



Ecole Polytechnique de l'Université de Tours  
Département Informatique  
64 avenue Jean Portalis  
37200 Tours, France  
Tél. +33 (0)2 47 36 14 14  
[polytech.univ-tours.fr](http://polytech.univ-tours.fr)

## Projet Recherche & Développement 2021-2022

# Segmentation/métrologie d'images dans l'agroalimentaire

Détecter les particules lipidiques dans l'image



Entreprise  
Polytech



Tuteur entreprise  
Hernan KENMOGNE

Étudiant  
Jiyuan WANG (DI5)

Tuteur académique  
Thierry BROUARD

# Liste des intervenants

## Entreprise

Polytech  
64 avenue Jean Portalis  
37200 Tours, France  
[polytech.univ-tours.fr](mailto:polytech.univ-tours.fr)



Nom	Email	Qualité
Jiyuan WANG	<a href="mailto:jiyuan.wang@univ-tours.fr">jiyuan.wang@univ-tours.fr</a>	Étudiant DI5
Thierry BROUARD	<a href="mailto:thierry.brouard@univ-tours.fr">thierry.brouard@univ-tours.fr</a>	Tuteur académique, Département Informatique
Hernan KENMOGNE	<a href="mailto:hkenmogne@groupe-bel.com">hkenmogne@groupe-bel.com</a>	Tuteur entreprise



# Avertissement

Ce document a été rédigé par Jiyuan WANG susnommé l'auteur.

L'entreprise Polytech est représentée par Hernan KENMOGNE susnommé le tuteur entreprise.

L'Ecole Polytechnique de l'Université de Tours est représentée par Thierry BROUARD susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assume l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



## Pour citer ce document

Jiyuan WANG, *Segmentation/méetrologie d'images dans l'agroalimentaire: Détecter les particules lipidiques dans l'image*, Projet Recherche & Développement, Ecole Polytechnique de l'Université de Tours, Tours, France, 2021-2022.

```
@mastersthesis{
  author={WANG, Jiyuan},
  title={Segmentation/méetrologie d'images dans l'agroalimentaire: Détecter les particules
    lipidiques dans l'image},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université de Tours},
  address={Tours, France},
  year={2021-2022}
}
```

# Table des matières

Liste des intervenants	<b>a</b>
Avertissement	<b>b</b>
Pour citer ce document	<b>c</b>
Table des matières	<b>i</b>
Table des figures	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1 Acteurs, enjeux et contexte .....	1
Particules riche en protéines .....	1
Particules cristallines.....	2
Poches de sérum.....	2
Gel lactique.....	2
Particules lipidiques .....	2
Matrice lipido-protéique .....	3
2 Objectifs.....	3
3 Hypothèses .....	3
4 Bases méthodologiques.....	4
<b>2 Description générale</b>	<b>5</b>
1 Environnement du projet .....	5
2 Caractéristiques des utilisateurs.....	5
3 Fonctionnalités du système .....	5
4 Structure générale du système.....	5

<b>3</b>	<b>État de l'art / Veille technologique</b>	<b>7</b>
1	Traitement d'image traditionnel.....	7
2	Deep Learning .....	8
2.1	Introduction .....	8
	FCN.....	9
	CNN .....	9
2.2	Exemple .....	9
3	SVM.....	11
3.1	Introduction .....	11
3.2	Exemple .....	12
4	Métriques .....	13
	Intersection sur Union (IoU).....	13
	Précision / Rappel / score F1.....	13
<b>4</b>	<b>Analyse et conception</b>	<b>14</b>
1	Analyse .....	14
1.1	Hypothèses utilisées.....	14
2	Spécifications.....	14
2.1	Modélisation proposée .....	14
2.2	Cas d'utilisation .....	14
2.3	Modélisation proposée .....	15
<b>5</b>	<b>Mise en oeuvre</b>	<b>17</b>
1	Outils et librairie utilisés.....	17
2	Segmentation.....	17
2.1	Données d'entrée .....	17
2.2	Traitement des images .....	17
2.2.1	Idée basique .....	17
2.2.2	Déterminer le seuil de binarisation.....	18
2.3	Traitement des données .....	19
2.4	Interface de logiciel.....	20
<b>6</b>	<b>Bilan et conclusion</b>	<b>22</b>
1	Bilan du semestre 9 .....	22
2	Bilan du semestre 10.....	22
3	Bilan sur la qualité .....	23
4	Bilan auto-critique.....	23

<b>Annexes</b>	<b>24</b>
<b>A Planification, gestion de projet</b>	<b>25</b>
1 Evolution du projet .....	25
2 Description des tâches.....	25
Tâche 1 : Analyse des besoins.....	25
Tâche 2 : Lecture et compréhension d'articles liés au projet .....	26
Tâche 3 : Rédiger le cahier des charges .....	26
Tâche 4 : Rédiger l'état d'art.....	26
Tâche 5 : Préparation de la soutenance à mi-parcours.....	26
Tâche 6 : Écrire l'algorithme pour compter le nombre de particules dans l'image segmentée .....	26
Tâche 7 : Implémenter l'algorithme pour compter plusieurs gra- phiques en même temps et traiter les données .....	26
Tâche 8 : Traiter les images de différentes manières et implémenter la binarisation.....	27
Tâche 9 : Développer une interface graphique, évaluer l'extraction...	27
Tâche 10 : Compléter le logiciel, le tester en entreprise .....	27
Tâche 11 : Rédiger le rapport final .....	27
Tâche 12 : Préparer la soutenance final.....	27
<b>B Description des interfaces</b>	<b>28</b>
1 Interfaces matérielles/logicielles .....	28
2 Interfaces homme/machine.....	28
Définir la taille des particules de lipides et la largeur de l'image.....	28
Exporter la liste des particules lipidiques.....	28
Dessiner un histogramme statistique.....	28
<b>C Cahier de Spécifications</b>	<b>29</b>
1 spécifications Fonctionnelles.....	29
1.1 Définition de la fonction 1 : ImageProcessing .....	29
Description de la fonction 1 : .....	29
1.2 Définition de la fonction 2 :Interface de logiciel .....	30
Description de la fonction 2 : .....	30
2 Spécifications non fonctionnelles .....	31
2.1 Contraintes de développement et conception .....	31
2.2 Contraintes de fonctionnement et d'exploitation.....	31
2.2.1 Performances.....	31
2.2.2 Capacités .....	31
2.2.3 Contrôlabilité .....	31
2.2.4 Sécurité .....	31

