG

II.5 La qualité de l'image numérique

La qualité d'une image numérique s'exprime en termes de rendu chromatique, de résolution et de netteté, bref de sa fidélité par rapport au sujet. Elle dépend de plusieurs facteurs :

- La résolution optique du système photographique. Elle varie en fonction de la résolution du capteur photosensible, de la qualité de l'optique, en particulier des aberrations résiduelles.
- La fidélité chromatique qui est influencée par l'espace de couleur de travail, la dynamique de l'appareil.
- Le vignetage qui dépend des caractéristiques de l'optique, de la taille du capteur photosensible et éventuellement des accessoires (pare-soleil, etc).
- La qualité de l'affichage à l'écran : définition de l'écran, calibrage de l'écran
- Le mode de représentation (bitmap, vectoriel). **[18]**

III. La composition des cellules sanguines

Le sang est constitué de cellules spéciales en suspension dans un liquide appelé plasma. Le sang représente environ 1/12 de la masse corporelle d'un homme adulte, ce qui correspond à de 5 à 6 litres de liquide. Il est constitué à 55 % de plasma et à 45 % de cellules appelées éléments figurés.

Le sang exécute de nombreuses fonctions importantes. Par la voie de l'hémoglobine contenue dans les érythrocytes, il transporte de l'oxygène vers les tissus et recueille le gaz carbonique (CO₂). Il transporte aussi des substances nutritives (p. ex. les aminoacides, sucres et sels minéraux) et rassemble la matière excrétée qui sera éliminée par le filtre rénal. Le sang transporte également des hormones, des enzymes et des vitamines. Il assure la défense de l'organisme grâce à l'activité phagocytaire des leucocytes, au pouvoir bactéricide du sérum et à la réaction immunitaire dont les lymphocytes sont les protagonistes.

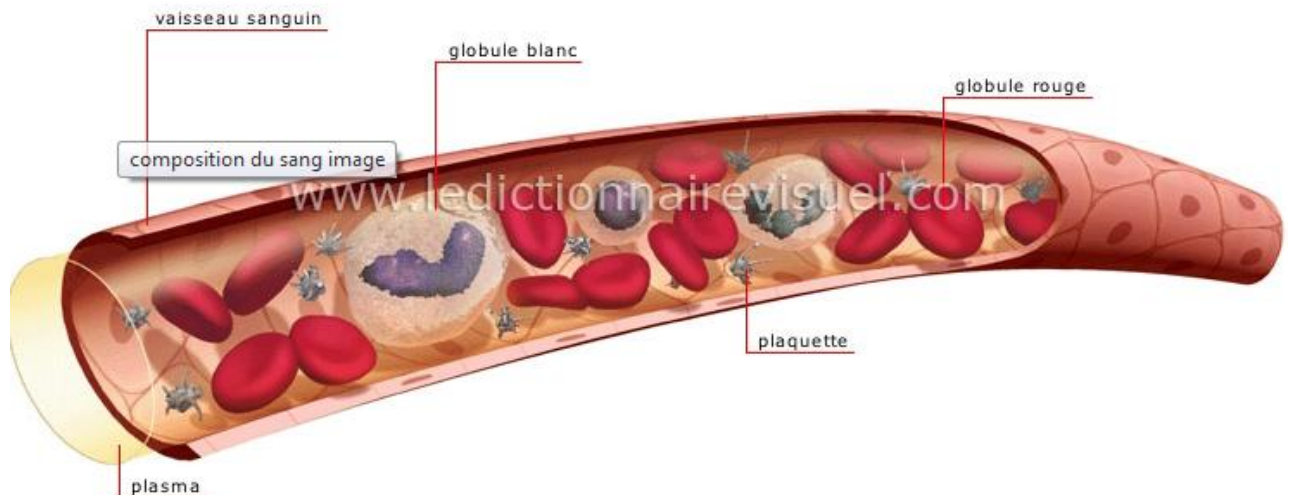


Figure 1.1 : composition du sang

III.1. Le plasma

Le sérum dépourvu de cellules, ou plasma, peut être obtenu au moyen de la centrifugation. Le plasma est un liquide légèrement alcalin, ordinairement jaunâtre. Il est constitué à 90 % d'eau et à 10 % de matières sèches. Neuf/dix de celles-ci sont constituées de substances organiques, la dernière l'étant de minéraux. Ces substances organiques sont composées de glucides (glucose), de lipides (cholestérol, triglycérides, phospholipides, lécithine, gras), de protéines (globulines, albumines, fibrinogènes), de glycoprotéines, d'hormones (gonadotrophine, érythropoïétine, thrombopoïétine), d'acides aminés et de vitamines. Les substances minérales sont dissoutes sous forme ionique, c'est-à-dire dissociées en ions positifs et négatifs.

III. 2. Erythrocytes (globules rouges)

Les érythrocytes sont les cellules sanguines les plus nombreuses, à raison de 4 à 6 millions d'entre elles par mm³. Ils sont aussi appelés « globules rouges » ou « hématies ». Chez les humains comme chez tous les mammifères, les érythrocytes sont dépourvus de noyau et prennent la forme d'une lentille biconcave. Chez les autres vertébrés (p. ex., poissons, amphibiens, reptiles et oiseaux), ils ont un noyau. Les globules rouges sont riches en hémoglobine, une protéine capable de se lier faiblement à l'oxygène. Ainsi, ces cellules sont chargées de l'apport d'oxygène aux tissus et en partie de la récupération

du gaz carbonique rejeté par le corps. Toutefois, l'essentiel du CO₂ est transporté dans le plasma sous forme de carbonates solubles.

Dans les globules rouges des mammifères, l'absence de noyau libère de l'espace pour l'hémoglobine, et la forme biconcave de ces cellules accroît le rapport entre la surface et le volume cytoplasmique. Ces caractéristiques rendent plus efficace la diffusion d'oxygène par ces cellules. Lorsqu'il y a anémie falciforme, aussi appelée « drépanocytose », les érythrocytes prennent généralement la forme d'une faucille. À l'aide d'un microscope à électrons, des biologistes ont constaté que les globules rouges peuvent avoir des formes différentes : normale (discocyte), épineux (échinocyte), crénelé, codocyte, à fuseau, faucille, casque, pointu, indenté, poïkilocyte, etc. La durée de vie moyenne des érythrocytes est d'environ 120 jours. Une fois arrivés à la fin de leur vie, ils sont retenus dans la rate où ils sont phagocytés par les macrophages.

III.3. Les plaquettes

La fonction principale des plaquettes, ou thrombocytes, est de faire cesser l'écoulement du sang par les plaies (hémostase). A cette fin, elles s'agglutinent et libèrent des facteurs favorisant la coagulation du sang. On trouve parmi elles la sérotonine, qui réduit le diamètre des vaisseaux lésés et ralentit le flux sanguin, et la fibrine qui capture les cellules et donne lieu à la coagulation. Même si les plaquettes sont en apparence plutôt rondes, ce ne sont pas de véritables cellules. Dans les frottis colorés au Giemsa, elles prennent une couleur pourpre intense. En raison de leur diamètre d'environ 2-3 µm, elles sont beaucoup plus petites que les érythrocytes. Leur densité dans le sang est de 200 000 à 300 000/mm³.

III.4. Leucocyte (globules blancs)

Les leucocytes, ou globules blancs, sont chargés de défendre l'organisme. Ils sont beaucoup moins nombreux dans le sang que les globules rouges. La densité des leucocytes dans le sang est de 5 000 à 7 000/mm³. Les leucocytes se divisent en deux catégories : les granulocytes et les cellules lymphoïdes ou agranulocytes. Le terme « granulocytes » est dû à la présence de granules dans le cytoplasme de ces cellules. Les

granules varient selon les différents types de granulocyte, ce qui aide à les distinguer entre eux. En fait, ces granules ont des affinités différentes à l'égard des colorations neutres, acides ou basiques, et donnent au cytoplasme des couleurs différentes. Ainsi, on distingue les granulocytes entre les neutrophiles, les éosinophiles (ou acidophiles) et les basophiles, tandis qu'on distingue les cellules lymphoïdes entre les lymphocytes et les monocytes. Comme nous le constaterons plus loin, la forme du noyau aide elle aussi à reconnaître les leucocytes.

Chaque type de leucocyte est présent dans le sang en proportions différentes :

Neutrophiles, de 50 à 70 %

Eosinophiles, de 2 à 4 %

Basophiles, de 0,5 à 1 %

Lymphocytes, de 20 à 40 %

Monocytes, de 3 à 8 %

III.4.1 Les neutrophiles

Phagocytent très activement les bactéries et sont présents en nombre élevé dans le pus des plaies. Malheureusement, ces cellules sont incapables de renouveler les lysosomes utilisés dans la digestion des microbes et meurent après en avoir phagocyté quelques-uns

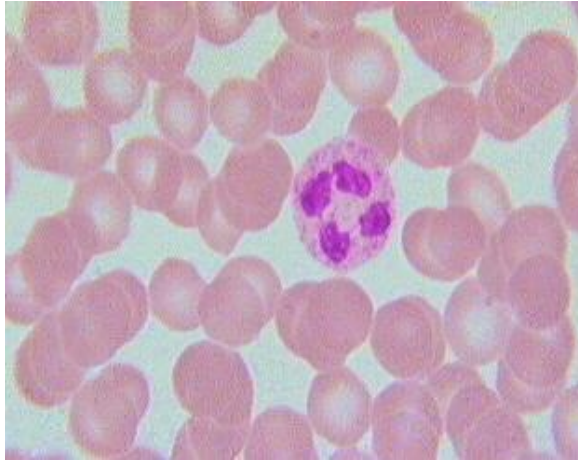


Figure 1.2 : neutrophiles

III.4.2 Les éosinophiles

Attaquent les parasites et phagocytent les complexes antigènes-anticorps.

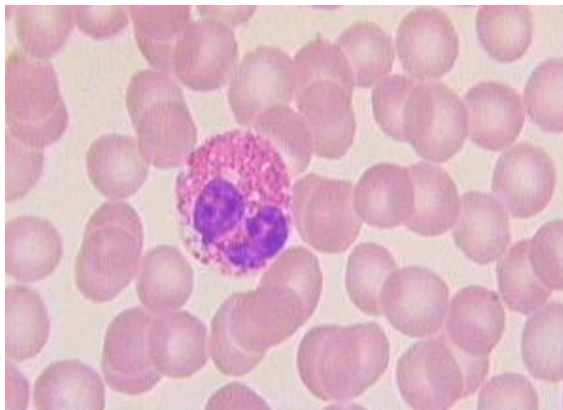


Figure 1.3 : Eosinophiles

III.4.3 Les basophiles

Sécrètent des substances anticoagulantes et vasodilatatrices comme les histamines et la sérotonine. Malgré leur capacité phagocytaire, leur principale fonction est de sécréter les substances qui servent à la médiation de la réaction d'hypersensibilité.

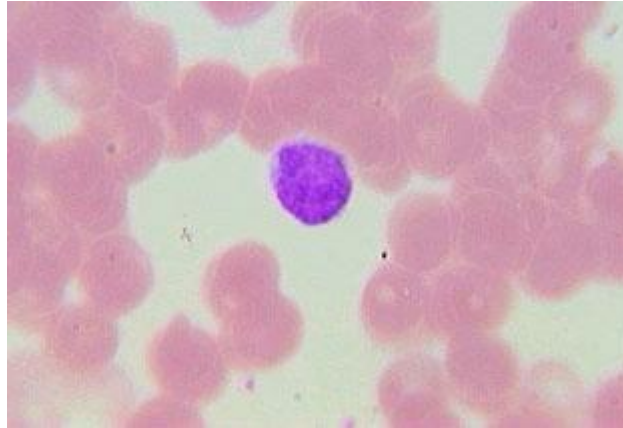


Figure 1.4: basophiles

III.4.4 Les lymphocytes

Sont des cellules qui, outre leur présence dans le sang, peuplent aussi les tissus lymphoïdes et les organes de même que la lymphe circulant dans les vaisseaux lymphatiques. Les organes lymphoïdes comprennent le thymus, la moelle osseuse (dans la bourse séreuse des oiseaux), la rate, les nodules lymphoïdes, les amygdales palatines, les plaques de Peyer et les tissus lymphoïdes du système respiratoire et du tube digestif.

Les lymphocytes sont les principaux éléments du système immunitaire, qui assure la défense contre les attaques de micro-organismes pathogènes comme les virus, bactéries, champignons et protistes. Les lymphocytes produisent aussi des anticorps et les disposent sur leur membrane. L'anticorps est une molécule capable de se lier à d'autres molécules de forme complémentaire appelée antigènes, et de les reconnaître. Les anticorps sont codifiés par des gènes, comme le sont toutes les protéines. En fonction du mécanisme de recombinaison de certains de ces gènes, chaque lymphocyte produit des anticorps d'une forme particulière.

En microscopie optique, ce sont des cellules de petites tailles, environ 7 μm de diamètre avec un noyau occupant la quasi totalité de la cellule. Leur forme est régulière

et arrondie. Il existe une petite frange cytoplasmique périphérique d'aspect mauve au MGG. Le noyau est sphérique.

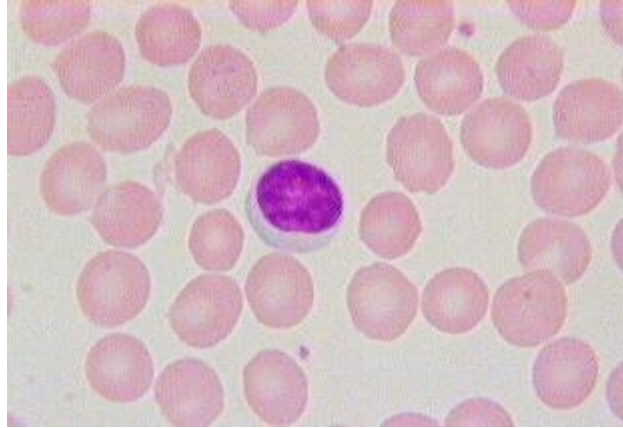


Figure 1.5: lymphocytes

III.4.5 Les monocytes

Sont les précurseurs des macrophages. Ce sont des cellules sanguines plus grosses qui, après avoir atteint leur maturité dans la moelle osseuse, entrent dans le flux sanguin où elles demeurent pendant de 24 à 36 heures. Elles migrent ensuite vers les tissus conjonctifs, où elles deviennent des macrophages et se déplacent dans les tissus. S'ils parviennent au site d'une inflammation, les monocytes migrent rapidement à partir du vaisseau sanguin et se lancent dans une intense activité phagocytaire. La phagocytose n'est pas le rôle exclusif de ces cellules car elles ont aussi une activité de sécrétion intense. Elles produisent des substances exerçant des fonctions de défense

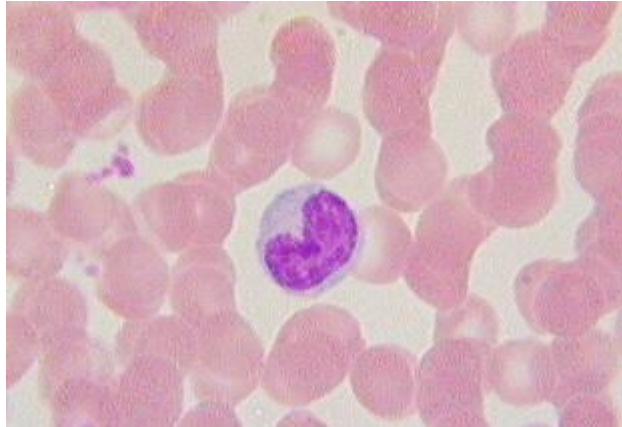


Figure 1.6 : monocytes

En microscopie électronique, la chromatine est fine, les organites bien développés et situés dans l'encoche du noyau. Il existe de nombreuses granulations azurophiles, de petite taille correspondant à des lysosomes. La membrane plasmique est irrégulière avec de nombreuses expansions et microvillosités. Les monocytes représentent 2 à 10 % de l'ensemble des globules blancs. [19]

IV. Conclusion

L'acquisition des images est simple et facile grâce à des outils plus performants mais ce n'est pas forcément qu'elle nous donne des bons résultats. Il y a pas mal des erreurs lors d'acquisitions amenées par l'utilisateur ou par l'appareil elle-même qu'on peut les résoudre par l'un des étapes de prétraitement qu'on va voir en détail dans le chapitre suivant.

Chapitre 2 : acquisition et prétraitement d'image numérique

I. Introduction

Un dispositif d'acquisition capture une information lumineuse et la convertit en un signal électronique qui peut être échantillonné, quantifié et stocké dans un fichier informatique sous la forme d'une image numérique qu'on peut faire des traitements sur ces images.

II. Acquisition des images couleurs

L'acquisition d'une image simple est faite à l'aide des appareils photo numériques, des caméras industrielles, des webcams, des caméras infrarouge...

En médecine, il existe des imageurs tels que l'IRM, TEP, scanner X, échographie Doppler, échographie, scintigraphie, Tomographie, etc.

Tous ces systèmes peuvent être comparés à des capteurs. Il ne faut pas oublier qu'il y a une étape de conversion analogique/numérique. C'est souvent cette étape qui limite la résolution de l'image. L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel). [15]

➤ Processus d'acquisition des cellules sanguines

L'acquisition d'une image de cellule de sang se fait par un microscope en passant par les étapes suivantes :

- On prend du sang d'un patient.
- Etalement d'une goutte sur la lame.
- Fixation : La procédure de fixation permet :
 - D'augmenter la visibilité des cellules,
 - D'accentuer les particularités morphologiques des cellules,
 - De conserver l'échantillon en vue d'expériences ultérieures.
- Coloration : Après la fixation, vient l'étape de coloration qui met en évidence certaines structures afin d'en faciliter l'étude.

_ Bleu de méthylène

_ Eosine

- Acquis par un microscope
- Puis on la passe sur un ordinateur
- Une meilleure acquisition de l'image d'une plaque cellulaire permet d'avoir une bonne qualité en termes d'exploitation des informations que contient cette dernière. Même avec une meilleure acquisition nous avons besoin d'appliquer des traitements sur l'image. Afin de rendre nos résultats plus fiables.

La figure suivante montre l'acquisition de l'image cellulaire :

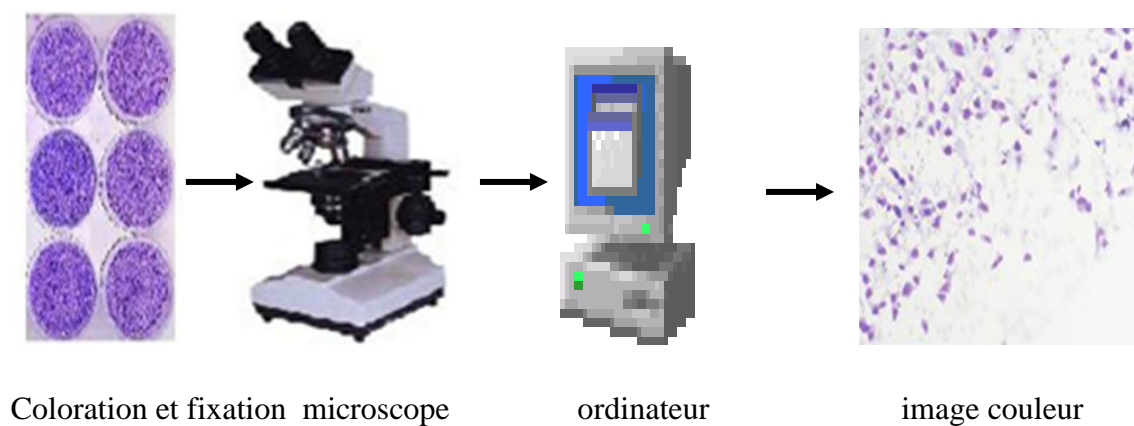


Figure 2.1: l'acquisition de l'image cellulaire

III. Prétraitement de l'image

Après l'obtention de l'image on la prépare avant le traitement (segmentation et classification) pour obtenir des meilleurs résultats est voici quelle que technique.

Restauration d'image : [20]

La restauration d'images est la première étape de tout traitement d'image. Elle permet de diminuer le bruit introduit lors de l'acquisition ou dû à la compression des données. Néanmoins, si l'information essentielle n'est pas présente sur l'image initiale, elle ne pourra pas être retrouvée par la suite. Il est donc toujours important d'assurer, autant que possible, une bonne acquisition des données.

On peut trouver plusieurs types de bruit

Bruit lié au contexte de l'acquisition : Bougé, dérive lumineuse, flou, poussière, ...

Bruit lié au capteur : distorsion de la gamme des niveaux de gris, distorsion géométrique, mauvaise mise au point, ...

Bruit lié à la numérisation : Codage, quantification, échantillonnage (moiré, effet poivre et sel), ...

Exemple :



Image source



Flou de bougé

Figure 2.2 : Exemple de bruit

III.1 Amélioration d'image

III.1.1 Seuillage

Le seuillage consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur (appelée seuil, en anglais treshold) et à la valeur maximale les pixels ayant une valeur supérieure. Ainsi le résultat du seuillage est une image binaire contenant des pixels noirs et blancs. Le seuillage permet de mettre en évidence des formes ou des objets dans une image. Toutefois la difficulté réside dans le choix du seuil à adopter. [17]

III.1.2 Egalisation d'histogramme

L'histogramme d'une image est une fonction discrète. Elle représente le nombre de pixels en fonction du niveau de gris.

Tandis que L'égalisation d'histogramme permet de d'harmoniser la répartition des niveaux de luminosité de l'image, de telle manière à tendre vers un même nombre de pixel pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image.



Avant

Après

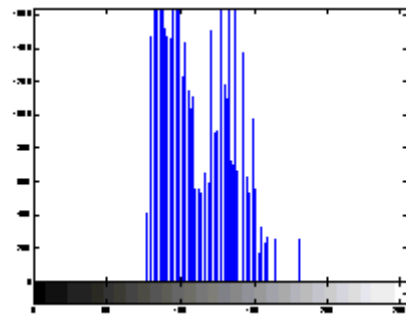
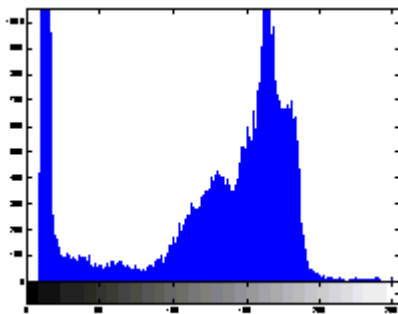


Figure 2.3 : Egalisation de l'histogramme

III.1.3 Rehaussement de contraste

Le principe de rehaussement consiste à mesurer les différences entre le niveau de gris d'un pixel et celui de son voisinage et à les augmenter.

L'opération de rehaussement de contraste devra, réduire le bruit dans les zones stationnaires et éviter les phénomènes de dépassement.

Pour l'opération de rehaussement et même pour l'autre opération de prétraitement la connaissance de grandeur fournissant des informations sur les donnée de l'image s'avère important, parmi ces Operations en peut citer :

III.1.4 Minimum et maximum de niveaux de gris

Le minimum et maximum de niveaux de gris sont respectivement la plus petite valeur et la plus grande valeur de gris de l'image, ce sont la borne minimale et maximale. la distribution présente une grande dynamique si cet intervalle est large.

Moyenne

Comme la précédente, la moyenne est une grandeur de position. Elle est obtenue en calculant la somme des valeurs des pixels de l'image divisé par l'effectif totale

III.1.5 Squelettisation

En termes simples, la squelettisation consiste à amaigrir une forme jusqu'à obtenir un ensemble de courbes centrées. L'ensemble obtenu est alors appelé **squelette** ou **axe médian**.

La squelettisation est une opération qui nous permet tout simplement de réduire l'espace entre les différentes particules de telle manière que la frontière soit de largeur 1 pixel seulement. En fait, cela crée une frontière au milieu deux particules adjacentes. L'algorithme consiste en fait à faire croître chaque particule de manière successive et de définir une frontière là où deux particules suffisamment dilatées se rencontrent

III.2 Restauration d'image par filtrage

III.2.1 Définition Filtrage

Un filtre est un dispositif qui transforme un signal; favorisant certaines fréquences et en Amortissant d'autres.

Filtrage permet de réduire les variations d'intensité au sein de chaque région de l'image tout en respectant l'intégrité des scènes.

III.2.2 Convolution

La plupart des filtres de traitement des images utilisent des matrices de convolution pour faire des liens entre différent filtre.

Une convolution est un traitement d'une matrice par une autre appelée matrice de convolution ou « noyau » qui consiste à recalculer l'intensité de chaque pixel en fonction des pixels voisins. Le filtre étudie successivement chacun des pixels de l'image. Pour chaque pixel, que nous appellerons « pixel initial », il multiplie la valeur de ce pixel et de chacun des 8 pixels qui l'entourent par la valeur correspondante dans le noyau. Il additionne l'ensemble des résultats et le pixel initial prend alors la valeur du résultat final [21]

Matrice de convolution :

Cas discret : Soit une image monochrome et $f(x, y)$ intensité lumineuse du pixel de coordonnées (x, y) . La convolution numérique de cette fonction avec une réponse impulsionnelle $h(m, n)$ conduit à une nouvelle image de fonction $g(x, y)$ qui s'écrit :

$$g(x, y) = f(x, y) * h(x, y) = \sum_{m=-d}^d \sum_{n=-d}^d h(m, n) \cdot f(x + m, y + n)$$

ou $h(m, n)$ est appelée masque de convolution sa taille est $(2d+1) \times (2d+1)$, ou $d=1$.

Cas continue : $g(x, y) = f(x, y) * h(x, y) = \int \int f(x - u, y - v) \times h(u, v) \, du \, dv$

Exemple de masque :

$$\begin{array}{c} A \\ \begin{array}{ccc} \boxed{1} & \boxed{1} & \boxed{1} \\ \boxed{1} & \boxed{1} & \boxed{1} \\ \boxed{1} & \boxed{1} & \boxed{1} \end{array} * 1/9 \\ m \end{array}$$

III.2.3 Type de filtrage

III.2.3.1 Filtrage linéaire

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution. Le temps de calcul est souvent réduit lorsqu'on veut séparer un filtre en deux filtres dont la convolution mutuelle permet de le reconstituer. Cette remarque est utilisée en particulier pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical

Le filtre local est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage.

✓ Filtre passe-bas (lissage) :

Consistant à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle on parle habituellement de lissage. Les filtres moyennés sont un type de filtres passe-bas dont le principe est de faire la moyenne des valeurs des pixels avoisinants. Le résultat de ce filtre est une image plus floue.

✓ **Filtre passe-haut (accentuation) :**

C'est l'inverse des passe-bas, atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, les filtres passe-bande permettant d'obtenir la différence entre l'image originale et celle obtenue par application d'un filtre passe-bas. Filtre Gaussien :

✓ **Le filtre Gaussien**

Est un filtre isotrope spécial avec des propriétés mathématiques bien précises et très communes dans la nature.

La fonction gaussienne est souvent utilisée dans les distributions statistiques elle est définie par la fonction $G(x)$ (dans le traitement d'image on traite des données à deux dimensions (x,y) , on introduit dans une fonction gaussienne à deux dimensions $G(x,y)$

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

En général un filtre gaussien avec un $\sigma < 1$ est utilisé pour réduire le bruit et si $\sigma > 1$ c'est dans le but de fabriquer une image qu'on va utiliser pour faire « masque flou » personnalisé, il faut noter que plus σ est grand, plus la cloche gaussienne est large et le flou appliqué à l'image sera marqué.

Le filtre gaussien donnera un meilleur lissage et une meilleure réduction du bruit que le filtre passe-bas.

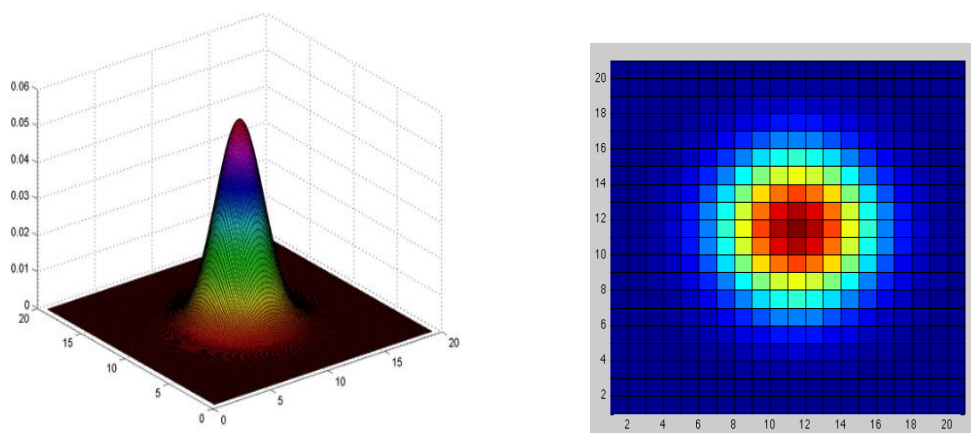


Figure 2.4 : filtre gaussien



Figure 2.5 : exemple de filtre gaussien 5*5 et 11*11

III.2.3.2 Filtres non linéaires

Ils sont conçus pour régler les problèmes des filtre linéaires, surtout ce qui concerne la mauvaise conservation des contours, leur principe est le même que les filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage. La différence est que cette fonction n'est plus linéaire mais une fonction quelconque.

✓ Filtres médian :

Le filtrage médian est un filtre non linéaire de la famille des filtres d'ordre. Son objectif est la réduction du bruit impulsionnelle. Son principe consiste à remplacer un pixel par la médiane de ses voisins.

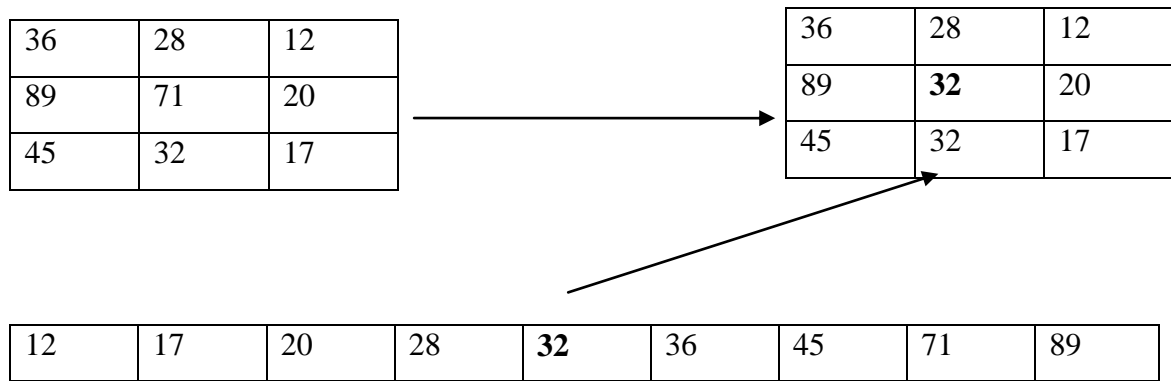


Figure 2.6 : Principe du filtre médian

✓ **Le filtre maximum**

On applique le même traitement que celui du filtre médian mais la valeur du pixel du centre va être remplacée par le maximum.

✓ **Le filtre minimum :**

On applique le même traitement que celui du filtre médian mais, cette fois, la valeur du pixel du centre va être remplacée par le minimum.

✓ **Filtre de nagao :**

Ce filtrage a été présenté dans l'article "Edge preserving smoothing" par M. Nagao et T. Matsuyama, dans le journal Computer Graphics and Image Processing de 1979. Il permet un lissage de l'image conservant les contours, dans l'optique d'un prétraitement pour une segmentation par régions. Le principe de ce filtrage est de remplacer chaque pixel de l'image par la valeur moyenne des pixels contenus dans une fenêtre particulière. Il s'agit de choisir la fenêtre la mieux adaptée parmi un certain nombre de fenêtres prédéfinies. [22]

III.3 Détection de contour

La détection de contours est une technique de réduction d'information dans les images, qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image. Si les structures extraites sont simples à manipuler (courbes fines, régulières, stables...), elles peuvent être utiles pour la mise en correspondance d'images (robotique, indexation...)

Le développement théorique pour la détection des contours sont étudiés dans le domaine continu puis discrétiser pour la programmation. Dans le domaine continu, les approches dérivatives sont les plus intuitives pour détecter la variation d'amplitude.

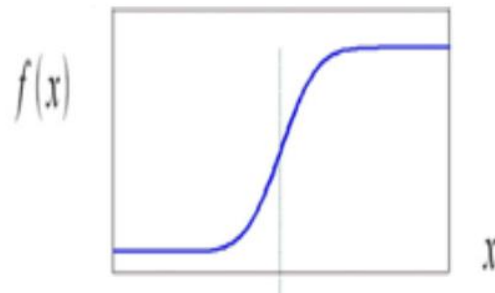


Figure 2.7 : La fonction d'image

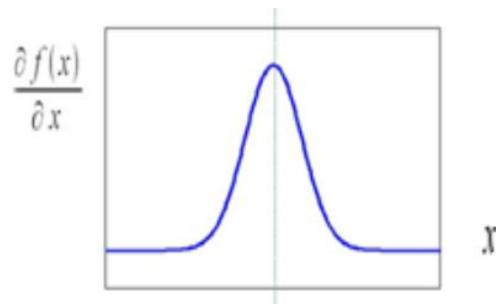


Figure 2.8 : Le maximum de la fonction image

Et une dérivée seconde présentera un passage par zéro du contour, ce qui est montré dans :

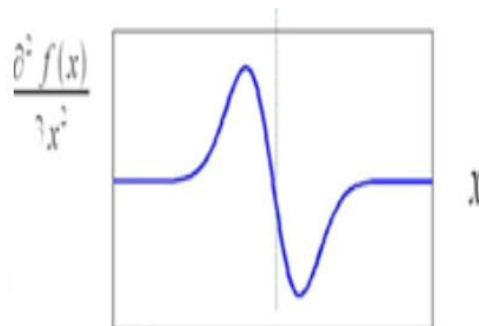


Figure 2.9 : Le passage par zéro de la fonction image

L'identification d'une transition peu être obtenue par le seuillage de la norme du gradient.

Une image est représentée par une fonction f :

$$f: \mathbb{R}^2 \longrightarrow \mathbb{R}$$

$$P(x,y) \longrightarrow f(x,y)$$

Les composants du vecteur gradient de la fonction f est défini au point de coordonnées (x,y) par :

$$\text{Grad}(x,y) = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix} = \begin{pmatrix} f_x(x,y) \\ f_y(x,y) \end{pmatrix}$$

Le module ou la norme du gradient est défini par :

$$\text{Norme} = \sqrt{(\partial f / \partial x)^2 + (\partial f / \partial y)^2}$$

Tandis que son orientation ou la phase est définie par :

$$\text{Phase} = \arctg((\partial f / \partial x) / (\partial f / \partial y)).$$

Dans le cas de discret les dérivées dans les direction vertical et horizontal ou point des cordonnés (x,y) sont approcher par des différence fini

$$f_x(x,y) = f(x+1,y) - f(x,y)$$

$$f_y(x,y) = f(x,y+1) - f(x,y)$$

Le calcul du gradient sur une image peut être obtenu par deux convolution :

L'une mettant en évidence le contour horizontaux et l'autre verticaux.

$$f_x(x,y) = f(x,y) \otimes W_x(x,y)$$

$$f_y(x, y) = f(x, y) \otimes W_y(x, y)$$

Où W_x correspond au filtre de dérivation et W_y correspond au filtre de dérivation vertical, le vecteur du gradient est donc $[f_x, f_y]$. [4]

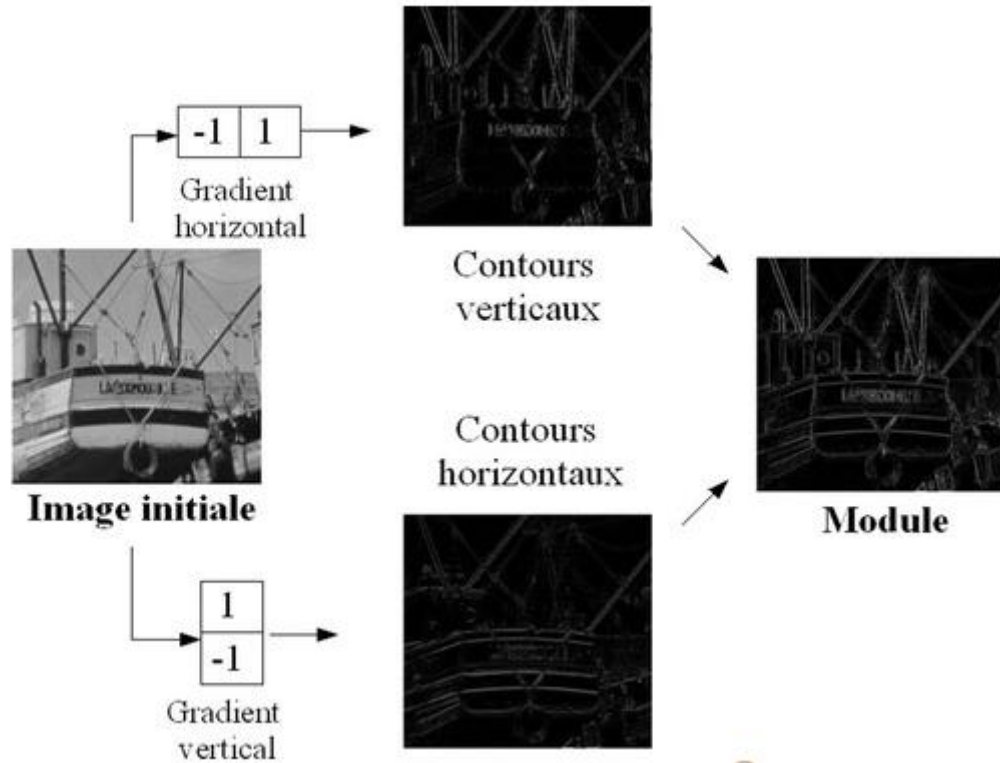


Figure 2.10: détection de contour

Il ya pas mal d'opérateur qui utilisent le gradient pour la détection des contours voici quelle que uns :

III.3.1 Opérateur de laplacien

Les opérateurs de gradient vus précédemment exploitent le fait qu'un contour dans une image correspond au maximum du gradient dans la direction orthogonale au contour. Or le passage par zéro de la dérivée seconde d'une rupture d'intensité permet également de mettre en évidence le contour. La dérivée seconde est donc déterminée par le calcul du Laplacien : [5]

$$L(x, y) = \partial^2 f(x, y) / \partial x^2 + \partial^2 f(x, y) / \partial y^2$$

L'opérateur de laplacien est aussi un filtre de convolution particulier utilisé pour mettre en valeur les détails qui ont une variation rapide de luminosité. Dans le cas de traitement d'image, l'image de départ $I(x,y)$ n'est plus une fonction continue, mais une fonction discrète a cause de la numérisation effectuée. Mais on peut tout même obtenir la dérivée seconde en bonne approximation. Il existe trois noyaux typiques de taille 3x3 qui peuvent remplir ce rôle :

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

1	-2	1
-2	4	-2
1	-2	1

Figure 2.11 : Masque de convolution LAPLACIEN

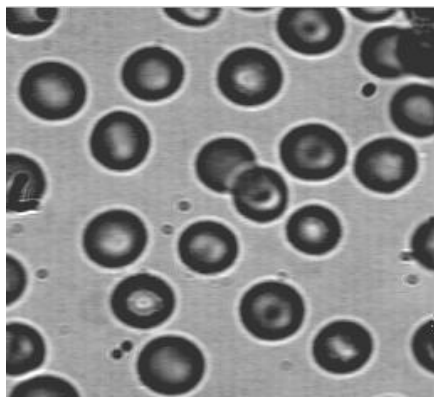
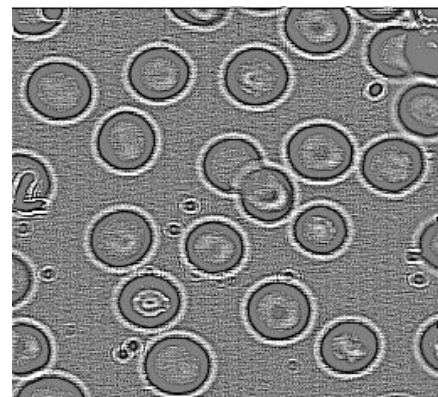


Image originale



Laplacien

Figure 2.12 : Opérateur de laplacien

III.3 .2 Opérateur de SOBEL

Il s'agit d'un des opérateurs les plus simples qui donne toutefois des résultats corrects. Cet opérateur est très sensible aux bruits, il est utilisé pour la détection des contours correspondant à des changements brusques des propriétés physiques ou géométriques de la scène. Les masques de détection de contour de Sobel détectent séparément les contours verticaux et horizontaux. On obtient après convolution une image des gradients. Cette image pourra par la suite être modifiée dans une phase de seuillage pour ne contenir que des pixels à 1 ou à 0 correspondant respectivement à un point de contour ou à un non point de contour.

Les dérivées directionnelles horizontale et vertical s'expriment sous la forme :

$$f_i [i,j]= h_i * f [i,j] \quad \text{ET} \quad f_j [i,j]= h_j * f [i,j]$$

$$\text{Avec } h_i = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{et} \quad h_j = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Figure 2.13: Masque de convolution

Horizontale de SOBEL

Figure 2.14 : Masque de convolution

verticale de SOBEL

Exemple :

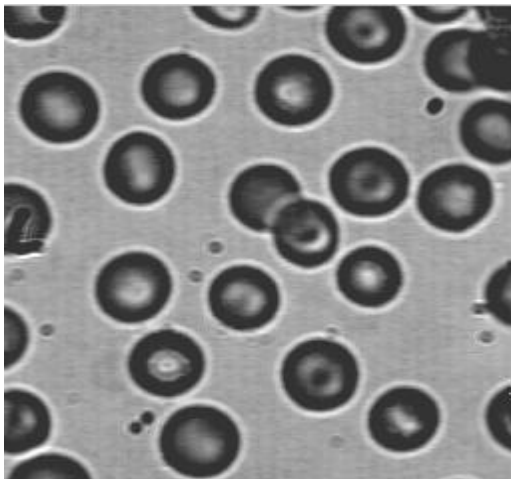


Image originale

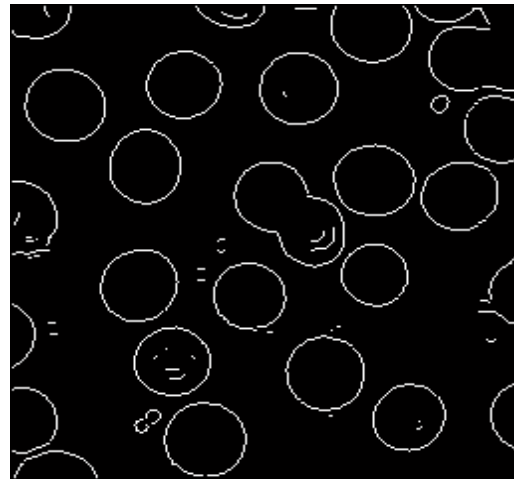


Image filtré (SOBEL)

Figure 2.15: Opérateur de SOBEL

III.3.4 Opérateur de ROBERT

Cet opérateur est très sensible au micro contour, mais aussi au bruit ; il donne des contours fins mais incomplets, il est donc appliqué aux régions à structure très fine. la qualité des contours est assez bonne.