<b>D</b>	<b>Cahier du développeur</b>	<b>32</b>
1	Introduction .....	32
2	Diagrammes architecturaux et UML .....	32
3	Descriptions détaillées de données exploitées .....	33
4	Descriptions détaillées des classes, modules, réalisations .....	34
4.1	Classe "ProcessImage" .....	34
4.1.1	Propriété .....	34
4.1.2	Méthode .....	35
4.2	Classe "ViewMain" .....	35
4.2.1	Propriété .....	35
4.2.2	Méthode .....	35
<b>E</b>	<b>Document d'installation</b>	<b>36</b>
<b>F</b>	<b>Document d'utilisation</b>	<b>37</b>
<b>G</b>	<b>Cahier de test</b>	<b>40</b>
1	Tests fonctionnel .....	40
<b>H</b>	<b>Bibliographie</b>	<b>42</b>

# Table des figures

## 1 Introduction

1.1	Particules riche en protéines.....	1
1.2	Particules cristallines .....	2
1.3	Poches de sérum .....	2
1.4	Gel lactique .....	3
1.5	Particules lipidiques.....	3

## 2 Description générale

2.1	model-view .....	6
-----	------------------	---

## 3 État de l'art / Veille technologique

3.1	Image engineering process.....	7
3.2	Image segmentation .....	8
3.3	Image Binarization .....	8
3.4	General Approach.....	8
3.5	FCN .....	9
3.6	Architecture of convolutional neural networks .....	10
3.7	General Approach.....	10
3.8	General Approach.....	11
3.9	Main process with SVM.....	12

## 4 Analyse et conception

4.1	Diagramme de Classe.....	15
4.2	Cas d'utilisation .....	15

4.3	Activity .....	16
<b>5</b>	<b>Mise en oeuvre</b>	
5.1	Original(Gauche) et Binarisation(droite) .....	18
5.2	Débruitage.....	18
5.3	Threshold10.....	19
5.4	Threshold35.....	19
5.5	Threshold25.....	20
5.6	ViewMain .....	21
5.7	Histogramme .....	21
<b>A</b>	<b>Planification, gestion de projet</b>	
A.1	Le diagramme de Gantt Final .....	25
A.2	Le diagramme de Gantt Final .....	25
<b>D</b>	<b>Cahier du développeur</b>	
D.1	class diagram.....	32
D.2	Cas d'utilisation .....	33
D.3	Activity .....	33
D.4	Données Exploitées.....	34
<b>F</b>	<b>Document d'utilisation</b>	
F.1	Importer.....	37
F.2	Exporter.....	38
F.3	Dessiner l'histogramme .....	38
F.4	Traiter tous les images .....	39
F.5	Erreur .....	39
<b>G</b>	<b>Cahier de test</b>	
G.1	Binarisation .....	40
G.2	Résultat.....	41

# 1

## Introduction

### 1 Acteurs, enjeux et contexte

Ce projet est supervisé par Thierry BROUARD, et est destiné au Groupe Bel (Basé à Vendôme). Ce projet se fera en collaboration avec mon encadrant, Thierry BROUARD. Acteurs :

- Client : Hernan KEMNOGNE and Benoit GOLDSCHMIDT
- MOA : Thierry BROUARD
- MOE : Jiyuan WANG

Les images présentées montrent différents types d'objets (des particules mêlant, plus ou moins, et principalement, des protéines et des lipides). On cherche à identifier six types de structure – cinq catégorisant les structures, et un qui résulte du fait que la zone n'appartient à aucun des cinq autres types, présentés ci-dessous.

#### Particules riche en protéines

Ces objets sont très visibles dans le plan vert, de forme plutôt arrondie, mais qui peut être concave, de teinte assez uniforme, mais potentiellement texturée (notamment en cas d'émulsion) ou qui peut présenter une zone centrale sombre (dans le plan vert) ou lumineuse (dans le plan rouge) correspondant à une goutte de lipide recouverte de protéine.

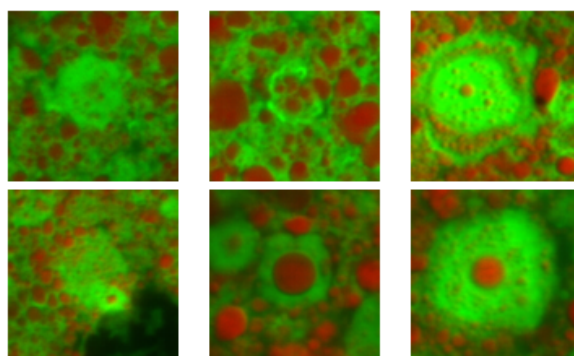


Figure 1.1 – *Particules riche en protéines*

### Particules cristallines

Ces particules cristallines ressemblent aux particules riches en protéines, mais leur zone centrale est plus ou moins sombre. À l'extrême, on observe une surface presque noire, bordée de vert (selon le plan d'observation). De l'autre côté de l'échelle, on constate une surface plutôt verte, mais on distingue clairement des zones sombres à l'intérieur de la surface verte de la particule.

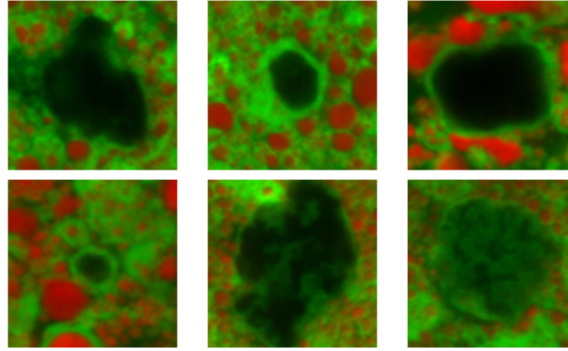


Figure 1.2 – *Particules cristallines*

### Poches de sérum

Les poches de sérum sont des zones sombres, de forme quelconque, n'ayant pas la caractéristique d'être bordées de protéine, car elles constituent une rupture dans la matrice lipido-protéique. Elles ne sont pas nécessairement d'aspect uniforme, et peuvent présenter, par exemple, de petites gouttelettes lipidiques en suspension.

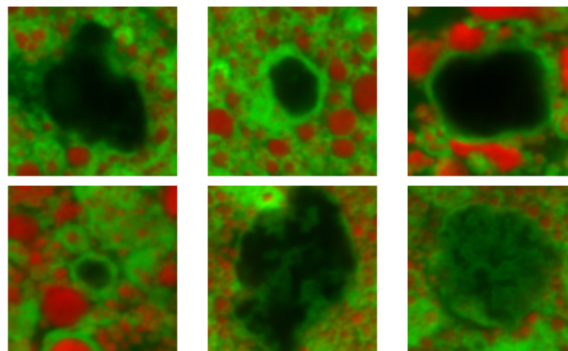


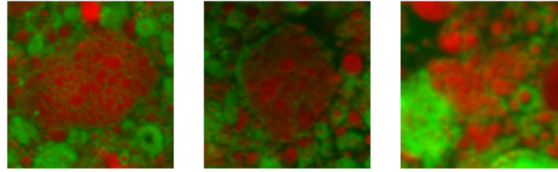
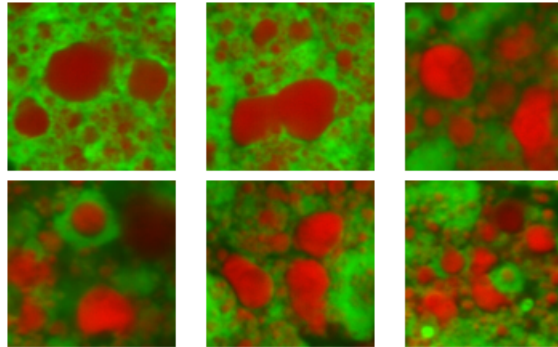
Figure 1.3 – *Poches de sérum*

### Gel lactique

Le gel lactique est une zone qui semble formée de petites gouttelettes de gras agglomérées, de texture assez rugueuse, plutôt du domaine rouge (dans le vert, cet zone ressemble à une éponge de couleur sombre avec cavités noires ou presque et dans le rouge on observe un agglomérat de gouttelettes).