Ceci revient à convoluer l'image avec les deux masques suivants :

1	0
0	-1

0	1
-1	0

Figure 2.16 : Masque de convolution 1

Figure 2.17 : Masque de convolution 2

de ROBERT

de ROBERT

Exemple

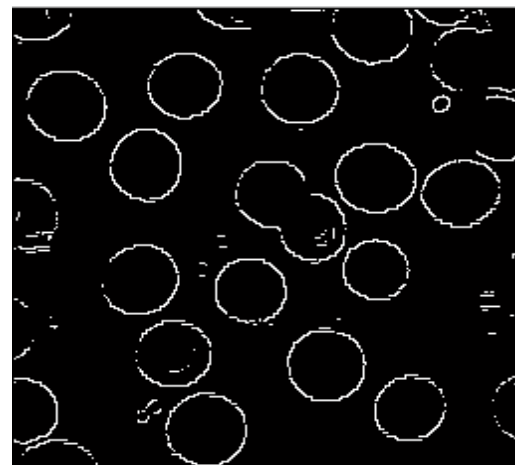
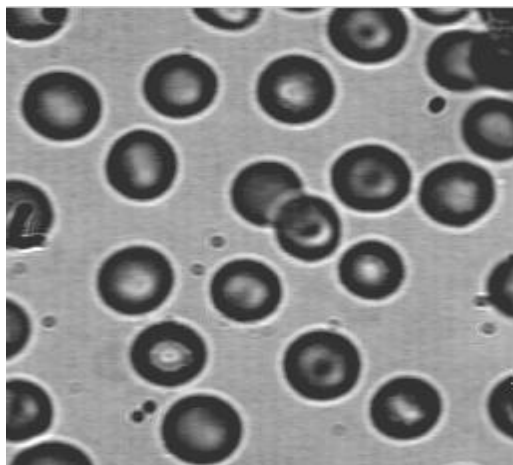


Image original

image filtré (ROBERT)

Figure 2.18: Opérateur ROBERT

III.3.5 Opérateur de PREWITT

Contrairement à l'opérateur de ROBERT, cet opérateur présente l'avantage d'être moins sensible aux bruits ; il favorise les contours à structure linéaire mais la qualité des contours est meilleure.

$$f_i[i, j] = h_i * f[i, j] \quad \text{Et} \quad f_j[i, j] = h_j * f[i, j]$$

Ainsi les deux masques de convolution de ce filtre sont les suivant :

$$h_i = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$h_j = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Figure 2.19 : Masque de convolution

Figure 2.20 : Masque de convolution

Horizontale de PREWITT

verticale de PREWITT

Donc la norme et l'orientation du gradient sont :

$$|\nabla f[i, j]| = \sqrt{(f^2_j[i, j] + f^2_i[i, j])}$$

$$\text{Et} \quad \theta = \frac{\pi}{2} - \text{arctg} \left(\frac{f_i}{f_j} \right)$$

Exemple :

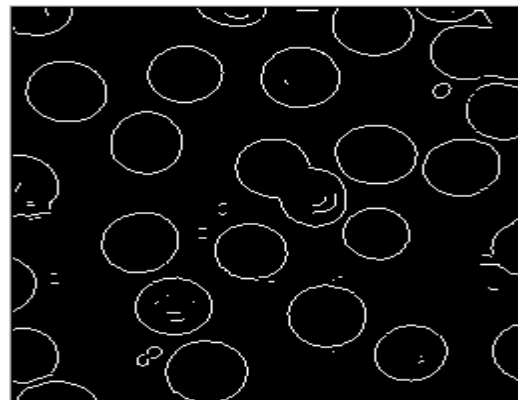
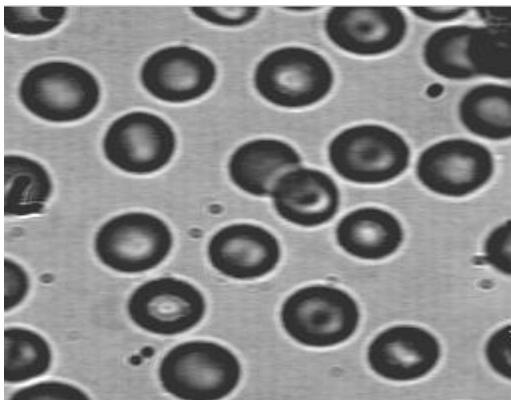


Image originale

image filtré (PREWITT)

Figure 2.21: Opérateur de PREWITT [6]

III.4 Morphologie mathématique

La morphologie mathématique est un ensemble de méthodes d'analyse d'images mis au point à l'Ecole des Mines de Fontainebleau par l'équipe de Jean Serra. Elle offre un grand nombre d'outils très puissants de traitement et d'analyse d'images que l'on retrouve sous différentes rubriques dans certains logiciels d'analyse d'images et même de retouche d'images. Les outils proposés ont été développés au départ pour traiter des images binaires : on fait alors de la morphologie mathématique ensembliste. Leur utilisation a été ensuite étendue aux images en niveaux de gris : on parle, à ce moment, de morphologie mathématique fonctionnelle. A noter que certains opérateurs fonctionnent désormais sur des images couleurs.

Le principe de base de la morphologie mathématique est de comparer l'image à analyser par rapport à un ensemble de géométrie connue appelé élément structurant que l'on déplace de façon à ce que son origine passe par toutes les positions de l'image, pour mettre en évidence certaines caractéristiques de l'image. [23]

Parmi les outils les plus importants de la morphologie mathématique on cite :

III.4.1 Erosion

On appelle érosion morphologique par l'élément structurant S , la soustraction de Minkowski par \check{S} , tel que $\check{S} = \{-s, s \in S\}$. L'érodé de X par S est l'ensemble des pixels pour lesquels S , translaté par x , est inclus dans X .

III.4.2 Dilatation

On appelle dilatation morphologique par l'élément structurant S , l'addition de Minkowski par \check{S} . Le dilaté de X par S est l'ensemble des pixels x pour lesquels S , translaté par x intersecte X

IV. Conclusion

La manipulation des images pose cependant des problèmes beaucoup plus complexe que celle du texte en effet l'image est un objet a deux dimensions, sensée représenté un espace a trois démentions induisant de conséquence majeure.

Le volume des données a traitée est beaucoup plus important, la structure de ses données est notamment plus complexe

Il résulte que la manipulation, le stockage et la représentation de ces données se heurtent à certaines limitations.

Mais grâce au prétraitement de l'image, ces contrainte sont levées ou contournées en effet, ce domaine cherche à détecter la présence de certaine contours ou certaine textures dans l'image et en même temps améliorer la qualité de l'image.

Bien que le prétraitement soit important pour la segmentation d'images, surtout les images médicales qui se caractérise par la complexité des textures et des contours qu'elle représentent.

Chapitre 3 : Segmentation et Méthode de classification

I. Introduction

L'étape de comptage des globules va passer par deux grandes parties : La segmentation des images qui constitue le cœur de tout système de vision, c'est une étape importante dans le processus d'analyse des images en utilisant l'algorithme Mean shift, et la partie de la classification automatique d'image en utilisant un type d'apprentissage spécifique.

Ces deux étapes ont comme but de déterminer les objets correspondants est le dénombrement des objets détectés.

II. Segmentation

La segmentation est une des étapes critiques de l'analyse d'images qui conditionne la qualité des mesures effectuées ultérieurement. En effet le terme segmentation a plusieurs définitions selon le type d'image sur laquelle on travaille, et selon la nature des outils de segmentation utilisés et surtout ce que l'on attend de cette procédure. Dans notre cas on définit la segmentation est l'opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image.

Il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes :

1. Segmentation fondée sur les régions (en anglais : *region-based segmentation*). On y trouve par exemple : la croissance de région (en anglais : *region-growing*), décomposition/fusion (en anglais : *split and merge*)
2. Segmentation fondée sur classification ou le seuillage des pixels en fonction de leur intensité (en anglais : *classification* ou *thresholding*)
3. Segmentation fondée sur les contours (en anglais : *edge-based segmentation*) comme : mean-shift. [7]

Exemple :

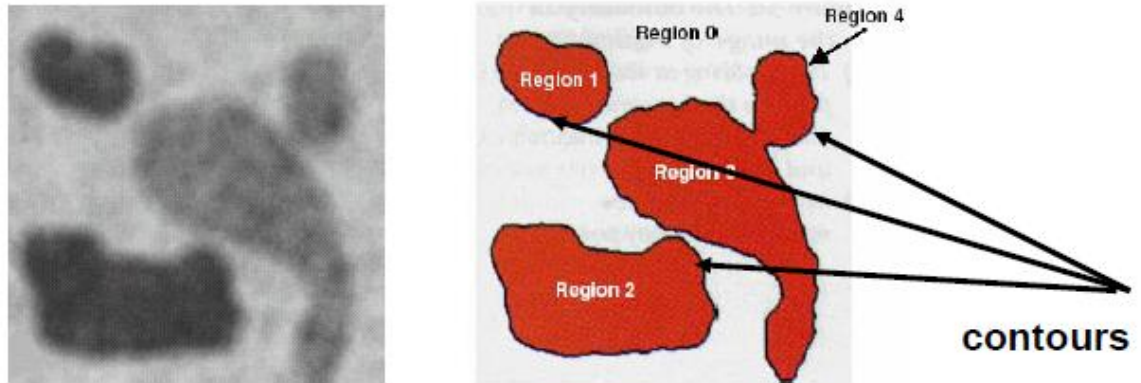


Figure 3.1 : segmentation

II.1. segmentation par région :

La segmentation par région consiste à prendre une région qui est un ensemble de pixels connexes ayant des propriétés communes qui les différencient de celles des autres régions on modifiant ces régions par regroupement ou divise la région par la méthode Division-Fusion (split-and-merge), ou on ajoute des pixels à des pixels voisins on parle de la méthode croissance de régions. Il y a d'autres méthodes fondées sur la modélisation statistique conjointe de la régularité des régions et des niveaux de gris de chaque région existent également. [8]

II.2. segmentation par classification (seuillage)

Le seuillage est une opération sur les pixels qui prend les coordonnées des points de l'image dans l'espace de caractéristiques retenu comme critère d'homogénéité comme par exemple la médiane, permettant de construire n classes d'intensité. Son principe est de décomposer l'ensemble de points ainsi obtenus en nuages de points distincts. On distinguera essentiellement de méthodes dans cette catégorie : les méthodes basées sur l'analyse d'histogrammes (pour la couleur) et les méthodes basées sur le clustering.

III.3. Segmentation par détection de contours

L'intérêt principal des méthodes de segmentation selon l'approche contours est de minimiser le nombre d'opérations nécessaires en cas d'itération du processus sur des séries d'images peu différentes les unes des autres. En effet, une fois que les contours des régions ont été trouvés dans la première image, l'application du modèle déformable

à l'image suivante est plus efficace que de tout recalculer, si la différence entre les images est peu importante. Parmi les algorithmes qui se reposent sur cette approche nous trouvons la méthode mean shift. [8]

II.3.1. Algorithme Mean Shift

Le mean-shift est un algorithme non paramétrique (non-paramétrique. couvre des techniques qui ne suppose pas que la structure d'un modèle est fixe. Typiquement, le modèle se développe en taille pour s'adapter à la complexité des données).

Mean shift a d'abord été proposé par Fukunaga et Hostetler (Fukunaga et Hostetler, 1975), adapté plus tard par Cheng (Cheng, 1995) dans le but d'analyse d'images, et plus récemment prolongé par Comaniciu, Meer et Ramesh à des problèmes de vision bas-niveau, y compris, la segmentation (Comaniciu & Meer, 2002), adapté lissage (Comaniciu & Meer, 2002) et suivi (Comaniciu, Ramesh & Meer, 2003) [9]

L'idée principale derrière mean shift est de traiter les points en fonction de la dimension d où chaque pixel est représenté par un point ayant pour coordonnées sa position en x , en y , et ses valeurs R , G , B . Les groupes de points dans cet espace représentent les ensembles de pixels homogènes à la fois spatialement et en couleurs.

II.3.2. Principe de l'algorithme

- Mean-Shift filtrage préservé discontinu (lissage discontinu).
- Déterminer les pôles (clusters) $\{C_p\}_{p=1..n}$.
- Attribuer des étiquettes de classe pour les clusters, d'éliminer les plus petites régions que les pixels P .

Pour la détection des clusters :

Mean Shift est un algorithme itératif qui a pour objectif de faire converger un point vers le maximum local le plus proche :

1. On commence par choisir un point de départ P .
2. On cherche l'ensemble E des points qui sont dans le voisinage de P .
3. On déplace P vers l'isobarycentre de E .

4. On réitère depuis l'étape 2 jusqu'à convergence.

Les déplacements successifs vers l'isobarycentre font converger le point P vers les zones de fortes densités. [10]

Exemple d'application:

Mean-Shift clustering: détection d'un cluster n points

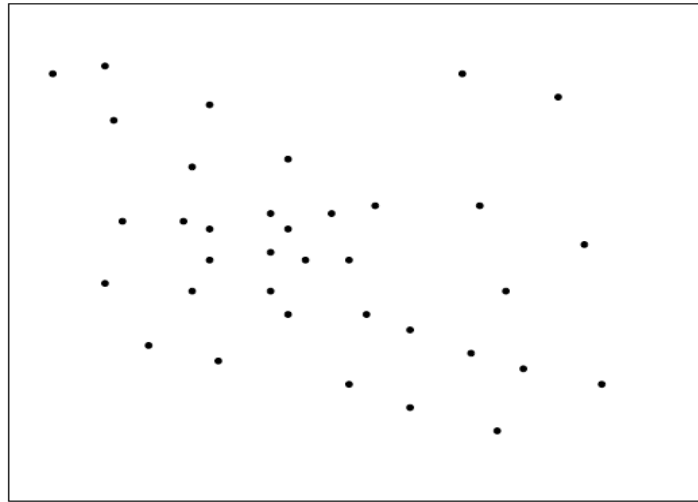


Figure 3.2 : ensemble des points d'image

n points 1 semence

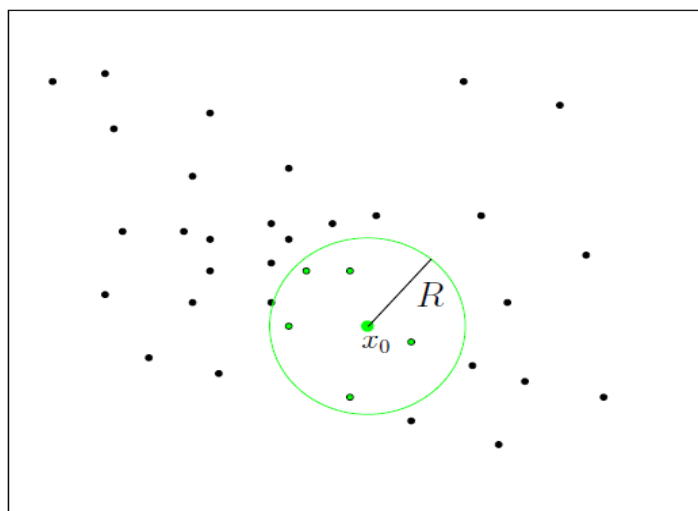


Figure 3.3 : point de départ

Dan un premier temps l’algorithme commence par choisir un point de départ P, et cherche les points de voisinage de P.

n points

1 semence

R rayon de la fenêtre

$$x_{i+1} = \text{signifie} \{ \text{voisine} (x_i) \}$$

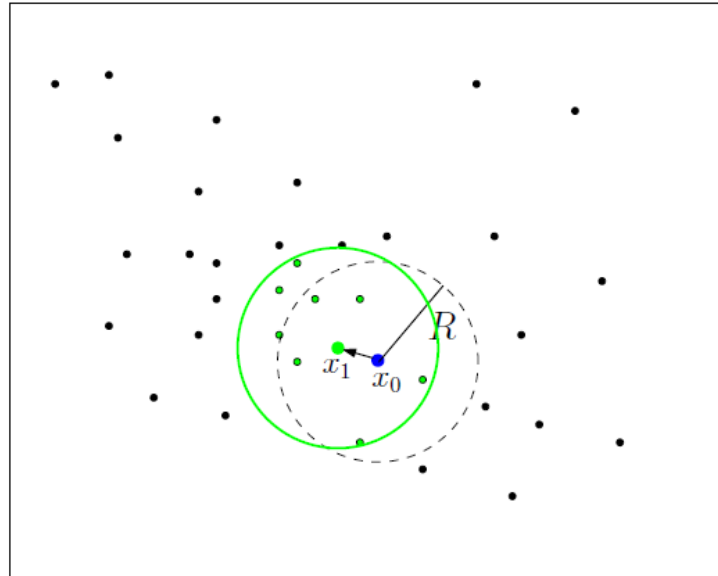


Figure 3.4 : le premier déplacement.

Après calcul de la densité, l’isobarycentre se déplace vers le point de densité plus élevé.

n points

1 semence

R rayon de la fenêtre

$$x_{i+1} = \text{signifie} \{ \text{voisine} (x_i) \}$$

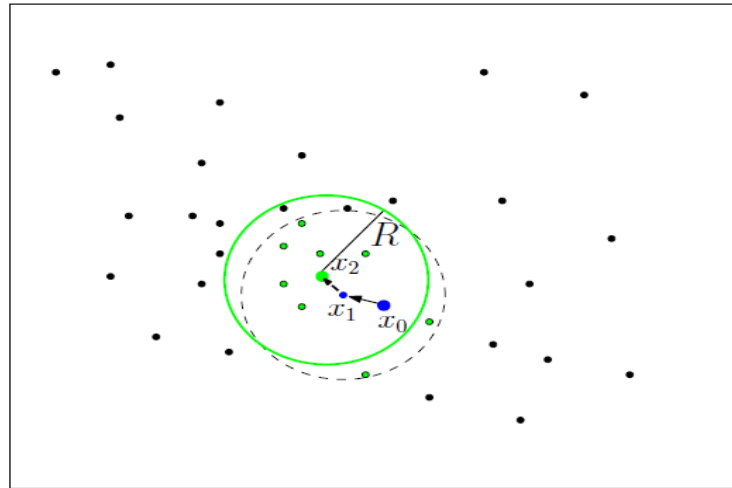


Figure 3.5 : deuxième déplacement.