### Particules lipidiques

Enfin, les structures lipidiques sont de surface variable, de forme plutôt convexe (sauf dans le cas où elle ne sont pas émulsionnées, où la forme peut être quelconque), souvent arrondie dans le cas des gouttelettes, de teinte rouge plus ou moins intense, mais assez uniforme.

Figure 1.4 – *Gel lactique*Figure 1.5 – *Particules lipidiques*

### Matrice lipido-protéique

Par définition, c'est tout ce qui n'a pas été classé dans l'une des cinq catégories précédentes. Il reste une zone mixte, formée d'un fond vert d'intensité variable, et contenant de nombreuses petites surfaces de type gouttelette, rouges (là aussi d'intensité variable). Cette structure relie toutes les autres entre elles.

## 2 Objectifs

Le but de ce projet est de définir des algorithmes pour automatiser la segmentation d'images. L'algorithme est censé décomposer une image en un ensemble de régions d'intérêt, chacune caractérisée selon la classe à laquelle elle appartient. Et peut déterminer la zone, la taille et la position de chaque zone de l'image. Ensuite, nous développerons une interface graphique. Le client peut importer plusieurs images, réaliser une segmentation et obtenir des particules en fonction de la plage de taille qu'il a saisie. Enfin, dessinez un histogramme basé sur les informations des particules et exportez la position, la taille et le nombre total de ces particules souhaitées dans un fichier .csv.

## 3 Hypothèses

Nous n'avons pas suffisamment de données d'image étiquetées, nous avons donc fini par utiliser le traitement d'image traditionnel. Et dans la première étape, nous n'avons détecté des particules lipidiques que dans six catégories de régions. Pour faciliter les calculs statistiques, nous supposons que toutes les particules lipidiques sont circulaires et leur diamètre est calculé à partir de leur surface.  $D = \sqrt{4 \cdot \text{Area} / \pi}$ .

## 4 Bases méthodologiques

Afin de mieux suivre le projet, je rencontre régulièrement des professeurs et des clients pour discuter. J'utilise les outils suivants tout au long de ce projet :

- Pour le versioning du projet : GitHub
- Pour le transfert de documents avec le client : Gmail
- Pour communiquer avec les clients
- Pour la gestion de l'avancement du projet : le diagramme de Gantt
- Pour la rédaction du rapport : LaTeX

# 2

## Description générale

### 1 Environnement du projet

Le projet vise à développer un logiciel pour réaliser la segmentation d'images et filtrer les particules appropriées. Il sera implémenté en Python, avant de coder, nous utilisons AstahUML pour réaliser la construction des diagrammes UML. Ensuite, nous utiliserons PyCharm pour coder. Pour les différentes parties du projet, nous utiliserons différents outils, qui seront précisés ultérieurement. Nous verrons plus précisément les bibliothèques utilisées dans chaque partie concernée.

### 2 Caractéristiques des utilisateurs

Le logiciel doit être simple d'utilisation, et accessible à tous : Nous veillerons donc à ce que l'interface soit compréhensible par des personnes qui ne connaissent pas forcément ce projet. Il doit donc être facilement utilisable par des personnes dont le domaine d'expertise réside dans l'analyse de la composition des fromages.

### 3 Fonctionnalités du système

Le système fournira aux utilisateurs de multiples fonctions. Pour les clients, ils peuvent

- Importez une image, identifiez et segmentez les particules lipidiques en fonction de l'image actuelle ou de toutes les images
- Informations statistiques telles que la taille, la quantité et l'emplacement des particules lipidiques
- Dessinez un histogramme statistique basé sur le diamètre et la surface des particules définies
- Exporter le fichier .csv par informations sur les particules lipidiques dans une seule image ou toutes les images

### 4 Structure générale du système

Sur la base du modèle MVC traditionnel, nous combinons ici des objets vue et contrôleur, et le résultat est une architecture modèle/vue. Cela sépare toujours la façon dont les données

sont stockées de la façon dont elles sont présentées à l'utilisateur, mais fournit un cadre plus simple basé sur les mêmes principes. Cette séparation permet d'afficher les mêmes données dans plusieurs vues différentes et d'implémenter de nouveaux types de vues sans changer la structure de données sous-jacente.

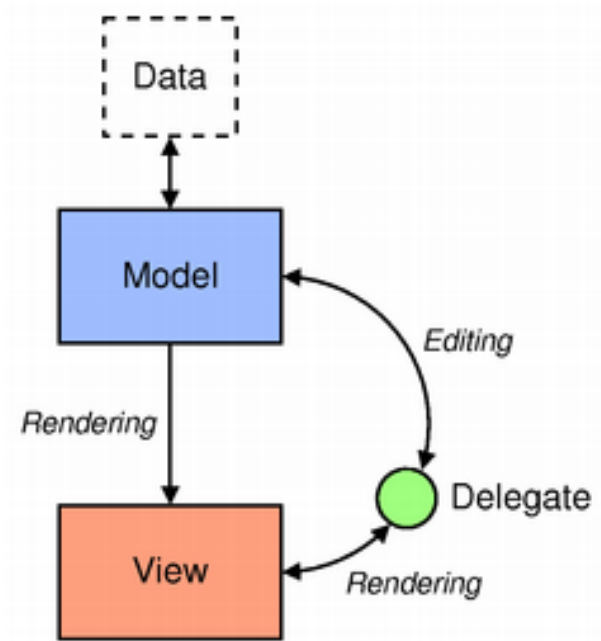


Figure 2.1 – *model-view*

Le modèle stocke les informations pertinentes de chaque image, y compris le résultat après chaque étape d'amélioration de l'image, et les informations sur les particules après segmentation. La vue comprend la vue principale pour les opérations de l'utilisateur et la vue pour dessiner des histogrammes statistiques. Elle obtient des index de modèle à partir du modèle et les utilisateurs peuvent effectuer différentes opérations en fonction de leurs différents besoins.