L'algorithme répète l'étape 2 illustré dans la figure 3.4

n points 1 semence R rayon de la fenêtre x_{i+1} = signifie { voisine (x_i) }

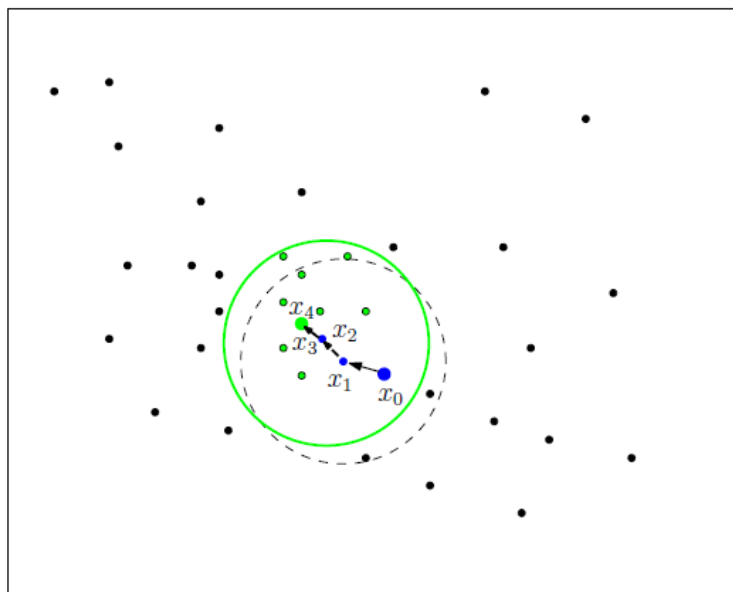


Figure 3.6 : troisième déplacement.

n points 1 semence R rayon de la fenêtre

x_{i+1} = signifie { voisine (x_i) }

Cluster détecté

On a convergence lorsque : $\nabla f(x) = 0$ x_5 point stationnaire du Mean-Shift

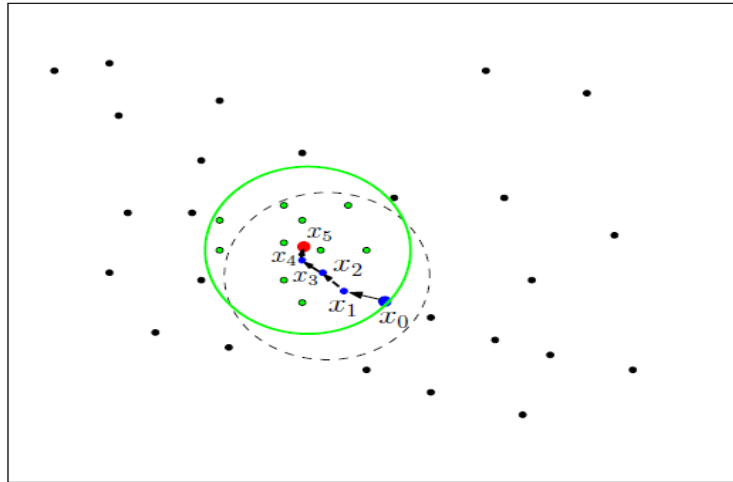


Figure 3.7 :quatrième déplacement.

n points 1 semence R rayon de la fenêtre x_{i+1} = signifie {voisine (x_i)}

Cluster détecté

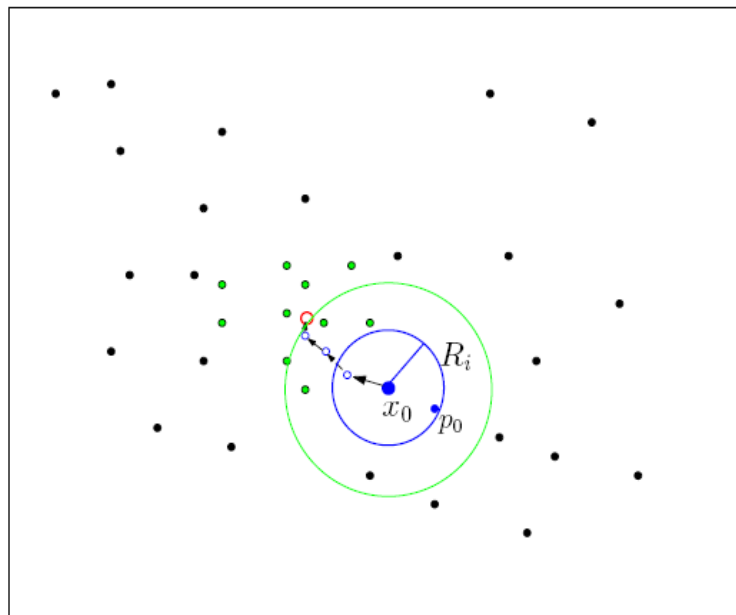


Figure 3.8 : cluster détecté

Le critère de convergence est vérifié.

n points 1 semence R rayon de la fenêtre x_{i+1} = signifie {voisine (x_i)}

Cluster détecté

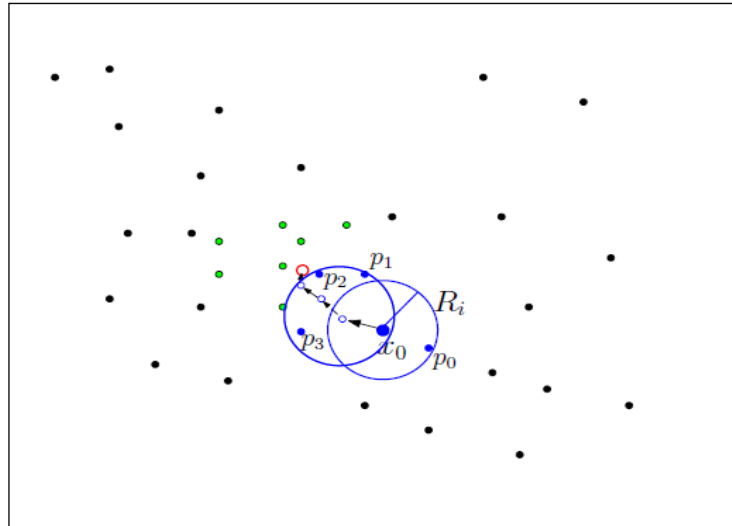


Figure 3.9 : détection des contours.

n points 1 semence R rayon de la fenêtre $x_{i+1} = \text{signifie } \{\text{voisine } (x_i)\}$
 Cluster détecté

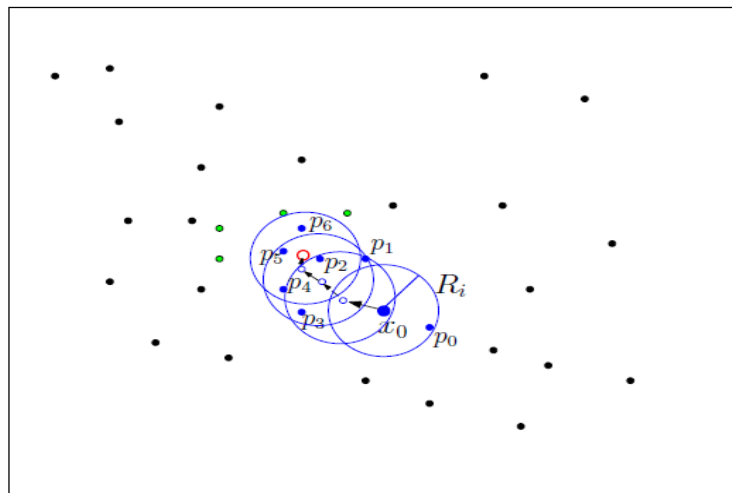


Figure 3.10 : détection des contours

n points 1 semence R rayon de la fenêtre $x_{i+1} = \text{signifie } \{\text{voisine } (x_i)\}$
 Cluster center

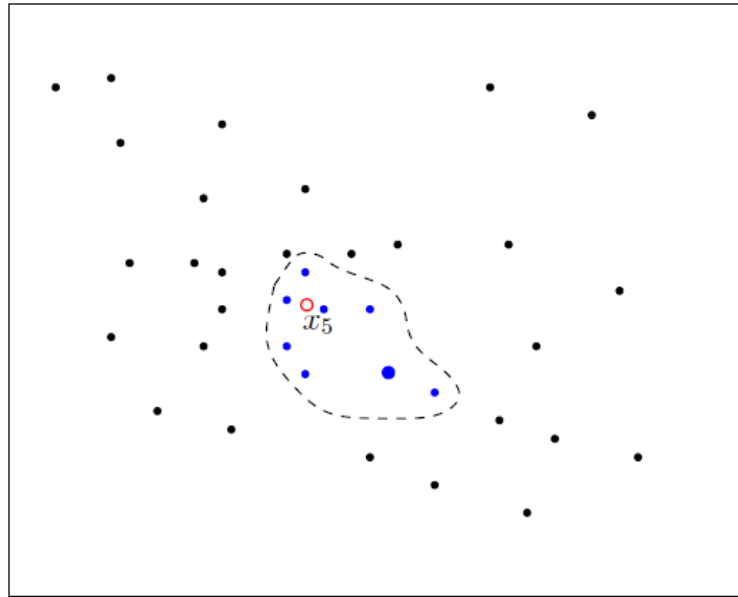


Figure 3.11: premier contours détecté.

n points 1 semence R rayon de la fenêtre $x_{i+1} = \text{signifie } \{ \text{voisine } (x_i) \}$

Cluster center

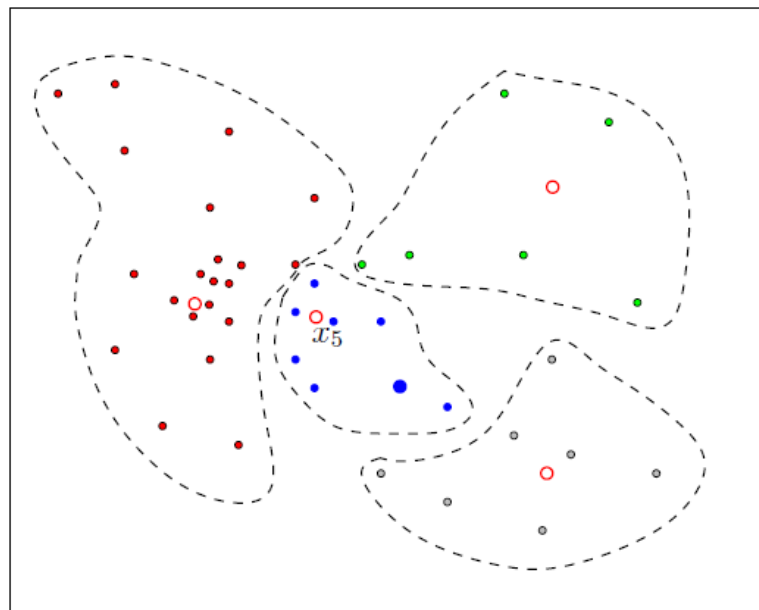


Figure 3.12 : ensemble des contours détectés.

II.3.3. La construction de Mean shift

Un outil pour trouver des modes dans un ensemble d'échantillons de données, manifestant une sous-jacente fonction de densité de probabilité (probability density function) (PDF).

L'espace des caractéristiques représente l'espace des points sur lesquels nous allons appliquer l'algorithme Mean-Shift.

Dans notre cas, cet espace devra contenir l'histogramme RVB de l'image bruitée, c'est donc un espace à 3 dimensions :

$R = \{\text{ensemble des points } P, \text{ avec } P = (\text{rouge, vert, bleu})\}.$

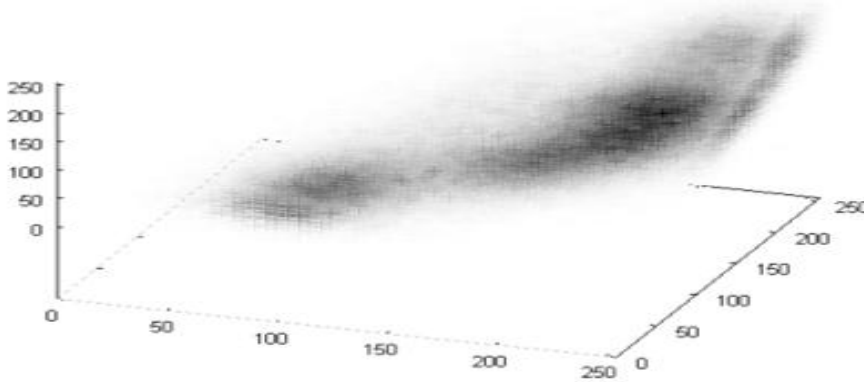


Figure 3.13 : Espace caractérisé

Estimation de la densité non paramétrique Les points de données sont échantillonnés à partir d'un fichier PDF sous-jacente

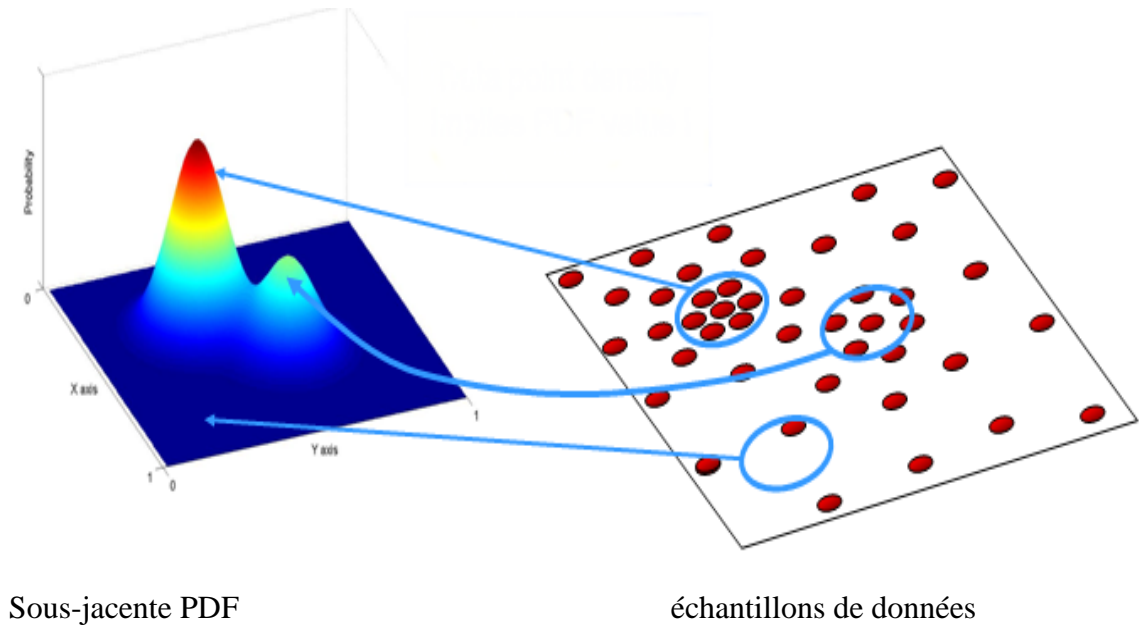


Figure 3.14 : échantillonnage des données

Ci-dessous (Figure 3.15) nous pouvons voir deux cas simples où l'estimation de la densité de probabilité par une gaussienne s'avère proche ou éloignée des données.

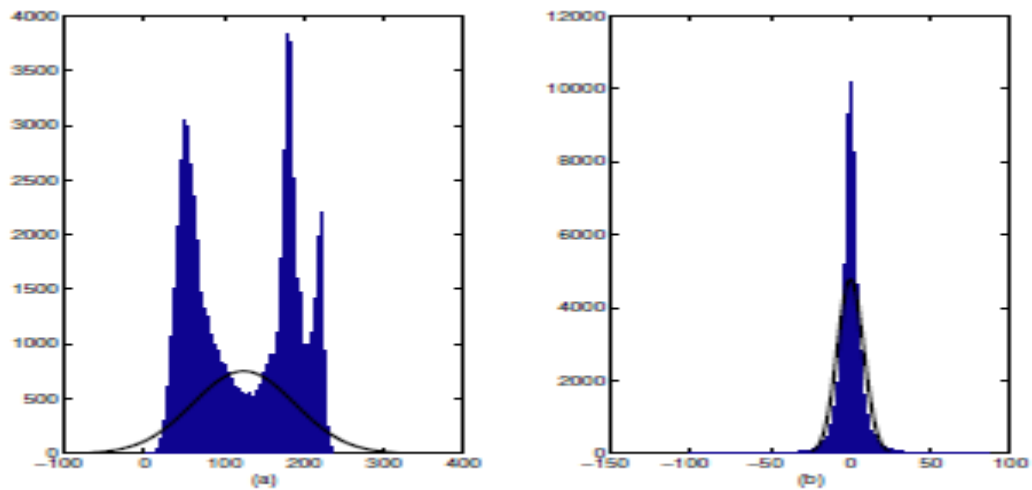


Figure 3.15: Approximation d'une distribution de probabilité par une gaussienne

Pour estimer la densité de probabilité d'un ensemble discret de N points, nous allons approcher cette distribution discrète par une fonction continue par Estimation de la densité de probabilité à l'aide de fenêtres de Parzen de même écart type

. (Figure 2.22)

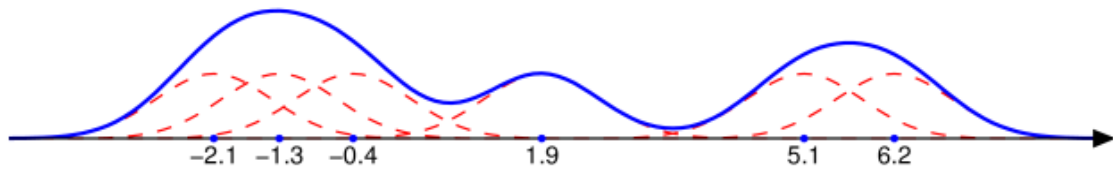


Figure 3.16 : estimation de la densité à l'aide de la fenêtre de parzen

Dans la figure (3.16) nous pouvons voir comment on affecte à chaque point de cet ensemble une fonction, ici une gaussienne dessinée en rouge pointillé, appelée fenêtre de Parzen ou noyau de Parzen. La somme de ces fenêtres va nous donner une approximation continue de la densité de probabilité de la distribution (en bleu).

Dans la forme la plus simple de l'estimation de densité par noyaux de Parzen, chaque fenêtre est une gaussienne de même écart type h.

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Le rôle du noyau est de définir la taille du voisinage dans laquelle doit être prise en compte l'estimation de la densité de probabilité

Il est facile de s'apercevoir, en observant la figure 3.16, que si deux points sont situés à une distance très supérieure à h, leurs influences mutuelles respectives par le biais des estimateurs de Parzen sont négligeables (par exemple les points 1.9 et 6.2) lors de l'estimation de densité.

La formule de l'estimation de la densité par noyaux de Parzen en un point x s'écrit comme une somme de fenêtres de Parzen. Ces fenêtres de Parzen prennent en argument la distance du point x considéré au reste des points de la distribution.

Soit $\{x_i\}$, $x_i \in R^d$, $i = 1 \dots n$, soit un ensemble de points dans l'espace euclidien. Le noyau de la densité est défini comme suit :

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n k\left(\frac{x-x_i}{h}\right)$$

Où h (appelé le paramètre de bande passante) définit le rayon du noyau.

Largeur de bande passante: la spécification d'un paramètre d'échelle. Difficile de trouver l'échelle mondiale optimale.

Le gradient de densité de l'estimateur est :

$$\nabla f(x) = \frac{2ck,d}{nh^{d+2}} \left[\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \right] \left[\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|\right)} - x \right]$$

Où $g(s) = -k'(s)$.

Le premier terme est proportionnel à l'estimation de la densité en x calculées avec le noyau :

$$G(x) = c_{k,d} g(\|x\|^2)$$

Et le second terme :

$$m_h(x) = \left[\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|\right)} - x \right]$$

Est le décalage moyen. Le vecteur mean shift pointe toujours vers la direction de l'augmentation maximale de la densité. La procédure mean shift, obtenue par successions :

- calcul de la moyenne de décalage vectoriel $m_h(x^t)$.
- translation de la fenêtre $x^{t+1} = x^t + m_h(x^t)$
- Itérer les étapes 1. et 2. jusqu'à convergence c.-à-d. $\nabla f(x) = 0$

L'élément le plus coûteux en calcul de la procédure de déplacement moyenne correspond à identifier les voisins d'un point dans l'espace (tel que défini par le noyau et de sa bande passante), ce problème est connu que la gamme multidimensionnelle recherche dans la littérature géométrie algorithmique.

Le programme a été développé en C++. Ce choix a été motivé par le fait que le logiciel EDISON offre une bibliothèque d'outils de segmentation d'images et, surtout, qu'il est « open-source », c'est-à-dire que le code est en accès libre et qu'il peut, au besoin, être modifié.

Ce logiciel produit des résultats d'une qualité exceptionnelle, et dans un temps très raisonnable (entre deux et dix secondes). Cependant, ce logiciel ne fait pas de la segmentation adaptative. Nous sommes alors essayés de contourner le problème, et nous avons été amenés à nous pencher sur le problème de l'échelle h . Un « Mean Shift » appliqué à basse échelle ($h = 3$) conserve les détails, mais produit de la sur-segmentation, comme le montre la figure suivante.

Exemple

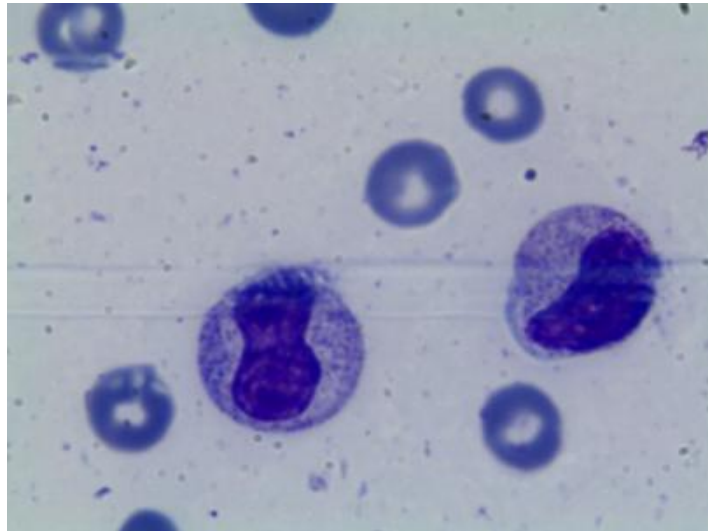


Figure 3.17: Photos *prise* de sang (image original)

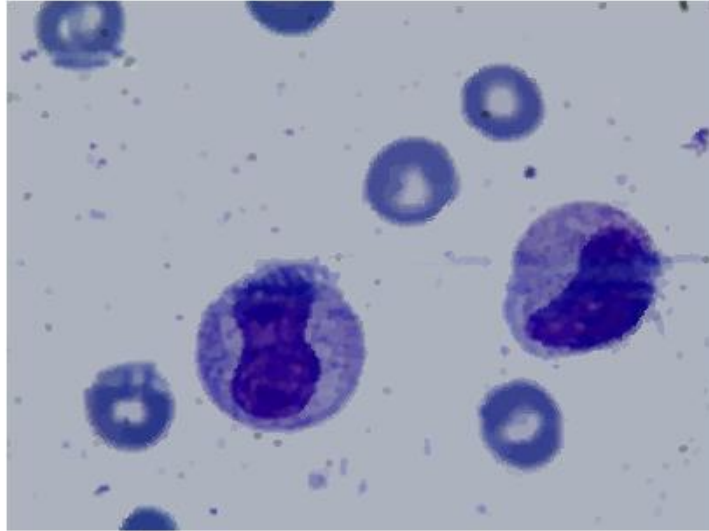


Figure 3.18 : Photos prise de sang (segmentation EDISON h=3)

Une segmentation « Mean Shift » avec un paramètre h unique d'échelle plus élevé permet de faire de la bonne segmentation au sens où les petites variations de couleurs sont éliminées. Cependant tous les détails sur les bords ne sont pas bien préservés et le fond s'entremêle un peu à l'objet

Comme le choix des paramètres est compliqué, nous avons décidé de travailler avec un seul paramètre en prenant $h_r = h_s = h$ ou :

$$k(x) = c k_s \left[\left\| \frac{x^s}{h_s} \right\| \right] k_r \left[\left\| \frac{x^r}{h_r} \right\| \right]$$

- h_s : paramètre de résolution spatiale: Affecte le lissage, la connectivité des segments choisis en fonction de la taille de l'image et les objets.
- h_r : paramètre de résolution couleur: affecte le nombre de segments qui devraient être maintenus bas si le contraste est faible.