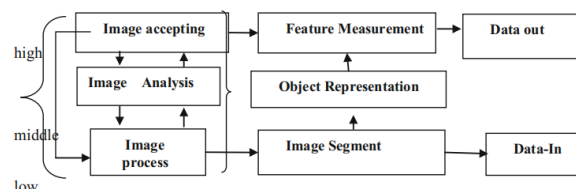
# 3

## État de l'art / Veille technologique

La segmentation automatique des cellules consiste à déterminer si certains types de cellules sont présents dans une image d'entrée et à les localiser dans l'image. Il existe de nombreuses méthodes pour y parvenir. Par exemple, avec le traitement d'image traditionnel, le SVM et le deep learning.

### 1 Traitement d'image traditionnel

L'ingénierie de l'image est divisée en trois niveaux, comme le montre la figure 3-1. [3]



**Figure 3.1** – *Image engineering process*

Le traitement d'image est les opérations de niveau inférieur. Ces opérations sont effectuées au niveau du pixel. Cette couche prend l'image et modifie cette image de version sous une autre forme ou effectue certaines transformations entre les images et améliore également l'effet visuel de l'image d'entrée. L'analyse d'image est l'opération centrale de cette structure qui se concentre sur la mesure. L'ACP produit un ensemble net d'images à partir d'un ensemble donné. La compréhension de l'image consiste à étudier plus avant et à se connecter ainsi qu'à l'explication des images originales. La segmentation d'image est le processus de fractionnement ou de division de l'image en parties plus petites. Dans les bords de segmentation d'image, les frontières, les régions sont identifiées pour le traitement. Dans le processus de segmentation d'image, la sortie se présentera sous la forme de pixels qui se recouvrent pour former une image entière.

La principale méthode pour réaliser la segmentation est illustrée dans la figure 3-2.

Il y a une autre idée de faire la détection d'objet. Dans "Adaptive-document-image-binarization\_-2000\_Pattern-Recognition", ils ont proposé une nouvelle méthode adaptative de binarisation d'image de document, qui considère la page comme une collection de sous-composants tels que le texte, l'arrière-plan et les images. Résolez les problèmes causés par le bruit, l'éclairage et de

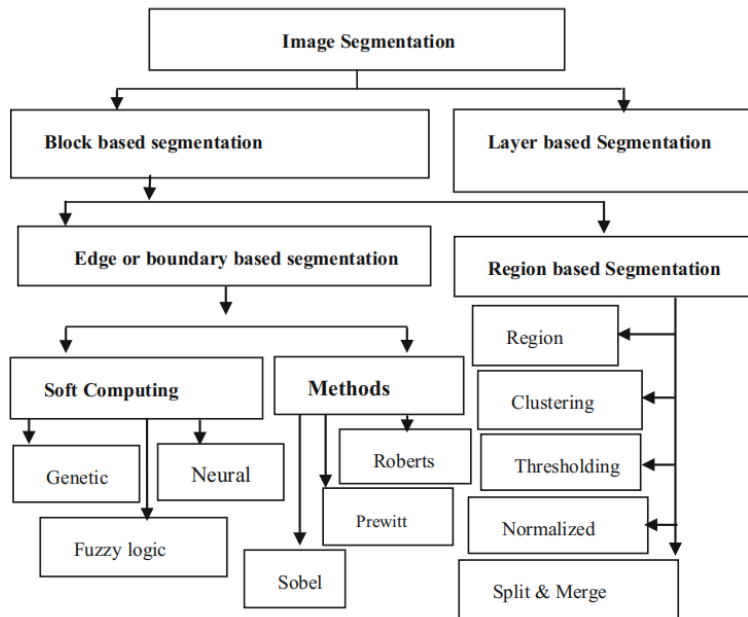


Figure 3.2 – Image segmentation

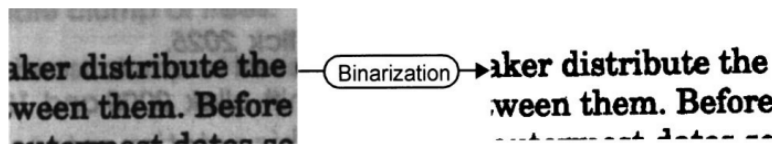


Figure 3.3 – Image Binarization

nombreuses dégradations liées au type de source. Et leur but est d'obtenir le contenu principal plutôt que le contenu du verso. Comme la figure 3-3. [4]

L'approche générale est illustrée à la figure 3-4.

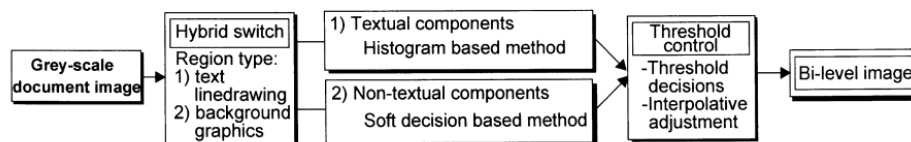


Figure 3.4 – General Approach

## 2 Deep Learning

### 2.1 Introduction

Pour faire de la segmentation avec l'apprentissage profond, nous avons les catégories suivantes : [5]

- Réseaux entièrement convolutifs
- Modèles convolutifs avec modèles graphiques
- Modèles basés sur les encodeurs-décodeurs
- Modèles basés sur des réseaux multi-échelles et pyramidaux
- Modèles basés sur le R-CNN (par exemple, la segmentation)

- Modèles convolutifs dilatés et famille DeepLab
- Modèles basés sur les réseaux neuronaux récurrents
- Modèles basés sur l'attention
- Modèles génératifs et formation contradictoire
- Modèles convolutifs avec modèles de contour actif

## FCN

Long et al. ont proposé l'utilisation de réseaux entièrement convolutifs (FCN). Un FCN ne contient que des couches convolutives, ce qui lui permet de prendre des images de n'importe quelle taille et de générer des cartes de segmentation de la même taille. L'auteur a modifié les architectures CNN existantes, telles que VGG16 et GoogLeNet, en remplaçant toutes les couches entièrement connectées par des couches entièrement convolutives afin de gérer des entrées et des sorties de taille non fixe. Par conséquent, le modèle produit une carte de segmentation spatiale.

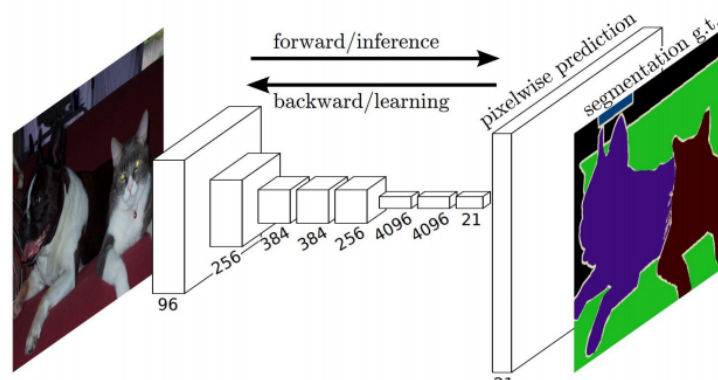


Figure 3.5 – FCN

Cependant, malgré sa popularité et son efficacité, le modèle FCN classique présente certaines limites : il n'est pas assez rapide pour l'inférence en temps réel, il ne tient pas compte efficacement des informations contextuelles globales et il n'est pas facilement transférable aux images 3D.

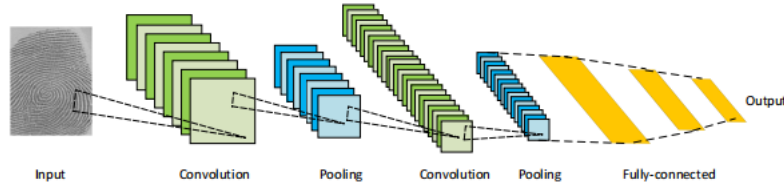
## CNN

Le CNN est principalement composé de trois types de couches : i) la couche convolutive, dans laquelle le noyau pondéré est convolué pour extraire les caractéristiques ; ii) la couche non linéaire, qui applique des fonctions d'activation à la cartographie des caractéristiques pour réaliser la construction du réseau de fonctions non linéaires Modulus ; iii) la couche de mise en commun, qui remplace un petit voisinage de la carte des caractéristiques par certaines informations statistiques. En empilant les couches pour former une pyramide multi-résolution, les couches de niveau supérieur apprennent les caractéristiques d'un champ réceptif de plus en plus large. Le principal avantage du CNN est que tous les champs réceptifs d'une couche partagent des poids, ce qui permet de réduire considérablement le nombre de paramètres par rapport à un réseau neuronal entièrement connecté. Parmi les architectures CNN les plus connues, citons : AlexNet, VGGNet, ResNet.

L'architecture des réseaux neuronaux convolutifs est présentée dans la figure 3.6.

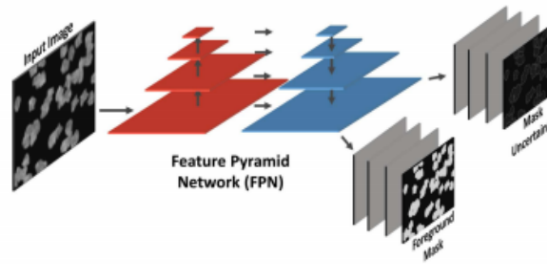
## 2.2 Exemple

Dans l'article intitulé "Using Deep Learning for Segmentation and Counting within Microscopy Data" (Utilisation de l'apprentissage profond pour la segmentation et le comptage dans les



**Figure 3.6** – *Architecture of convolutional neural networks*

données de microscopie)[2], les auteurs décrivent une approche de réseau neuronal convolutif, utilisant un réseau pyramidal de caractéristiques récemment décrit combiné à un réseau neuronal de type VGG, pour la segmentation et le comptage ultérieur des cellules dans une image de microscopie donnée.



**Figure 3.7** – *General Approach*

Pour la segmentation cellulaire, le FPN est un réseau d'extraction de caractéristiques permettant de construire des cartes de caractéristiques à plusieurs échelles spatiales. Il s'agit d'un algorithme efficace en termes de calcul qui permet de réticuler l'image convolutive sous-échantillonnée et l'image suréchantillonnée correspondante dans le réseau. Le réticulage utilise une seule couche convolutive 3x3 avec 128 filtres et un rembourrage pour préserver la taille de l'image sous-échantillonnée. Ces liens croisés permettent au réseau non seulement de déduire des caractéristiques apparentées à différentes échelles, mais aussi de déduire leur corrélation spatiale. Enfin, chaque couche de suréchantillonnage passe par trois couches convolutives 3x3 : les deux premières ont 256 filtres, et la dernière n'en a qu'un seul pour générer le masque de premier plan en noir et blanc. À l'exception de la couche de sortie finale, toutes les couches convolutionnelles sont traitées par normalisation par lots. Le résultat est un nombre quelconque de masques de premier plan à différentes échelles de l'image originale. Pour ce réseau, chaque couche de la pyramide est sous-échantillonnée par un facteur de 2, et la profondeur de la pyramide est de 4, résultant en quatre masques de sortie, dont le plus grand est la moitié de la taille de l'image originale, et c'est le masque que le réseau de compte utilisé plus tard. Un ensemble de couches convolutionnelles distinctes mais similaires est utilisé pour générer l'incertitude du masque, exprimée en tant que variance logarithmique.

Chaque masque a sa fonction de perte associée, qui contient deux éléments : la perte arbitraire par rapport au vrai premier plan et la perte de variation totale (TV). Comme mentionné précédemment, la perte arbitraire est représentée par une fonction de perte L2 avec l'ajout d'une incertitude de masque. La perte TV lisse le masque de premier plan et supprime tous les artefacts indésirables de manière semi-autonome. La somme de ces pertes, pour chaque masque mis à l'échelle, est ensuite utilisée comme perte finale du modèle FPN.

Pour le comptage des cellules, ils utilisent le réseau VGG-11, qui consiste en 11 couches de convolution bidimensionnelles avec une taille de filtre de 3x3 pixels. Au fur et à mesure que le réseau évolue, le nombre de filtres varie de 64 à 512. En outre, chaque couche de convolution est

suivie d'une couche de normalisation par lots, dont il a été démontré qu'elle améliore la stabilité de la formation en empêchant l'instabilité numérique dans le calcul du gradient. Cette couche est suivie d'une activation de type ReLU (leaky rectified linear unit) et d'une couche de max-pooling. Le réseau VGG-11 est utilisé comme extracteur de caractéristiques pour le réseau de comptage. Pour générer les prédictions de comptage, ils utilisent trois couches entièrement connectées qui sont à nouveau séparées par une couche de normalisation de lot et une ReLU fuyante. Ces couches entièrement connectées ont des dimensions de 1024, 512 et 1, respectivement, et sont terminées par un seul nombre à virgule flottante. La dernière couche comprend une couche ReLU pour empêcher le réseau de produire des nombres négatifs. Un réseau entièrement connecté similaire sans couche ReLU finale est utilisé pour prédire la variance logarithmique.