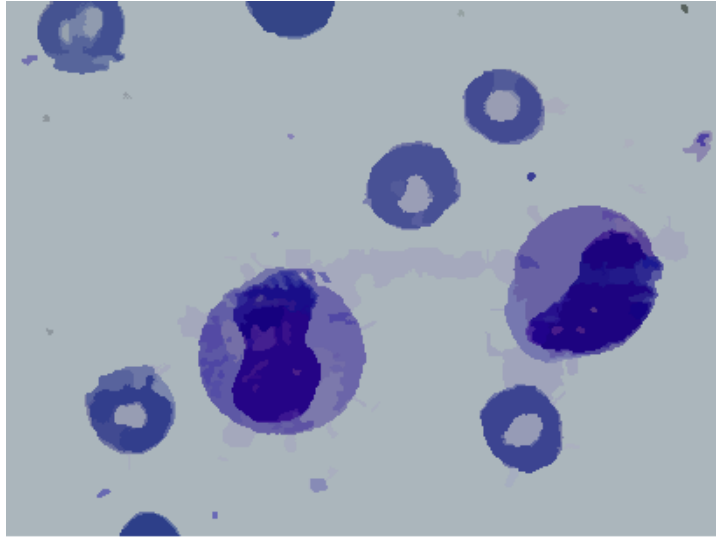


Figure 3.19 : Photos *prise* de sang (segmentation EDISON $h=h_s=h_r=8$)

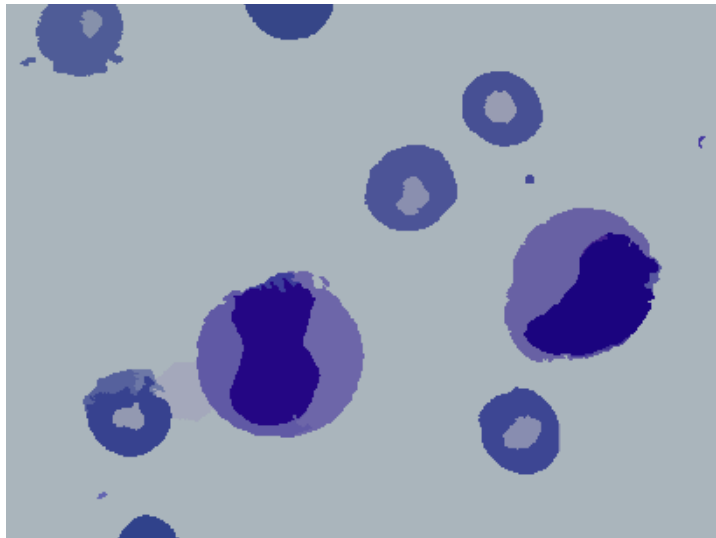


Figure 3.20 : Photos *prise* de sang (segmentation EDISON $h=h_s=h_r=13$)

Il est intéressant d'observer que le fond envahit l'objet de façon très brusque à mesure que l'on augmente le paramètre d'échelle linéairement. L'objet collapse brusquement.

Pourtant, la segmentation la plus intéressante à une échelle h élevée, (même si comme on l'a dit celle-ci elle ne préserve pas les détails sur les contours), est celle qui précède l'effondrement de l'objet (Figure 3.19).



Figure 3.21: Photos *prise* de sang (segmentation EDISON $h=h_s=h_r=30$)

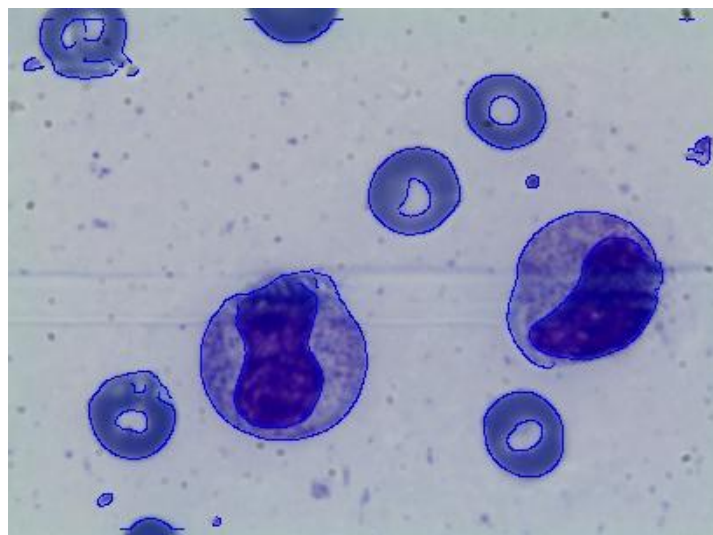


Figure 3.22 : Photos prise de sang (détection des contours EDISON).

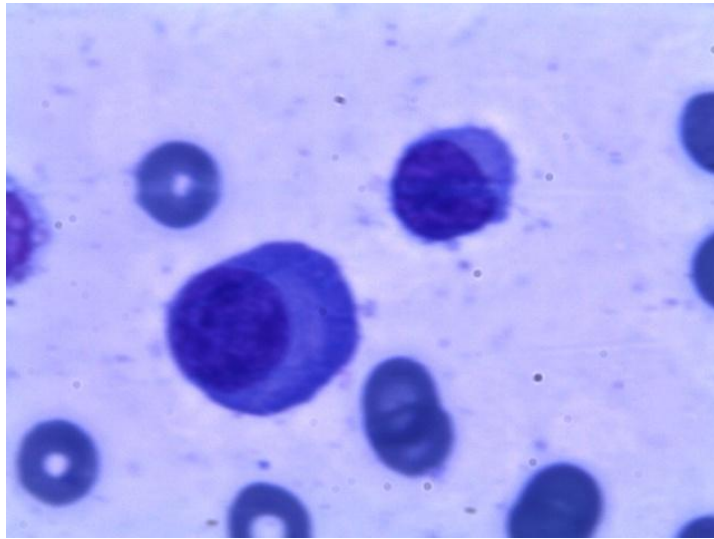


Figure 3.23 : Photos prise de sang (image original)

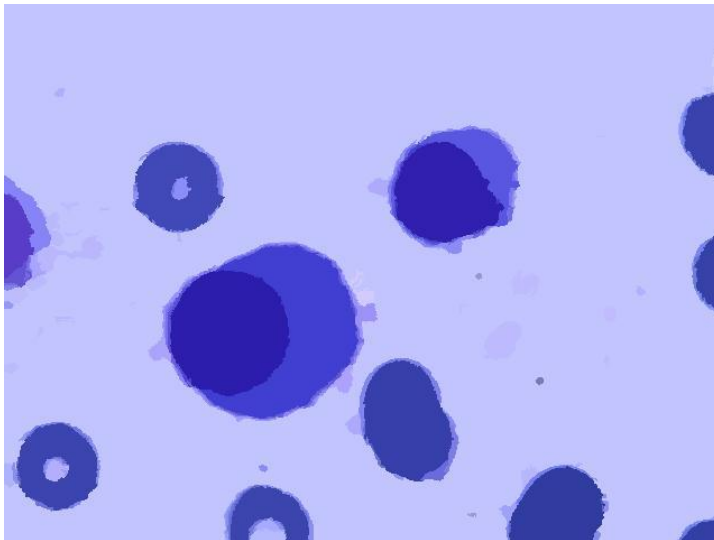


Figure 3.24 : Photos prise de sang (segmentation EDISON $h=h_s=h_r=8$)

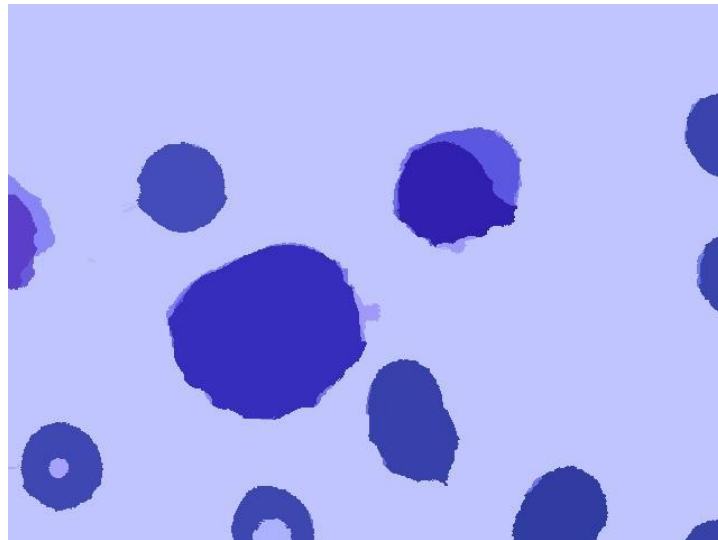


Figure 3.25 : Photos *prise* de sang (segmentation EDISON $h=h_s=h_r=13$)

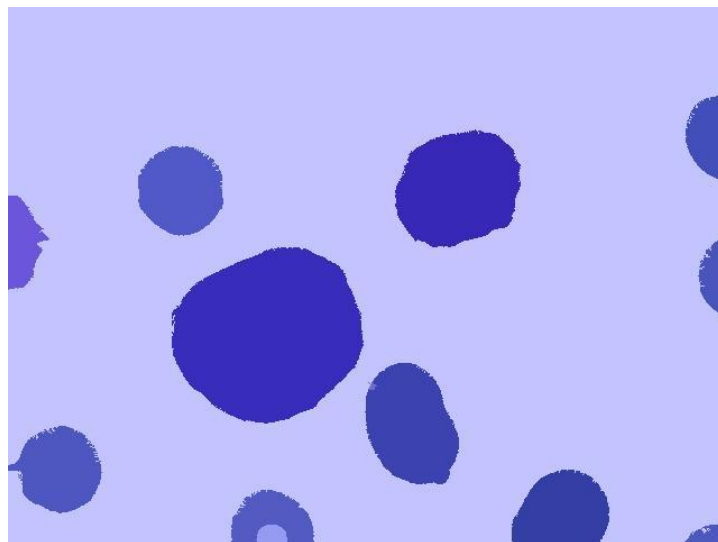


Figure 3.26 : Photos *prise* de sang (segmentation EDISON $h=h_s=h_r=30$)

En plus d'un filtrage « Mean Shift », le logiciel EDISON, une fois la convergence des points atteinte, procède à fusionner certains modes, pour éviter la sur-segmentation, et atteindre des résultats visuellement acceptables (Figure 3.19, Figure 3.26). Le logiciel

fait alors largement appel à des graphes d'adjacence et à de la contraction de graphes en utilisant un algorithme d'étiquetage d'éléments connexes dit de ' Union-Find'.

III. Méthode classification

III.1. Définition

On désigne par la classification d'objet (ou détection d'objet) une méthode permettant de détecter la présence d'une instance (reconnaissance d'objet) ou d'une classe d'objets dans une image numérique. Ces méthodes font souvent appel à l'apprentissage supervisé et ont des applications dans de multiples domaines tels que l'intelligence artificielle.

III.2. Apprentissage

L'apprentissage est un champ de l'intelligence artificielle consiste à acquérir ou à modifier une représentation d'un environnement de façon à permettre avec celui-ci des interactions efficaces ou de plus en plus efficaces.

Du point de vue des machines, nous dirons qu'une machine apprend dès lors qu'elle change sa structure, son programme ou ses données en fonction de données en entrée ou de réponses à son environnement de sorte à ce que ses performances futures deviennent meilleures. [11]

III.3. Apprentissage automatique

III.3.1 Définition

Issu du domaine de l'intelligence artificielle, l'apprentissage automatique est la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques. Les méthodes d'apprentissage automatique offrent un champ d'applications aussi large que varié par exemple :

- Bioinformatique
- Aide au diagnostic médical.
- Aide à la création de nouveaux médicaments.

- Analyse et prédiction de données biologiques (génomique, protéomique, ...).

De nombreux algorithmes ont été développés pour l'apprentissage automatique. Nous nous contenterons, ici, de citer les algorithmes les plus fréquemment utilisés, à savoir:

- Les arbres de décision
- La méthode des k plus proches voisins
- Les réseaux de neurones
- Les machines à support vectoriel.

III.4. Types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient.

III.4.1. L'apprentissage supervisé

L'apprentissage supervisé cherche à établir un modèle à partir d'une base de données contenant des objets préalablement étiquetés et issus de l'ensemble d'apprentissage. Ces étiquettes déterminent les classes d'appartenance des objets (C_1, C_2, \dots, C_m) et sont attribuées par un expert du domaine étudié.

Soient :

B Un ensemble d'apprentissage,

$O_i = (A_1, 2, \dots)$ des objets caractérisés par des attributs et constituant B .

		Les entrées			les sorties	
		A_1	A_2	...	A_n	\hat{Y}
Objets		2.3	On		3.4	C_4
		1.2	Off		0.3	C_1
		3.9	Off		2.8	C_m
	

$$\hat{Y} = f(A_1, A_2, \dots, A_n) \quad \text{Modèle}$$

Figure 3.27 : Apprentissage supervisé

Mathématiquement, le modèle est une fonction de prédiction f prenant, en entrée, les attributs des objets pour donner, en sortie, la classe d'appartenance la plus probable de ces objets. (Voir figure (3.27) : Apprentissage supervisé).

Le modèle permet de prédire la classe à laquelle un nouvel objet appartient sur base de ses attributs. Il permet également de donner des informations sur les relations existantes entre les entrées et la sortie d'un objet.

III.4.2. Apprentissage non-supervisé

Dans le domaine informatique, l'apprentissage non-supervisé (ou classification automatique) (parfois dénommé « clustering ») est une méthode d'apprentissage automatique. Il s'agit pour un logiciel de diviser un groupe hétérogène de données, en sous-groupes de manière à ce que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se retrouvent dans d'autres groupes distincts ; l'objectif étant de permettre une extraction de connaissance organisée à partir de ces données. Cette méthode se distingue de l'apprentissage supervisé par le fait qu'il n'y a pas de sortie a priori. Dans l'apprentissage non-supervisé il y a en entrée un ensemble de données collectées. Ensuite le programme traite ces données comme des variables aléatoires et construit un modèle de « densités jointes » pour cet ensemble de données.

Néanmoins, la classification automatique n'est pas une étape triviale et constitue même un problème épineux car les résultats ne répondent pas toujours aux exigences voulues. Ceci s'explique par le fait que les images biomédicales présentent de nombreuses variations dues :

Au contexte d'acquisition (manque ou variation de luminosité, matériel et protocole utilisés, sources de bruit,...).

Aux corps étudiés : les cellules, par exemple, ont une structure non rigide. Il n'est donc pas toujours aisé de les caractériser selon leur forme. [12]

III.4.3. Apprentissage semi-supervisé

L'apprentissage semi-supervisé est une classe de techniques d'apprentissage automatique qui utilise un ensemble de données étiquetées et non-étiquetées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et

l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées. Il a été démontré que l'utilisation de données non-étiquetées, en combinaison avec des données étiquetées, permet d'améliorer significativement la qualité de l'apprentissage. Un autre intérêt provient du fait que l'étiquetage de données nécessite l'intervention d'un utilisateur humain. Lorsque les jeux de données deviennent très grands, cette opération peut s'avérer fastidieuse. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, revêt un intérêt pratique évident. [13]

III.4.4. L'apprentissage partiellement supervisé

L'apprentissage partiellement supervisé (probabiliste ou non), quand l'étiquetage des données est partiel. C'est le cas quand un modèle énonce qu'une donnée n'appartient pas à une classe A, mais peut-être à une classe B ou C.

III.4.5. Apprentissage par renforcement

L'apprentissage par renforcement fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences, ce qu'il convient de faire en différentes situations, de façon à optimiser une récompense numérique au cours du temps.

Un paradigme classique pour présenter les problèmes d'apprentissage par renforcement consiste à considérer un agent autonome, plongé au sein d'un environnement, et qui doit prendre des décisions en fonction de son état courant. En retour, l'environnement procure à l'agent une récompense, qui peut être positive ou négative.

L'agent cherche, au travers d'expériences itérées, un comportement décisionnel (appelé stratégie ou politique, et qui est une fonction associant à l'état courant l'action à exécuter) optimal, en ce sens qu'il maximise la somme des récompenses au cours du temps. [7]

III.5. Méthodologie utilisée

III.5.1. Introduction

La construction d'un ensemble d'apprentissage par un expert est une tâche difficile et coûteuse. De plus les exemples choisis ne sont pas nécessairement les images les mieux adaptées aux systèmes de recherche. La recherche interactive permet d'exprimer et d'affiner la requête de l'utilisateur. Elle instaure un dialogue entre l'utilisateur et le système en vue de construire un ensemble d'apprentissage avec un nombre limité d'exemple, en fonction des besoins du système.

Une classification ou système de classification est un système organisé et hiérarchisé de catégorisation d' « objets » qui est défini par :

- fournir à l'ordinateur une fonction lui permettant d'associer une catégorie.
- déterminer les variables pertinentes.
- Introduire un type de fonctions de classement.

Il existe différents méthode de classement parmi les quelles on trouve :

- le classifieur linéaire.
- Les arbres de décision.
- Méthode de boosting.

III.5.2. Le classifieur linéaire

En apprentissage automatique, le terme de classifieur linéaire représente une famille d'algorithmes de classement statistique. Le rôle d'un classifieur est de classer dans des groupes (des classes) les échantillons qui ont des propriétés similaires, mesurées sur des observations. Un classifieur linéaire est un type particulier de classifieur, qui calcule la décision par combinaison linéaire des échantillons.

Le classificateur linéaire crée un hyperplan (coupe l'espace en deux illustrer dans la figure 3.2 : Le classificateur linéaire) : d'un côté, ce qui est dans la catégorie, de l'autre côté, ce qui n'y est pas, la modélisation est simpliste mais facile (et rapide) à calculer

par une machine, les coefficients ne sont en revanche pas interprétables « intuitivement » par des humains.

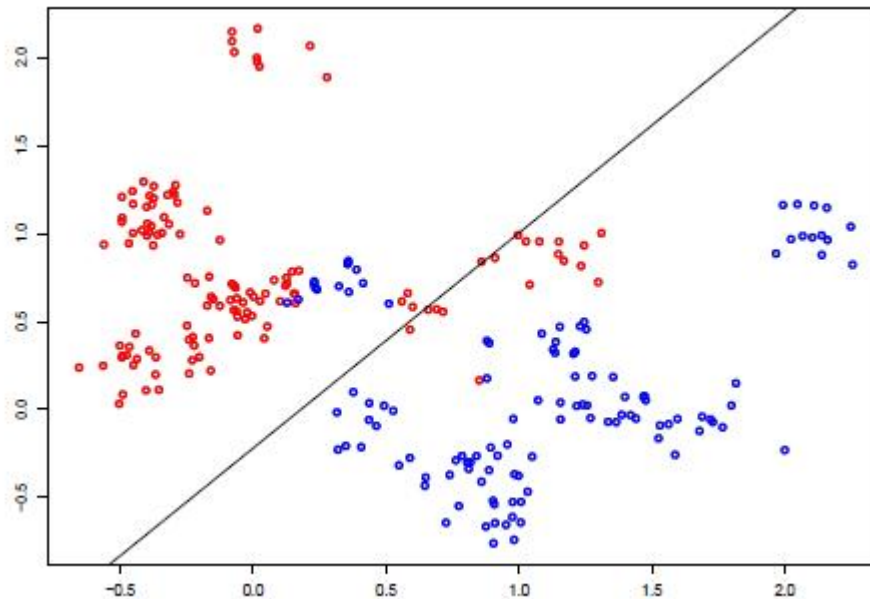


Figure 3.28 : Le classificateur linéaire

Les classifieur linéaires sont souvent employés dans les situations où une faible complexité est souhaitée (par exemple : classification de texte), car ce sont les classifieur les plus simples et donc les plus rapides. [7]

III.5.3. Les arbres de décision

Un arbre de décision est un outil d'aide à la décision qui représente la situation plus ou moins complexe à laquelle on doit faire face sous la forme graphique d'un arbre de façon à faire apparaître l'extrémité de chaque branche les différents résultats possibles en fonction des décisions prises à chaque étape

Cette technique est autant populaire en statistique qu'en apprentissage automatique. Son succès réside en grande partie à ses caractéristiques

- **lisibilité** du modèle de prédiction, l'arbre de décision, fourni. Cette caractéristique est très importante, car le travail de l'analyste consiste aussi à faire comprendre ses résultats afin d'emporter l'adhésion des décideurs.
- **capacité** à sélectionner automatiquement les variables discriminantes dans un fichier de données contenant un très grand nombre de variables potentiellement intéressantes. En ce sens, un arbre de décision constitue une technique exploratoire privilégiée pour appréhender de gros fichiers de données.