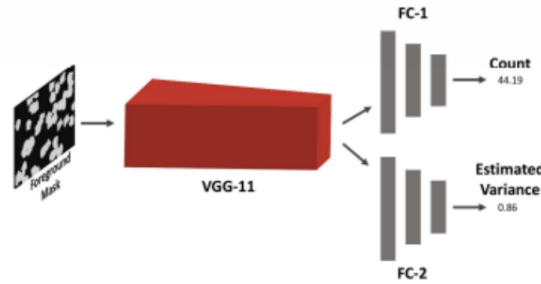


Figure 3.8 – General Approach

Enfin, ils utilisent le jeu de données BBBC005 de la Bioimage Benchmark Collection du Broad Institute pour s'entraîner et pour le tester. Après l'entraînement, pour la segmentation, l'erreur quadratique moyenne (EQM) sur 100 images de validation a convergé vers une valeur inférieure à 0,1 après environ 50 époques. L'inspection visuelle des masques sur chaque époque a également corroboré cette convergence, les masques étant presque identiques à la vérité terrain. La convergence des masques d'incertitude s'est également produite au cours des 50 époques, évoluant d'une incertitude uniforme sur l'image vers les contours des groupes de cellules. Pour le comptage, le MSE de validation moyen des comptages de 1420 masques générés séparément a convergé après 50 époques vers une valeur finale inférieure à 11,2. Le meilleur modèle est capable d'atteindre une valeur R2 de 0,987, avec une erreur L1 moyenne de 2,4 cellules, sur l'ensemble de test.

## 3 SVM

### 3.1 Introduction

Les machines à vecteurs de support sont des modèles d'apprentissage supervisé avec des algorithmes d'apprentissage associés qui analysent les données pour la classification et l'analyse de régression. Étant donné un ensemble d'exemples d'apprentissage, chacun étant marqué comme appartenant à l'une des deux catégories, un algorithme d'apprentissage SVM construit un modèle qui attribue les nouveaux exemples à l'une ou l'autre catégorie, ce qui en fait un classificateur linéaire binaire non probabiliste. [1] Le SVM fait correspondre les exemples d'apprentissage à des points dans l'espace de manière à maximiser la largeur de l'écart entre les deux catégories. Chaque classe possède un vecteur de caractéristiques à  $d$  dimensions  $X = (x_1, x_2, \dots, x_d)$  et son niveau de classe  $y$  se voit attribuer deux valeurs  $+1, -1$ . L'hyperplan limite est alors défini comme suit :

$$W^T X + b = 0$$

où  $W$  est le vecteur des coefficients de pondération, et  $b$  est le terme de biais. L'objectif de cet algorithme est de minimiser la fonction de coût  $J(W)$ , définie comme suit :

$$J(W) = \frac{1}{W^T W}$$

Et pour les cas de données linéairement non séparables :

$$W^T \Phi(x) + b = 0$$

Les nouveaux exemples sont ensuite mis en correspondance dans ce même espace et leur appartenance à une catégorie est prédite en fonction du côté de l'écart où ils se trouvent.

## 3.2 Exemple

Dans l'article "Machine learning approach for automated screening of malaria parasite using light microscopic images", [1] les auteurs ont combiné la statistique F, les techniques d'apprentissage statistique, c'est-à-dire l'apprentissage bayésien et la machine à vecteur de support (SVM), afin de fournir une précision de classification plus élevée en utilisant le meilleur ensemble de caractéristiques discriminantes.

Le processus principal est le suivant figure 3.9 :

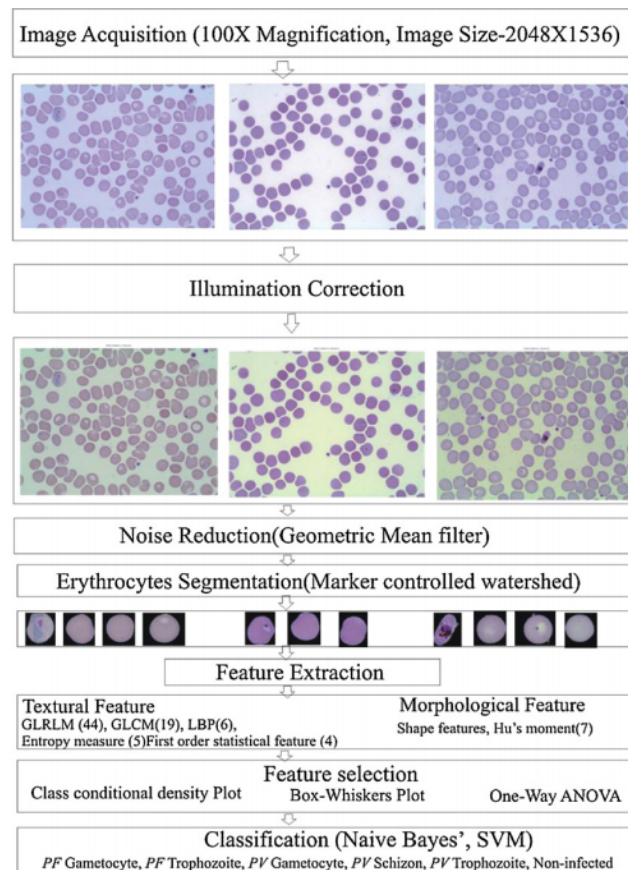


Figure 3.9 – Main process with SVM

Après l'entraînement, le SVM conduit à une précision de dépistage de 83,5 %, une sensibilité de 96,62 % et une spécificité de 88,51 % pour la reconnaissance des deux parasites.

## 4 Métriques

Il existe plusieurs façons d'évaluer le résultat de la segmentation d'une image.

### Intersection sur Union (IoU)

L'IoU est une norme permettant de mesurer la précision de la détection d'objets correspondants dans un ensemble de données spécifique. Plus la corrélation est élevée, plus la valeur est élevée. C'est le résultat de la division de la partie de chevauchement des deux régions par la partie de l'ensemble des deux régions. En segmentation, la zone est la carte de segmentation prédite et la vérité terrain.

$$IoU = \frac{A \cap B}{A \cup B}$$

où A et B sont la vérité terrain et la carte de segmentation prédite.

### Précision / Rappel / score F1

Pour un problème de classification binaire, l'échantillon a deux catégories, positive et négative. Il existe alors 4 combinaisons entre le résultat prédit par le modèle et l'étiquette réelle : TP, FP, FN, TN. TP fait référence à la fraction de vrais positifs, FP à la fraction de faux positifs, FN à la fraction de faux négatifs et TN à la fraction de vrais négatifs.

La précision indique combien de prédictions sont correctes dans l'échantillon où la prédiction est positive, le rappel indique combien d'échantillons avec des étiquettes réelles positives sont prédits avec succès, comme suit :

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Habituellement, nous combinons les taux de précision et de rappel et le score F1 est défini comme la moyenne harmonique de la précision et du rappel :

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

# 4

## Analyse et conception

### 1 Analyse

L'objectif du projet est de mettre en place un logiciel permettant de segmenter des particules lipidiques dans des images statistiques. Le plan initial était d'utiliser le traitement d'image traditionnel, les modèles SVM et de réseau neuronal pour segmenter les images respectivement, mais comme BEL ne pouvait pas fournir suffisamment de données étiquetées, seul le traitement d'image traditionnel (binarisation d'image) a été utilisé pour réaliser la segmentation à la fin. En termes de logiciel, en important les images dans le dossier, nous pouvons traiter respectivement l'image actuelle ou toutes les images du dossier, dessiner l'histogramme et exporter le fichier.