Comme introduit précédemment, le but d'un algorithme de classification est de créer un modèle qui, appliqué à une série d'images, permet de les classer le plus précisément possible. Dans le cadre d'un apprentissage supervisé, l'algorithme d'apprentissage peut créer des arbres de décision qui définissent alors des modèles facilement interprétables. Les arbres de décision sont des méthodes de classification qui permettent de séparer les objets selon leurs attributs. [14]

Comme le montre l'exemple ci-dessous (figure 3.29 : Arbre de décision), le principe est de construire des arbres binaires dont les nœuds intérieurs permettent de tester un attribut et où chaque feuille détermine une prédiction de la classe d'appartenance de l'objet.

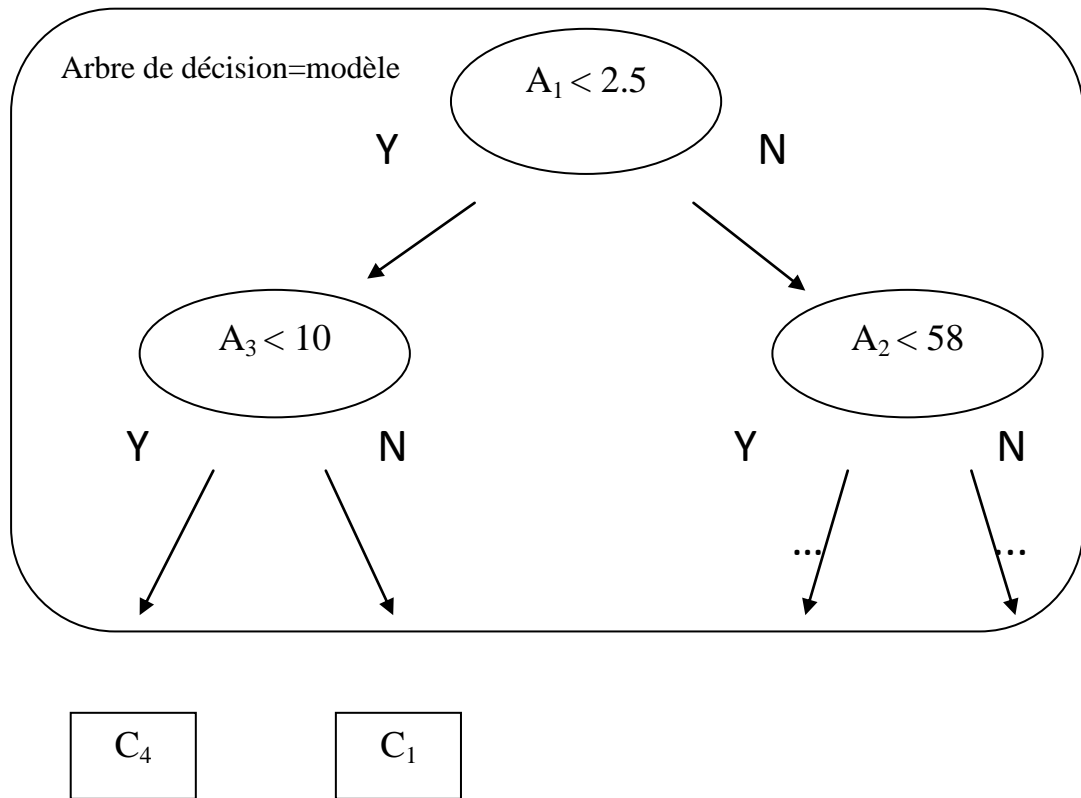


Figure 3.29 : Arbre de décision.

Il s'agit de trouver le test dont le partitionnement est le plus « pur » de sorte à minimiser la profondeur finale de l'arbre généré, l'inconvénient des arbres de décision est de ne pouvoir prendre que des décisions binaires.

III.6. Le Boosting

III.6.1. Définition

Le boosting : est une classe de méthodes d'apprentissage basée sur la sélection de classifieur. Ce sont des techniques utilisées dans de nombreuses activités d'apprentissage, aussi bien pour de la catégorisation et le filtrage de texte, que pour des problèmes linguistiques, de bioinformatique ou de détection et classement d'objet dans des images.

Un classifieur : représente une famille d'algorithmes de classement statistique. Le rôle d'un classifieur est de classer dans des groupes (des classes) les échantillons qui ont des propriétés similaires, mesurées sur des observations. [7]

III.6.2 Principe général

Le boosting repose sur l'observation suivante :

'Faire simple pour faire compliquer'

- il est facile de trouver des règles simples, généralement justes
- il est très difficile de trouver une règle "fortes" qui est vérifiée dans tous les cas.

Dans le contexte de la détection d'objet, le boosting est utilisé pour sélectionner des classifieurs faibles d'objet dans des images. En effet, le résultat de la recherche d'une catégorie d'objets correspond à la classification de la base en deux ensembles d'images:

Positif X_p : les images qui contiennent les objets.

Négatif X_n : celles qui ne les contiennent pas.

$[h(x_p) > h(x_n)] ?$

Les images d'apprentissage pré-annotées (positivement ou négativement) permettent de sélectionner des classifieurs faibles h_t dans un ensemble W . A chaque itération t de sélection, un classifieur est choisi en fonction d'un critère reposant sur la capacité du nouveau classifieur faible à classer correctement les images qui ont été mal classées par les classifieurs faibles précédemment choisis. Le choix de ces classifieurs se fait itérativement en utilisant l'ensemble de la base.

Le classifieur fort final H effectue la somme des résultats de chacun des classifieurs faibles sélectionnés en les pondérant en fonction de la qualité de leur classement. H est alors meilleur que chacun des h_t pris séparément. Ce processus d'apprentissage peut être formalisé par la production d'un classifieur fort H connaissant un ensemble d'exemples X et un ensemble de classifieurs faibles W

App : $(X, W) \rightarrow H$

Les classifieurs faibles utilisés peuvent être de nature quelconque, il faut juste qu'ils soient légèrement meilleurs que le hasard.

Exemple simple : Détecter des mots au pluriel

Règle faible 1 : les mots se terminant par “s” sont au pluriel. (Majorité des cas)

On se concentre sur les mots pour lesquels la règle 1 ne fonctionne pas

Règle faible 2 : les mots se terminant par “x” sont au pluriel

Règle forte : les mots se terminant par “s” **OU** “x” sont au pluriel.

Le boosting a été testé sur des domaines d’application variés. En général, l’utilisation du boosting a pour résultat d’améliorer souvent sensiblement les performances en apprentissage.

Ceci est-il l’objet ($H(x) = 1$) ? (3.10)

Les avantages du boosting sont qu’il s’agit d’une méthode facile à programmer et aisée d’emploi. Elle ne nécessite pas de connaissance a priori sur l’algorithme d’apprentissage « faible » utilisé, et elle peut s’appliquer de fait à n’importe quel algorithme d’apprentissage faible. Les seuls paramètres à régler sont la taille de l’ensemble d’apprentissage m et le nombre total d’étapes T , qui peuvent être fixés par l’utilisation d’un ensemble de validation.

Exemple de mot au pluriel :

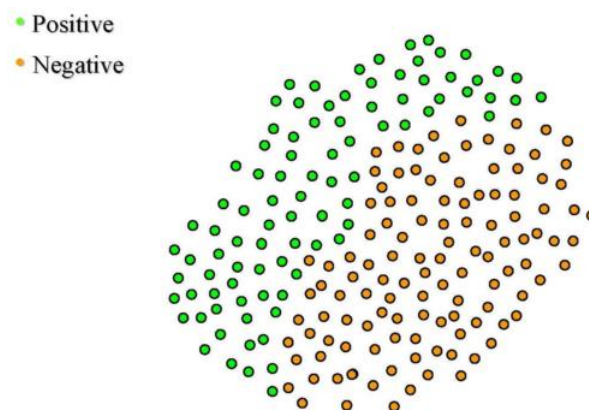


Figure 3.30 : Ensemble à classifier

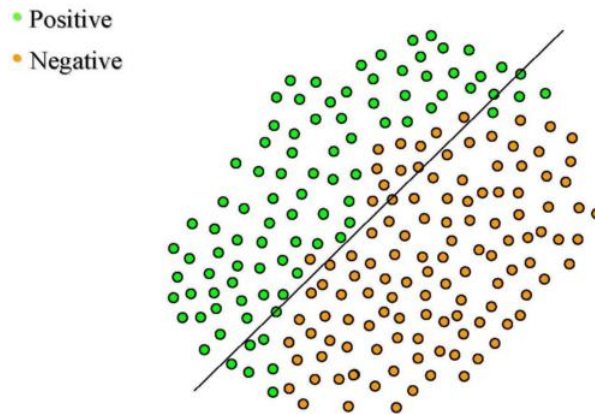


Figure 3.31 : Itération 01 : Classifieur faible

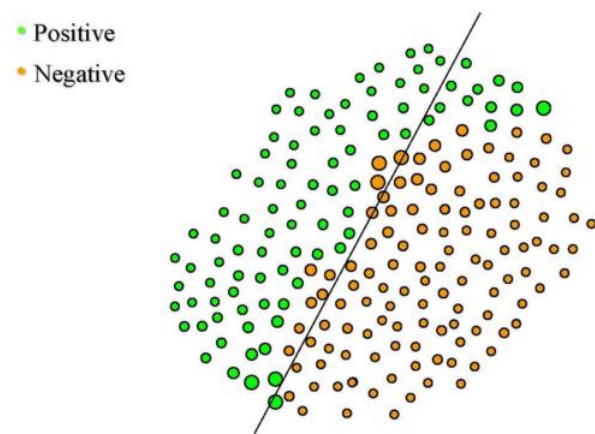


Figure 3.32 : Itération 02 : Classifieur fort

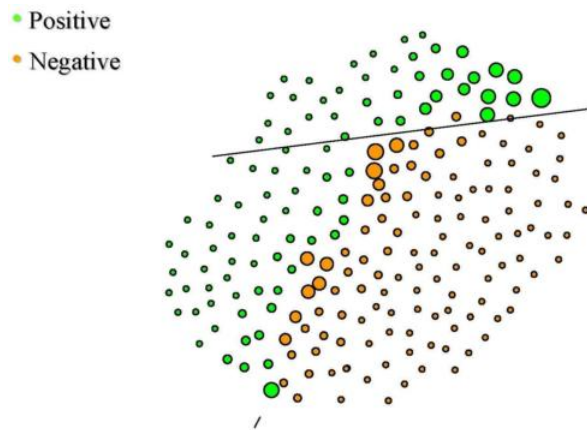


Figure 3.33 : Itération 03 : Classifieur plus fort

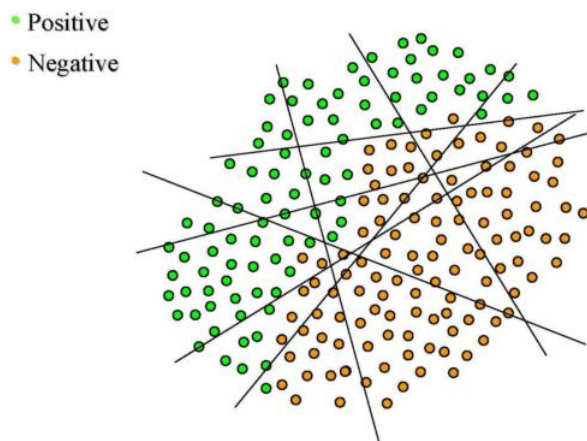


Figure 3.34 : Itération 04 : Classifieur plus fort

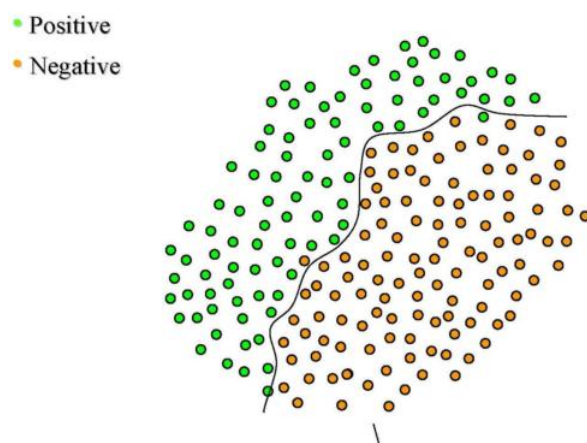


Figure 3.35 : Itération 05 : Classifieur fort final [24]

IV. Conclusion

La phase de segmentation est une partie essentielle dans mon travail pour que je puisse continuer j'utilise des méthodologies de classification d'objet tels que d'appliquer le principe de boosting par ce qu'il permet de renforcer le classifieur grâce à ses itérations, ce qui rendra les résultats de catégorisation des globules plus fiable.

I. Introduction

Le travail effectué est centré sur le développement d'un système de reconnaissance des cellules sanguines et plus exactement le comptage de ces dernières par la méthode boosting qui présente de nombreux avantages par rapport aux autres méthodes d'apprentissage (le classifieur linéaire, les arbres de décision) que ce soit en terme de précision de calculs et renforcement de classifieur.

II. Le langage support

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000. En plus de Java, NetBeans 6.7.1 permet également de supporter différents autres langages, comme Python, C, C++, etc. Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développement Kit JDK (version 1.6) est requis pour les développements en Java.

III. Processus de développement

La conception de mon application elle doit passer par plusieurs phases.

La phase de segmentation et détection de contour:

La segmentation est réalisée à l'aide de l'application EDISON¹, son code source est hébergé gratuitement (open source). Le système est construit en utilisant deux modules distincts: un détecteur de contours et un segmenteur mean shift d'image.

Pour la segmentation on charge l'image (image ci-dessus).

¹ EDISON V1.0 (Edge Detection and Image Segmentation) est un outil de vision de bas niveau qui effectue la détection de contours basée sur la segmentation d'images. Il est emballé sous une plateforme d'interface utilisateur graphique indépendante.

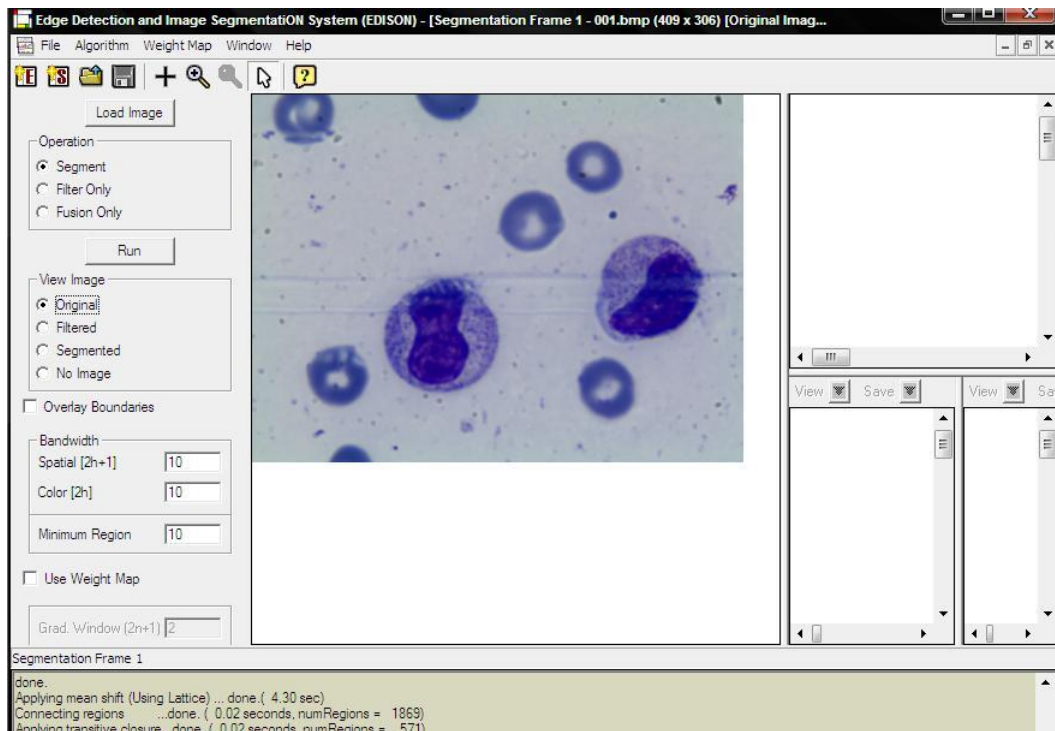


Figure 4.1 : Chargement des images.

Puis on effectue la segmentation sur l'image chargé.

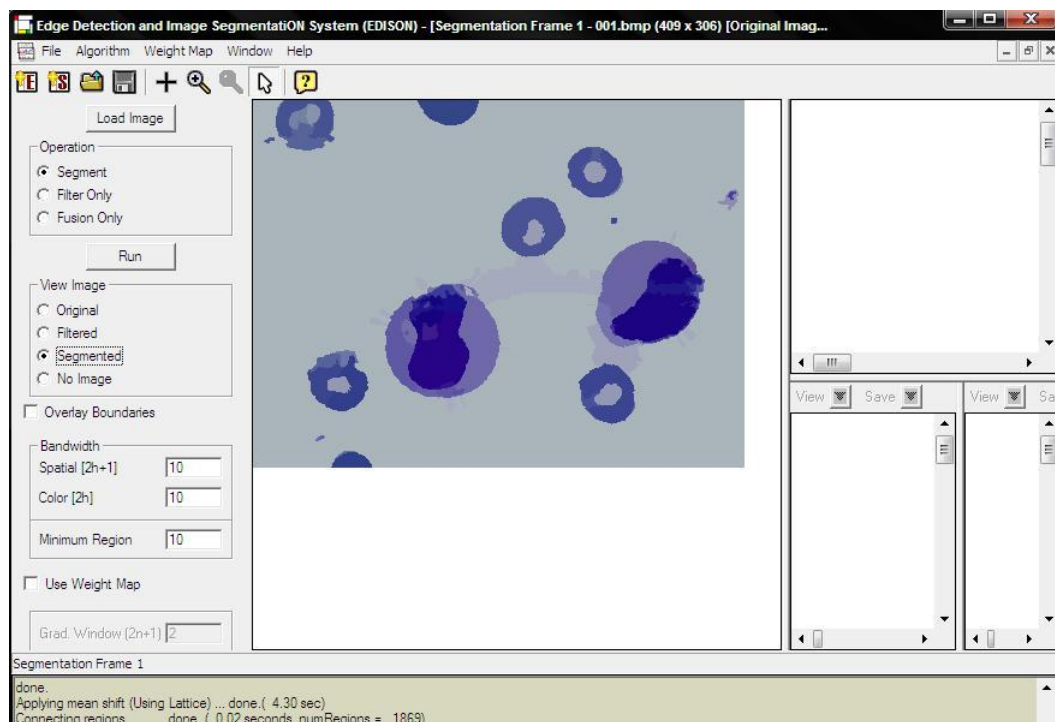


Figure 4.2: Démonstration de la segmentation.

III.1 Détection de contour :

Après la segmentation EDISON offre le choix d'enregistrement de l'image segmentée selon différentes formes (pnm, png, pcx, pgm, jpg), nous avons choisi l'image avec l'extension PNG. Cette dernière va générer 03 images, nous appliquerons à la suite la détection de contours à l'image segmentée.

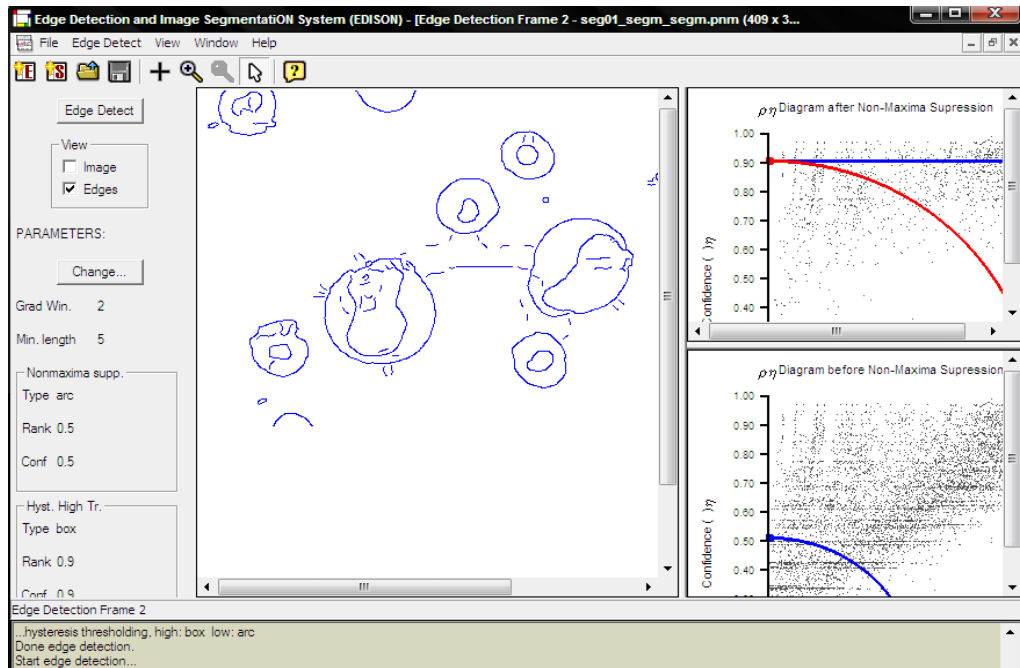


Figure 4.3: Contour détecté.