#### 1.1 Hypothèses utilisées

Dans l'image, les particules lipidiques sont des particules rouges sur fond vert (protéines, etc.). Nous résumons la tâche de modélisation consistant à extraire les points rouges des images. Nous traitons toutes les particules comme des cercles et le diamètre est calculé à partir de la surface

### 2 Spécifications

#### 2.1 Modélisation proposée

Nous utilisons l'architecture du modèle MV pour implémenter le logiciel, et la vue implémente la segmentation et les statistiques en appelant la méthode de traitement d'image dans la classe ProcessImage. Le diagramme de classe est illustré dans la figure 4.1.

#### 2.2 Cas d'utilisation

Les capacités de l'utilisateur sont illustrées dans la figure 4.2.

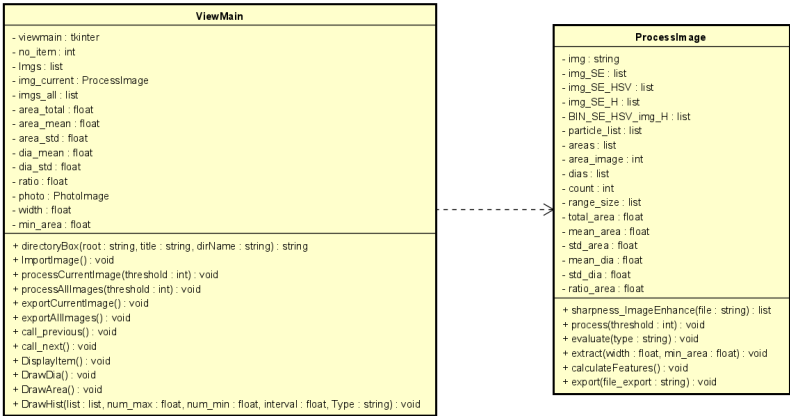


Figure 4.1 – Diagramme de Classe

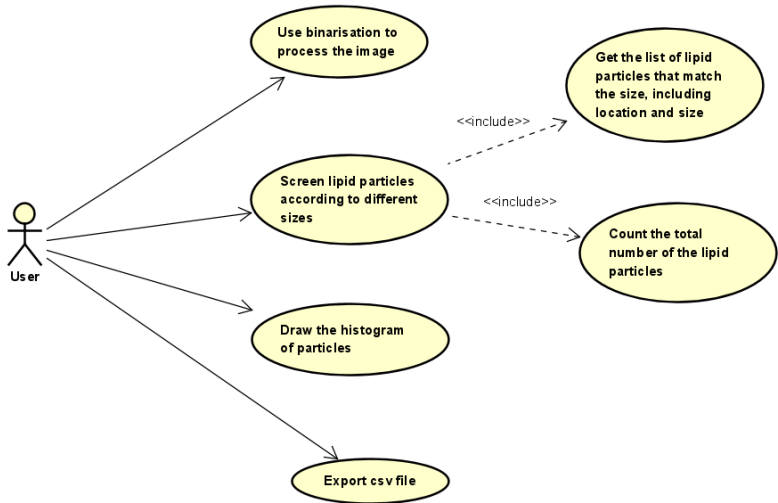
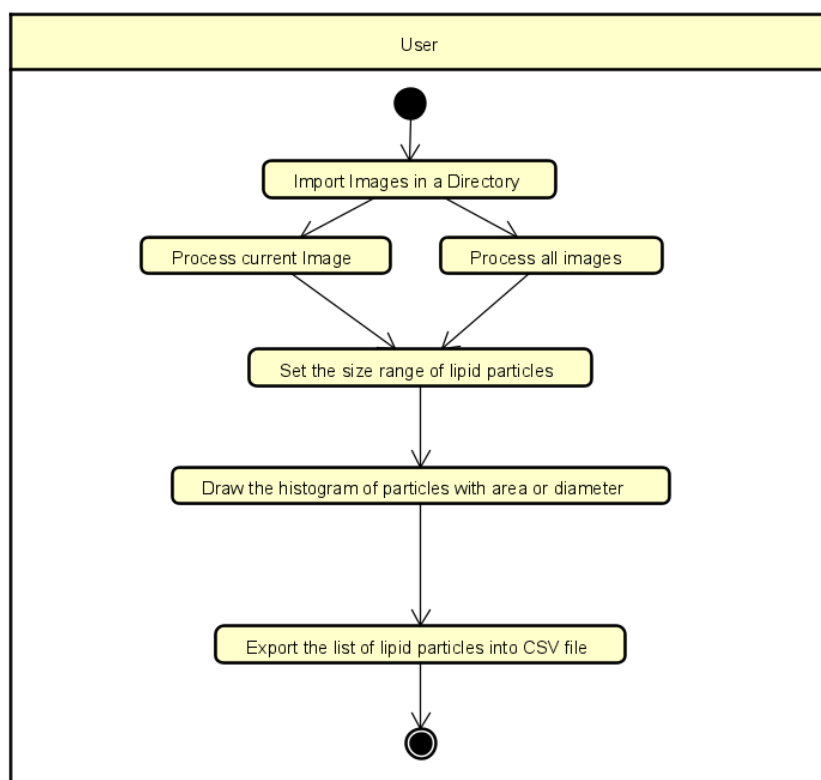


Figure 4.2 – Cas d'utilisation

2.3 Modélisation proposée

Le flux de son utilisation du logiciel est illustré dans la figure 4.3.

**Figure 4.3** – *Activity*

# 5

## Mise en oeuvre

### 1 Outils et librairie utilisés

Le codage de ce projet est implémenté sur la base de pycharm + python, et les bibliothèques/packages associés sont utilisés comme suit :

- Opencv pour le traitement lié à l'image
- Tkinter pour la conception d'interfaces
- Numpy et Matplotlib pour le traitement des données
- Pyinstaller pour empaqueter et générer des fichiers .exe

### 2 Segmentation

#### 2.1 Données d'entrée

La taille des images est de  $2048 \times 2048$  pixels, au format TIFF, et l'élément étudié peut être identifié sur la base de la fluorescence excitée par le laser par sa composition. Les deux couleurs principales sont le rouge, qui est lié aux graisses, et le vert, qui est lié aux protéines. Ainsi pour chaque observation, nous avons trois images : une image du plan vert, une image du plan rouge, et l'ensemble résultant.