Une fois la segmentation faite est les contours détectés nous passons à l'étape de reconnaissance des objets.

III.2 La phase de reconnaissance d'objet :

Cette phase elle-même passe par plusieurs étapes, tout d'abord pour le comptage des globules j'ai utilisé l'algorithme boosting que j'ai définie dans le chapitre précédent, pour cela on doit créer une base d'apprentissage en utilisant des annotations pour étiqueter les globules blancs et rouges avec leurs propriétés et leurs valeurs géométriques.

III.3 Création des annotations :

Le but de création des annotations d'organiser et retrouver des objets d'intérêts dans une donnée on crée des classes dans des objets.

Pour avoir un meilleur résultat d'annotation, Les image de la base d'apprentissage choisies sont assez bien segmentées.

J'ai créé les annotations sur le lien suivant comme montre la figure ci-dessous. [26]

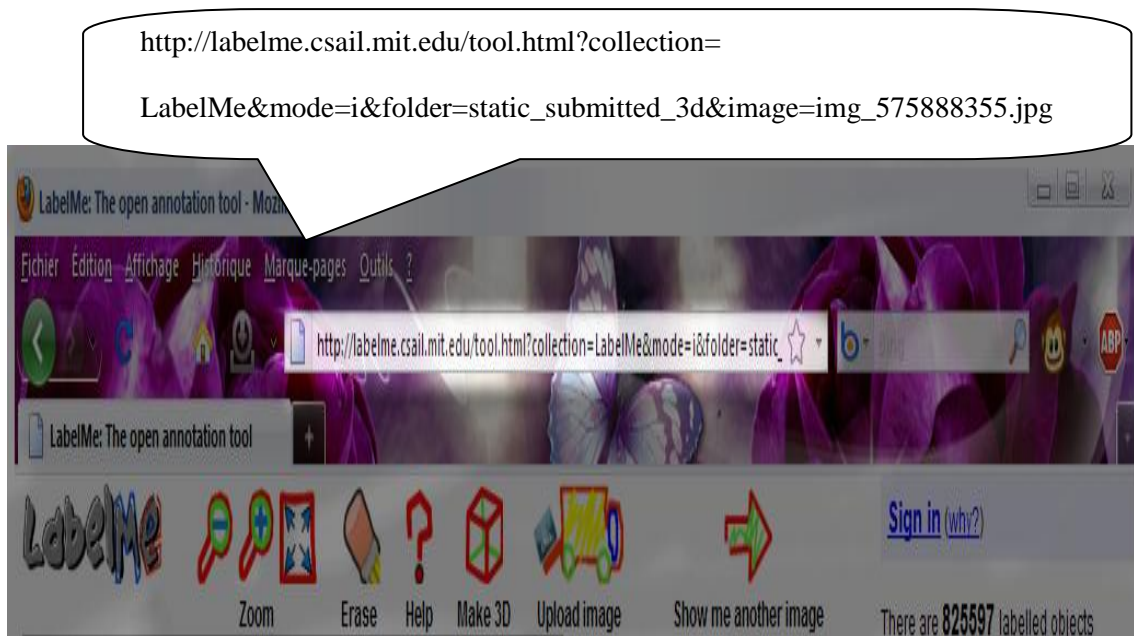


Figure 4.4 : LabelMe annotations.

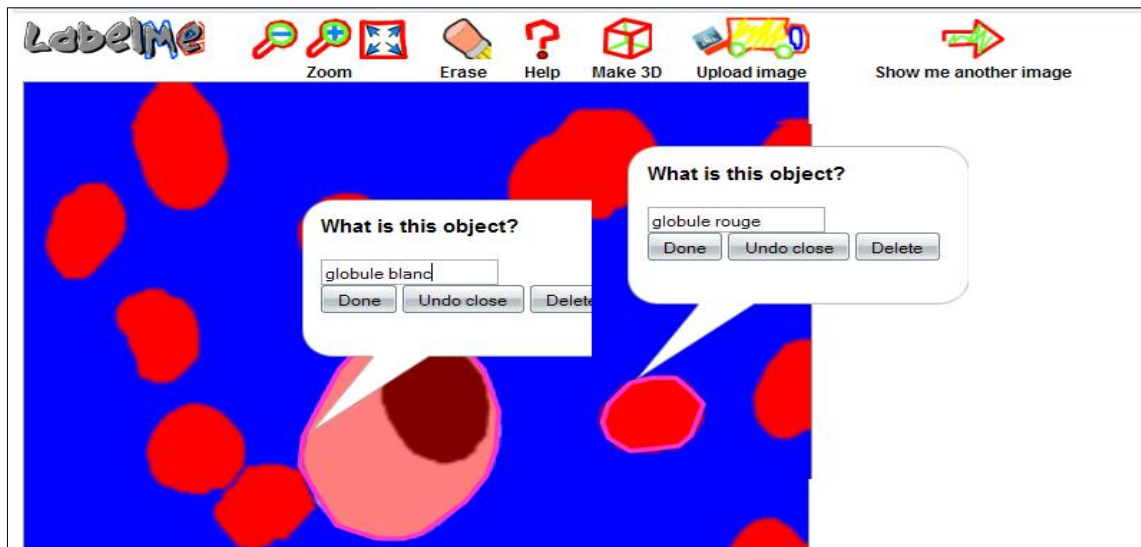


Figure 4.5 : créations des annotations

Le but de LabelMe est de fournir un outil d'annotation en ligne pour construire une grande base de données d'images annotées. Il permet d'extraire ces annotations sous forme d'un fichier XML.

Pour chaque image nous aurons son propre fichier XML qui va contenir les coordonnées (x, y) de réfraction du chaque polygone qui définira les contours des globules blancs ou rouges. (Voir les figures [4.6], [4.7] si dessous).

```

<object>
<name>globule rouge</name>
<deleted>0</deleted>
<verified>0</verified>
<id>0</id>
<polygon>
<username>anonymous</username>
- <pt>
<x>197</x>
<y>114</y>
</pt>
-<pt>
<x>204</x>
<y>112</y>
</pt>
-<pt>
<x>212</x>
<y>112</y>
</pt>
-<pt>
<x>217</x>
<y>117</y>
</pt>

```

Figure 4.6: fichier XML des annotations
pour un globule rouge.

```

<object>
<name>globule blanc</name>
<deleted>0</deleted>
<verified>0</verified>
<id>1</id>
<polygon>
<username>anonymous</username>
-<pt>
<x>91</x>
<y>136</y>
</pt>
-<pt>
<x>97</x>
<y>123</y>
</pt>
-<pt>
<x>103</x>
<y>110</y>
</pt>
-<pt>
<x>111</x>
<y>100</y>
</pt>

```

Figure 4.7 : fichier XML des annotations
pour un globule blanc.

III.4 Reconnaissance des globules

Dans mon projet la reconnaissance des globules se fera en comparant le résultat de la détection de contours des images queregistre le fichier.txt qui contient les coordonnées (x, y) de chaque régions avec les annotations.

Pour pouvoir faire cette comparaison on divisé les annotations selon leur nature (globule blanc, rouge) et ensuite extraire les coordonnées (x,y) des points de l'image de la base d'apprentissage.

La comparaison des régions et les différents annotations commence par choisir le minimum et le maximum des x et des y entre eux pour créer un nouveau repère de comparaison. Comme montre la figure 4.10 : Repaire de comparaison.

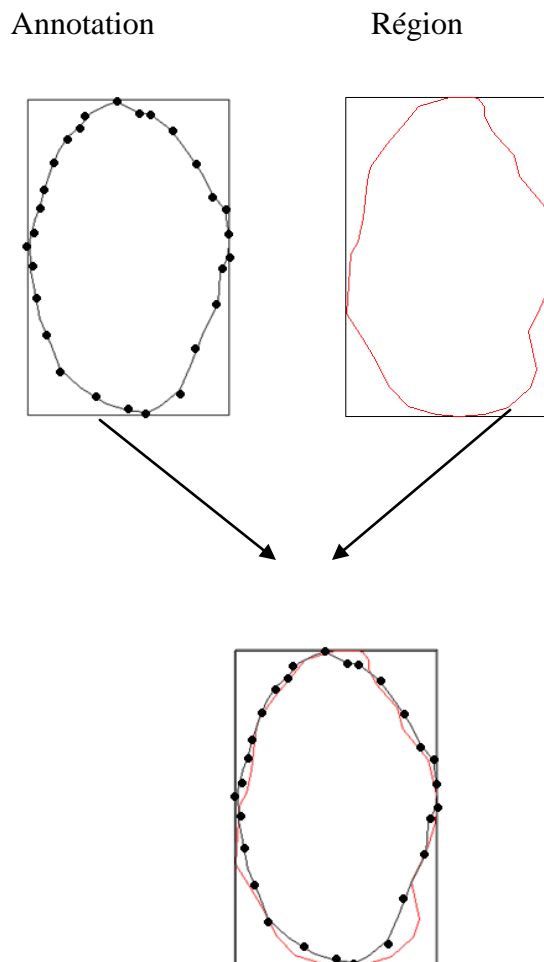


Figure 4.8: Repaire de comparaison

III.5 Classification

Le boosting est un domaine de l'apprentissage automatique. Son principe est issu de la combinaison de classifieurs (appelés également hypothèses). Par itérations successives, la connaissance d'un classifieur faible - weak classifier - est ajoutée au classifieur final - strong classifié.

Algorithme boosting :

Il ya plusieurs algorithmes de boosting celle choisie dans mon application l'algorithme Adaboost (Adaptive boosting). Il s'appuie sur :

- Choix de l'algorithme, selon un critère, le meilleur du classifieur faible par rapport aux pondérations des exemples
- Les poids $D_t(i)$ des exemples sont mis à jour à chaque étape en fonction des poids précédent et des résultats du classifieur sélectionnés.
- Au départ, tous les exemples ont la même importance.

Algorithme AdaBoost:

Le principe de l'algorithme est de définir à chacune de ses étapes $1 \leq t \leq T$, une nouvelle distribution D_t de probabilités a priori sur les exemples d'apprentissages en fonction des résultats de l'algorithme à l'étape précédente. Le poids à l'étape t d'un exemple (x_i, y_i) d'indice i est noté $D_t(i)$. Initialement, tous les exemples ont un poids identique, puis à chaque étape, les poids des exemples mal classés par l'apprenant sont augmentés, forçant ainsi l'apprenant à se concentrer sur les exemples difficiles de l'échantillon d'apprentissage.

Soit un ensemble d'apprentissages $S = \{X_i, Y_i\}$

Telle que $i=1 \dots N$, $X_i \in X$ et $Y_i \in \{-1, 1\}$ label de classe

Initialisation : $t=1$, $D_1(i) = 1/n$ et $E=1$

While $E > \zeta$ **do**

Tirer un échantillon d'apprentissage S_t

Apprendre une règle de classification h_t sur S_t

Soit ζ_t l'erreur apparente de h_t sur S_t . Calculer $\alpha \leftarrow \frac{1}{2} \ln \frac{1-\zeta_t}{\zeta_t}$

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha y(i) h(t) X(i))}{\sum_i D_{t+1}(i)} \quad \text{si } h_t(x_i) = u_i \text{ (bien classé par } h_t)$$

$$D_{t+1}(i) = \frac{D_t(i) \exp(+\alpha y(i) h(t) X(i))}{\sum_i D_{t+1}(i)} \quad \text{si } h_t(x_i) \neq u_i \text{ (mal classé par } h_t) \quad (E = \sum_{H(X_i) \neq y_i} \frac{1}{N}).$$

t++

Fourni en résultat l'hypothèse finale : $H(x) = \text{sign} \sum_t \alpha_t h_t(X)$

Sortie : Un classifieur fort C constitué d'un ensemble de classifieur faibles

Dans mon application :

Construction de la base d'apprentissage qui contient l'ensemble de globule rouge et blanc.

Méthode qui calcule le nombre de globule **nbr_globule(string t[] , int nbr)**

```

Déclaration
int nb=0 , boolean trv=false, string tb[1000],
début
pour int i=0 jusqu'à i<nbr (nbr parametre d'entrée)
    { pour j=0 jusqu'à j<nb
        { si tb[j]=tb[i][0]
            Faire trv = true
        }
        j++}
    Si trv=false faire tb[nb]=tb[i][0]
    nb++ }
retourne nb ;
fin
    
```

Figure 4.9: Méthode pour le calcul de nombre de globule

Méthode qui calcule la similarité entre les valeurs obtenus par la détection de contour et les valeurs obtenus par l'annotation est savoir si c'est un globule blanc ou globule rouge

sémilarite (int nbr, GridLayout bd, int choix)

GridLayout bd :un pointeur de genre tableau pour accéder au cellule voulu .

int choix : un entier qui prend deux valeurs 0 pour globule rouge et 1 pour globule blanc.

La méthode **polygone_gb(Image I, int v)** qui prend en paramètre une image est un entier.

La méthode **separation_bdd()** qui prend deux paramètre un entier int b est un GridLayout pour pointer sur cellule GridLayout Bd.

Le principe de cette méthode :

Je fais un test si c'est un globule blanc et globule rouge j'incrément le nombre de globule puis j'effectue un notre teste pour charger les globules par exemple si le nombre de globule blanc est supérieur a 0 (>0) fait le chargement de globule blanc

Code source :

```
Déclaration :  
  
// appel a la methode AdaBoost()  
  
//charge de la bibiothèque AdaBoost  
  
void AdaBoost(int m, int T, double epsilon, int globule)  
  
//m : entier e {1,2,...}  
  
//T : entier pour l'incrémentation les étapes de classifieur  
  
//epsilon :plus petit valeur pour le calcule de l'erreur  
  
//globule : entier qui prendre deux valeur 0 pour globule rouge et 1 pour blanc  
  
    {  
  
        double[] D;  
  
        //On initialise la distribution des exemples par  $D(i)=1/m, i=1, \dots, m$ . (4.1)  
  
        D = new double[m];  
  
        for (int i = 0; i < m; i++)  
  
            D[i] = (double)(1 / m);  
  
        int t = 1;  
  
        double E = 1;  
  
        int x;  
  
        int y;
```

Figure 4.10: algorithme Adaboost en java

```

//tanque le nombre d'itération t est < T et E est supérieur a petite valeur epsilon faire
    while ((t <= T) && (E>epsilon)) {
        int M = 0;
        while (M < m) {
            if (globule == 0) //pour les globules rouge
{
separation_bdd(M,Form2->PT); //methode qui fait la separation entre globule blanc et rouge
            if (chx() == true) // si le choix de globule et vrai
                {
                    for (int i = 1;i <= nbr_globule(BDGR, NGR); i++)
                        polygone_gr(lmg2,i) ; //methode qui charge les globule rouge
                }
            System.out.println("Global Rouge"); //affichage
                //methode qui calcule la similarité
                SEMELARITE(int nbr, GridLayout bd, int choix);  }
// la même chose pour globule blanc
//en effectue t++ itération jusqu'a aboutir le classifieur fort

```

Figure 4.10: algorithme Adaboost en java

IV. Interface de l'application

Au lancement de l'application, la fenêtre principale s'affiche.



Figure 4.11: fenêtre principale

On clique sur « menu » pour afficher la menu principale de l'application

On clique sur « fichier » un menu déroulant vous permette de choisir la partie que vous voulez « segmentation » ou « classification »

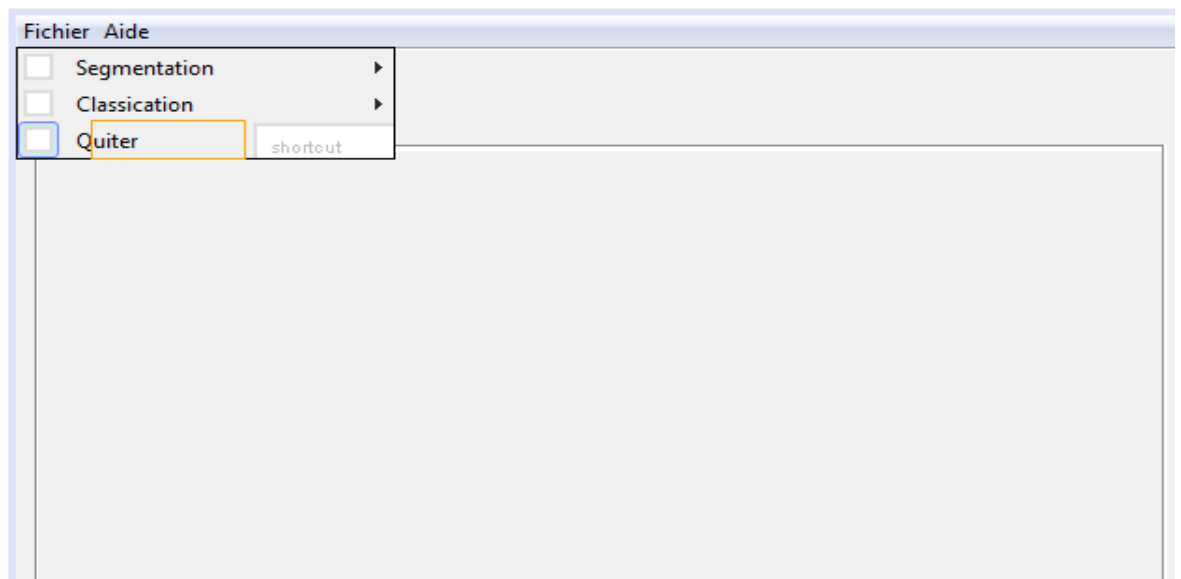


Figure 4.12: la forme de segmentation

On clique sur « segmentation » un autre menu se déroule qui permet de choisir entre faire la segmentation par l'appelle à l'application Edison ou charger une image segmenter

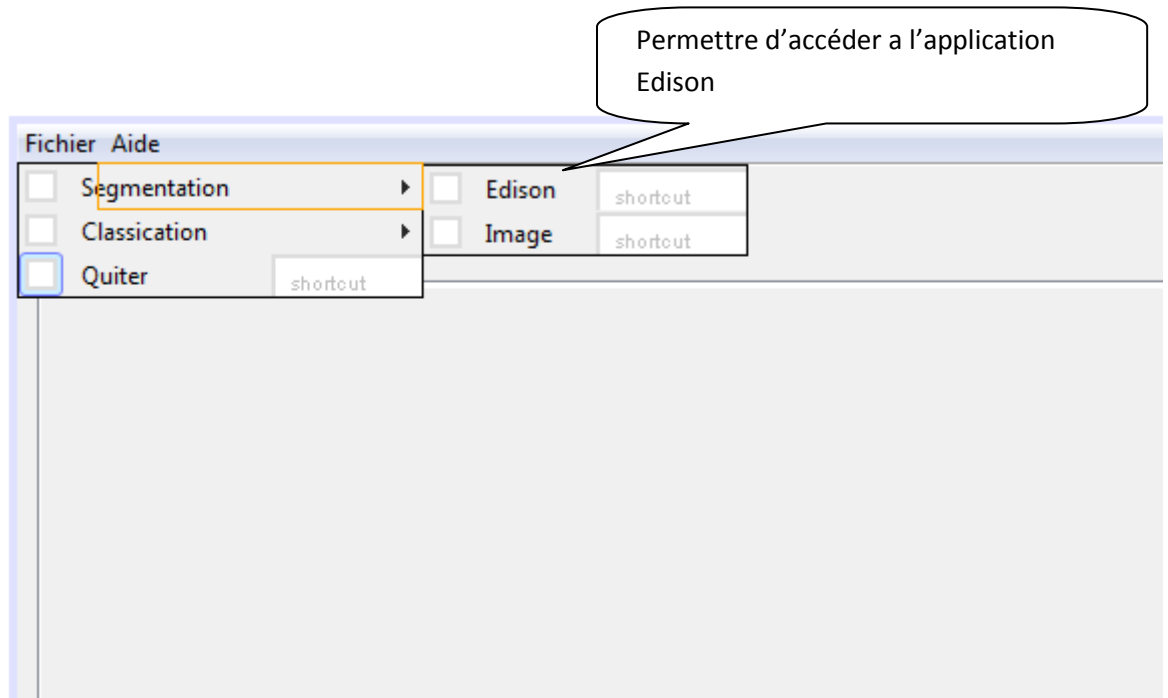


Figure 4.13 : la forme de segmentation

On clique sur « classification » un menu se déroule nous permettre d'accéder au comptage a travers le boosting.

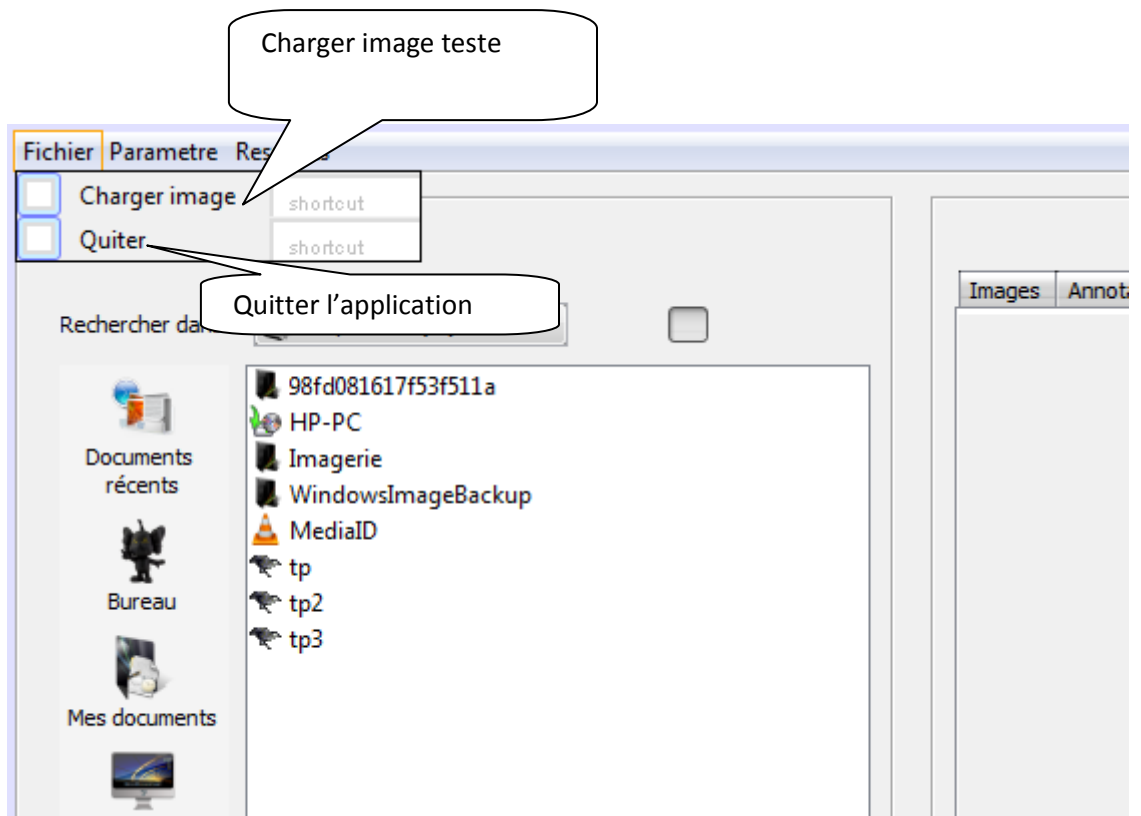


Figure 4.14 : la forme de classification

Le choix de la nature des globules

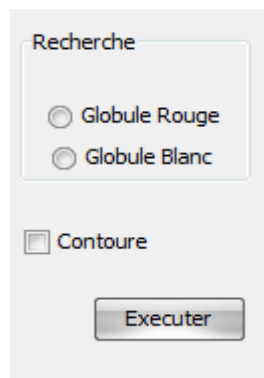


Figure 4.15 : le choix de globule

Code source :

Partie segmentation :

Accéder a l'application Edison on cliquant sur bouton

Charger image segmenter depuis un répertoire on clique sur bouton on utilise un « filechooser »

```
Private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String file = getImageFile(); //appelle a la méthode qui permettre de lire le fichier  
  
    if (file != null) { //on teste si le fichier n'est pas null  
  
        try {  
  
            img = ImageIO.read(new File(file)); } // lire l'image  
  
        catch (IOException ie) { ie.printStackTrace(); }  
  
    }  
  
    //pour l'affichage de l'image dans le panel  
  
    void affiche(){  
  
        Graphics g = null;  
  
        jPanel1.paint(g);  
  
        g.drawImage(img, 0, 0, null);  
  
        jPanel1.show();  
  
    }  
  
}
```

Figure 4.16 : chargement d'image segmenté

Pour l'affichage le tableau des pixels :

```
void chargement() {  
    h = img.getHeight();  
    w = img.getWidth();  
    //dimension de l'image  
  
    TableModel jTable1 = new AbstractTableModel() { //creation d'un model de table  
        public int getColumnCount() { return h; } //nombre de colonne  
        public int getRowCount() { return w; } // nombre de ligne  
        public Object getValueAt(int row, int col) {  
            int[] rgbs = new int[h*w]; //on crée l'espace neccessaire  
            img.getRGB(row, row, w, h, rgbs, col, col); //tableau des pixels  
            return rgbs; }  
    }:  
}
```

Figure 4.17 : Affichage le tableau des pixels

Partie classification :

Cette partie regroupe tous les parties, partie d'affichage d'image annotée, partie d'affichage d'image de détection de contour, la partie choix de globule, partie journal d'événement

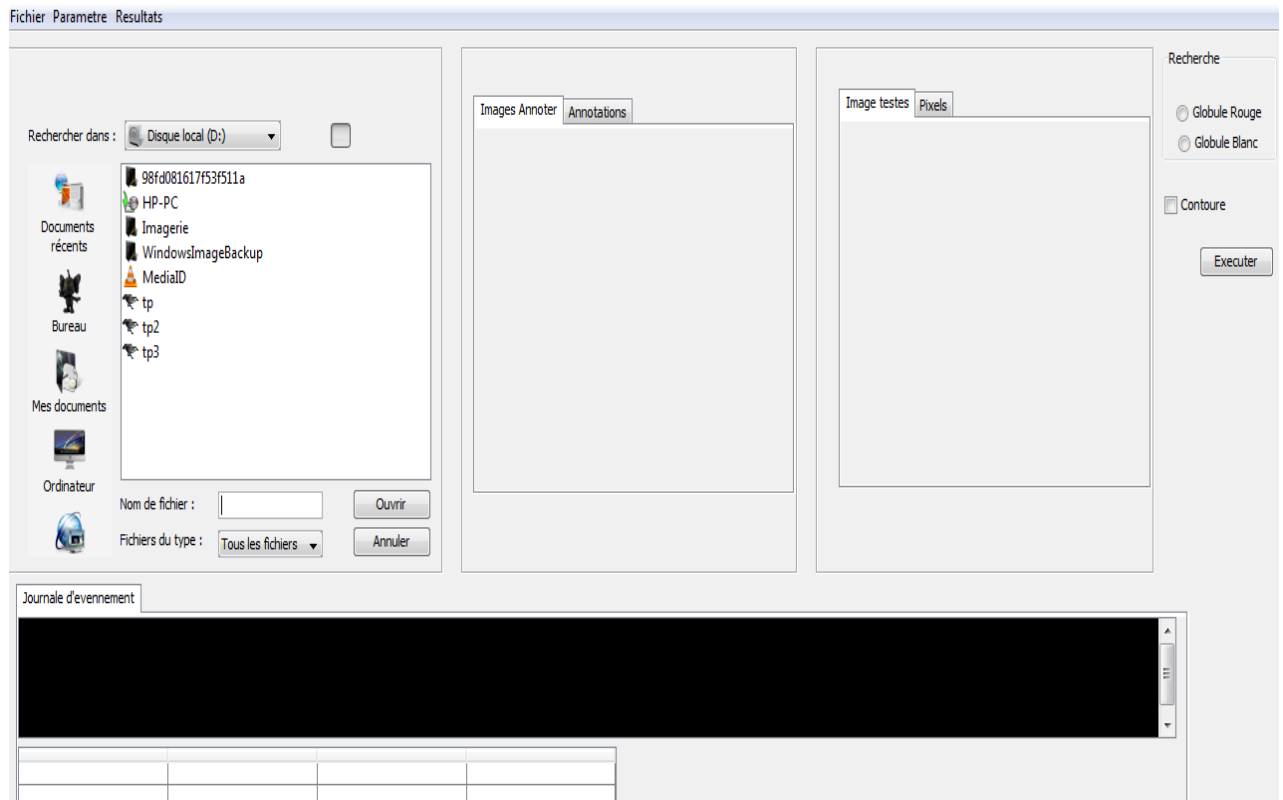


Figure 4.18 : Interface de classification

Chargement d'image annoter .puis accoter j'affiche le fichier xml de l'image que j'ai chargé

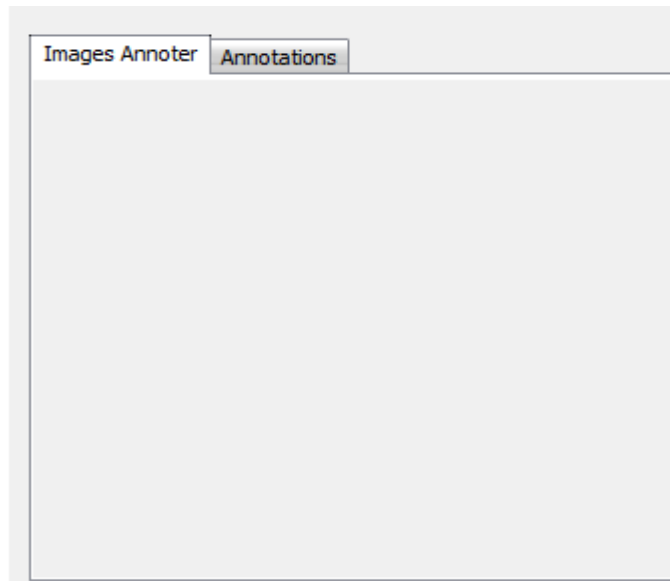


Figure 4.19 : la partie d'annotation

Le code source c'est la même chose que chargement d'image segmenté

Chargement d'image détection de contour son code source c'est la même chose que les autres.

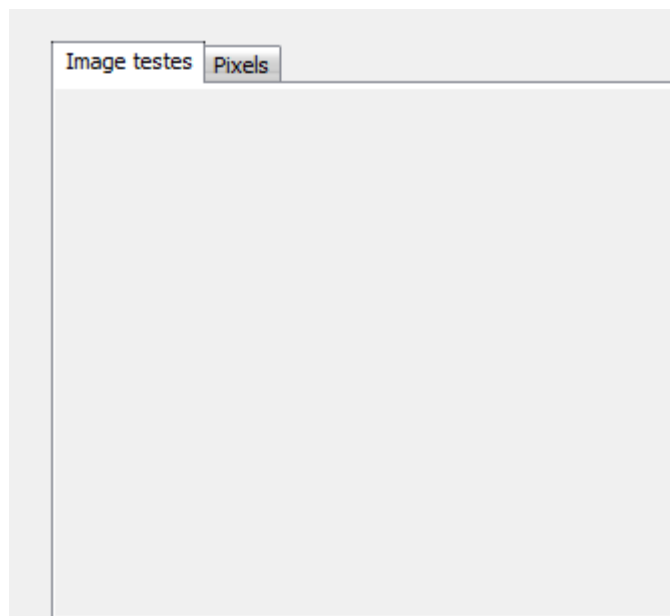
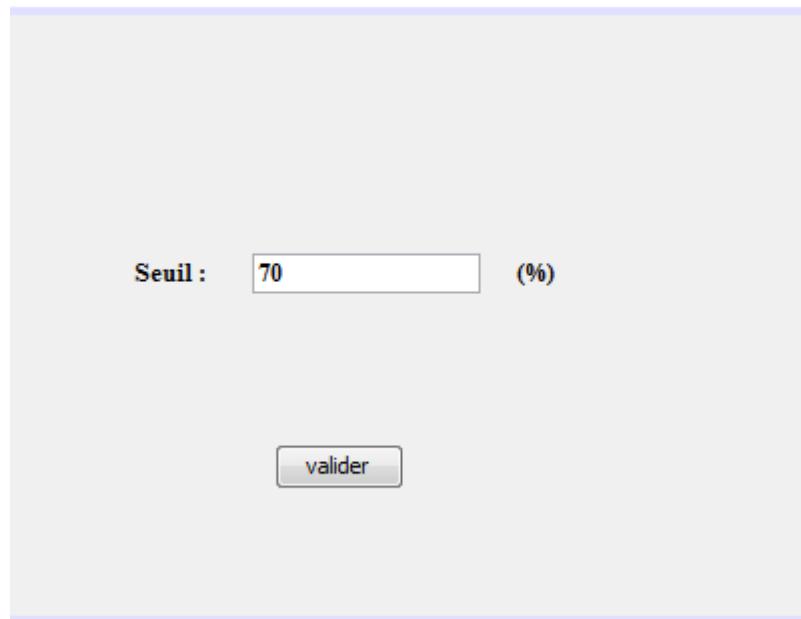


Figure 4.20 : la partie de détection de contour

Pour le choix de seuil :



The image shows a simple graphical user interface for setting a parameter. It consists of a light gray rectangular area. On the left side, the text "Seuil :" is displayed. To its right is a text input field containing the number "70". Further to the right, the text "(%)" is shown. Below these elements, centered horizontally, is a button with the text "valider".

Figure 4.21 : la forme de paramètre

Pour le comptage en utilisant l'algorithme Adaboost j'ai besoin de plusieurs méthodes, telle que la méthode qui calcule le taux de similarité et pour le boosting on ajoute la bibliothèque jboost (fichier jar).

Chargement de la bibliothèque jboost 2.4

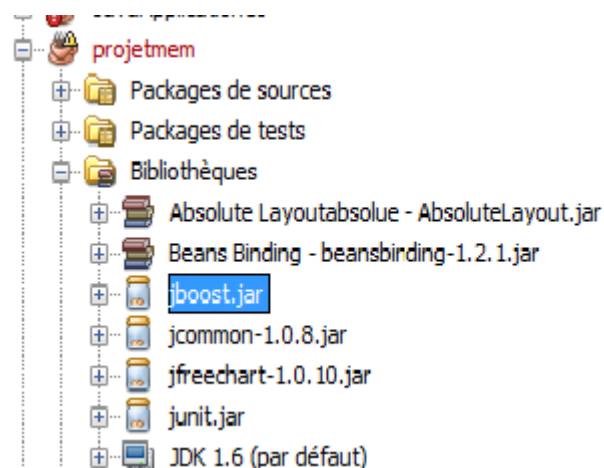
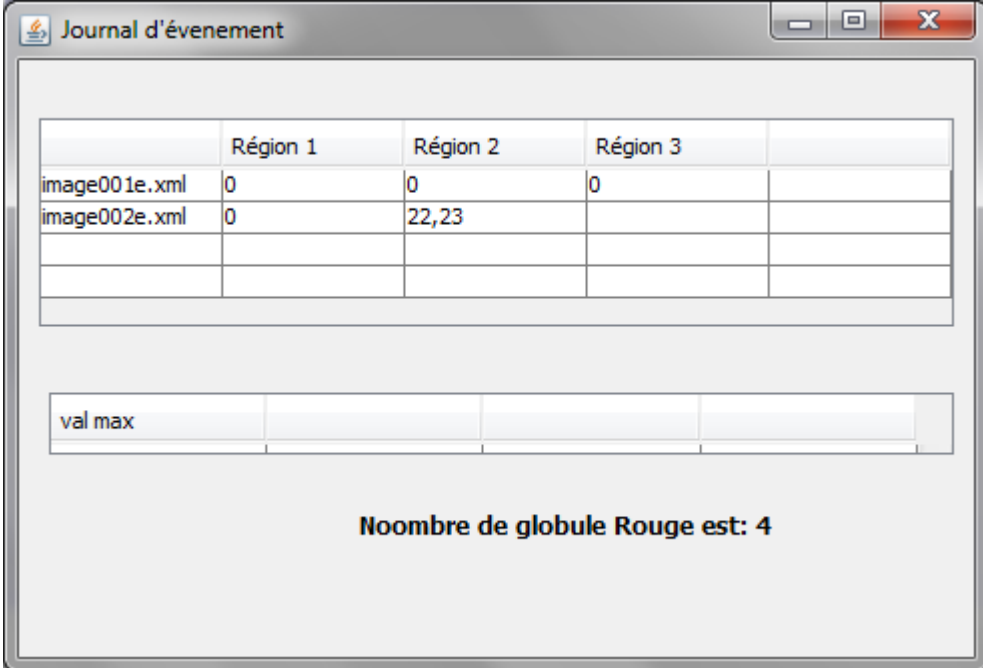


Figure 4.22 : l'ajout de bibliothèque jboost [25]

On clique sur menu résultat :



The screenshot shows a window titled "Journal d'événement" with a table and a text label. The table has four columns: an unlabeled column, "Région 1", "Région 2", and "Région 3". The rows contain data for "image001e.xml" and "image002e.xml". Below the table is a row of four input fields, the first containing "val max". At the bottom, a text label reads "Noombre de globule Rouge est: 4".

	Région 1	Région 2	Région 3	
image001e.xml	0	0	0	
image002e.xml	0	22,23		

val max			
---------	--	--	--

Noombre de globule Rouge est: 4

Figure 4.23 : la forme des résultats de comptage

V. Conclusion :

J'ai présenté dans ce chapitre les différentes approches implémentées dans ce projet en se basant sur l'algorithme AdaBoost, sont utilisation en java en ajoutant la bibliothèque jboost 2.4, puis j'ai présenté les différentes interfaces de mon prototype.

Ce qui reste à faire c'est l'affichage de l'image dans l'interface, l'affichage de tableau des pixels, chargement est l'affichage des images annoté.

Conclusion et perspectives :

En regardant l'état de l'art, j'ai constaté que l'analyse d'images en général, la segmentation et en particulier le comptage est un domaine très large qu'on ne peut cerner ou limiter à une approche ou une méthode particulière.

Mon but est de mettre en œuvre un algorithme de classification et de comptage de cellules sanguines.

L'algorithme de boosting choisit pour ce travail (AdaBoost) est une démarche fonctionnel bien défini qui consiste à décrire implicitement une classification performante des globules en prenant un classifieur faible pour construire un classifieur fort. Outil de comptage est implémenté en java.

Les approches utilisées ne sont pas les seules. Cela nous mènent à poser des questions sur des perspectives intéressantes comme une segmentation plus enrichie de façon où les contours soient détectés même avec un chevauchement. Pour adapter notre classifieur à d'autres méthodes de segmentation et avoir précisément des résultats de comptages plus exacts.

Nous pouvons dire que ce qui s'ouvre en perspective est plus large que se qui a été accompli.

Bibliographies

- [1] M.ANDRE, Introduction aux techniques de traitement d'images, Eyrolle 1987
- [2] R.C.GONZALES, P.WINTZ, Digital Image Processing-Addison Wesley -, 1977
- [3] A. D'HARDANCOURT, Fou du multimédia, Sybex 1995
- [4] Antoine MANZANERA, Cours : Traitement et reconnaissance d'images Master 2 IAD, 2007/2008
- [5] V. Berzins, Accuracy of laplacian edge detectors. Computer Vision Graphics and Image Processing, 1984.
- [6] Ali J. Tababai, O. Robert Mitchell, Edge location to sub pixel values in digital imagery, 1984
- [7] M.Bouterfas, M.Souidi, Développement d'un système de reconnaissance des cellules sanguine, mémoire, 2011.
- [8] Barba D. Benois J, Image segmentation by region-contour cooperation for image coding, 1992
- [9] Comaniciu, D. & Meer, P, Mean shift: A robust approach toward feature space analysis, 2003
- [10] Cheng Y, Mean shift mode seeking and clustering, 1995
- [11] Tom Mitchell, Machine Learning, 1997
- [12] A. Cornuéjols, L. Miclet Apprentissage Artificiel : Concepts et Algorithmes, 2002
- [13] P. Geurts., Applied inductive learning, 2008
- [14] M. Pietikainen, D. Harwood, Segmentation of color images using edge-preserving filters, 1986

Webographies

- [15] http://fr.wikipedia.org/wiki/Image_num%C3%A9rique, vu le 29/04/2012
- [16] http://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner5.html#toc29, Jean-marie, La liaison automatique des plusieurs images perçues sur un scanner, vu le 06/05/2012
- [17] <http://www.commentcamarche.net/contents/traitimg.php3>, vu le 14/06/2012
- [18] http://www-lisic.univ-littoral.fr/~vandenbroucke/enseignement/Cours_TI_Chapitre2.pdf, vu le 07/06/2012
- [19] http://www.funsci.com/fun3_fr/sang/sang.htm#0, vu le 26/04/2012
- [20] http://www.etud.iro.umontreal.ca/~rougierc/ift2730/cours/Cours10_IFT2730_2008_2.pdf
- [21] <http://docs.gimp.org/fr/plug-in-convmatrix.html>, vu le 07/06/2012
- [22] <http://humbert-florent.ftp-developpez.com/algorithmique/traitement/traitement.pdf>
- [23] http://urfist.enc.sorbonne.fr/anciensite/image_numerique/chapitre3_1.htm, vu le 29/04/2012
- [24] <http://fr.scribd.com/doc/34323411/Segmentation-des-images-medicales>, vu 21/06/2012
- [25] <http://jboost.sourceforge.net/downloads.html>, vu le 26/11/2012
- [26] http://labelme.csail.mit.edu/tool.html?collection=LabelMe&mode=i&folder=static_submitted_3d&image=img_575888355.jpg, vu le 18 /09/2012