#### 2.2 Traitement des images

##### 2.2.1 Idée basique

Nous affinons d'abord l'image. En agissant sur les deux côtés du bord de l'objet, le bord semble plus proéminent et net, et les résultats suivants sont obtenus. Ensuite, nous convertissons l'image en canal HSV et ne gardons que le canal H, puis, nous binarisons l'image du canal H pour obtenir des particules lipidiques, les points noirs dans l'image. Le résultat après traitement est dans la figure 5.1.

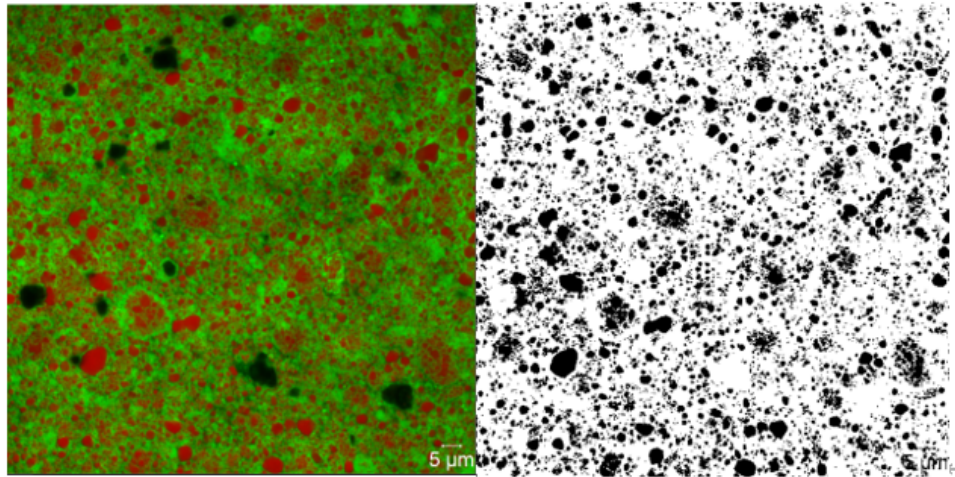


Figure 5.1 – *Original(Gauche) et Binarisation(droite)*

En comparant l'image binarisée et l'image d'origine, nous pouvons constater que l'image binarisée contient beaucoup de bruit poivre et sel. Par conséquent, nous effectuons une opération de fermeture et un filtrage lissage sur l'image pour améliorer la qualité de l'image et augmenter le signal-à-bruit. Permet aux particules d'obtenir de meilleures performances. Le résultat après traitement est dans la figure 5.2.

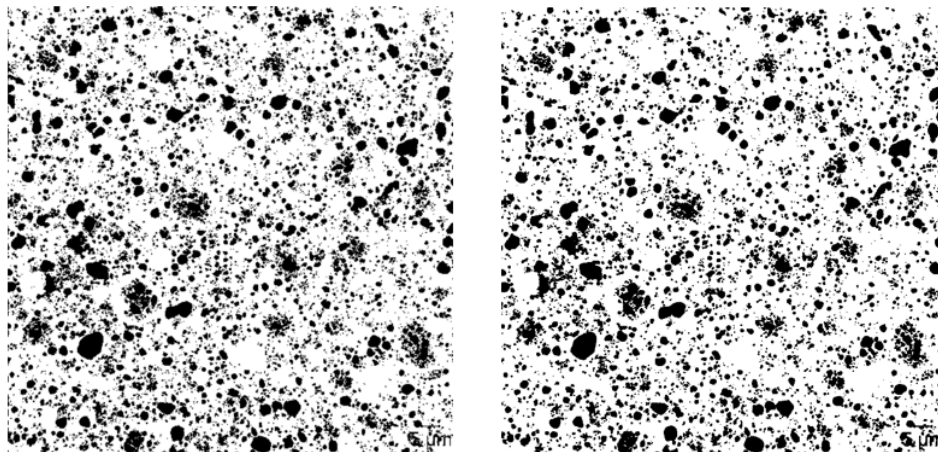


Figure 5.2 – *Débruitage*

### 2.2.2 Déterminer le seuil de binarisation

Après binarisation, l'image ne contient que des couches noires et blanches, où le noir représente les particules lipidiques et le blanc représente les autres parties. En définissant différents seuils, nous pouvons modifier la sensibilité de la binarisation au rouge. Plus le seuil est petit, moins la zone rouge est conservée et plus le seuil est grand, plus la zone rouge est conservée. Pour déterminer le meilleur seuil, nous proposons les notions de « false positive image » et « false positive image ».

Pour l'image binarisée traitée, les particules lipidiques sont la zone noire, la valeur de pixel est 0 et la valeur de pixel de la zone blanche est 1. Nous multiplions l'image originale par l'image binarisée pour obtenir une zone qui ne devrait pas être rouge. De cette façon, nous voyons tous

les pixels rejetés, nous pouvons donc savoir s'il manque des particules rouges. Si omis, nous pouvons appeler une telle photo une "false positive image"

Inversement, nous inversons l'image binarisée, les particules lipidiques sont la zone blanche, la valeur de pixel est 1 et la valeur de pixel de la zone blanche est 0, puis multiplions l'image avec l'image d'origine pour obtenir une nouvelle image qui ne devrait que ont des particules rouges, de cette façon, nous pouvons voir s'il y a des régions qui ne sont pas des particules lipidiques qui sont identifiées à tort comme des particules lipidiques. S'il y a un faux positif, on peut appeler une telle photo une "false positive image"

Les figures 5-3 et 5-4 montrent les images FN et FP aux seuils de 10 et 35, respectivement.

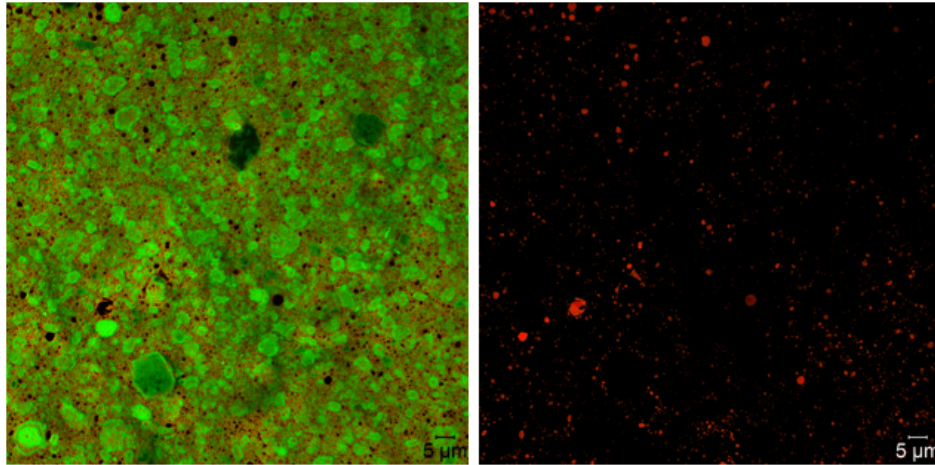


Figure 5.3 – *Threshold10*

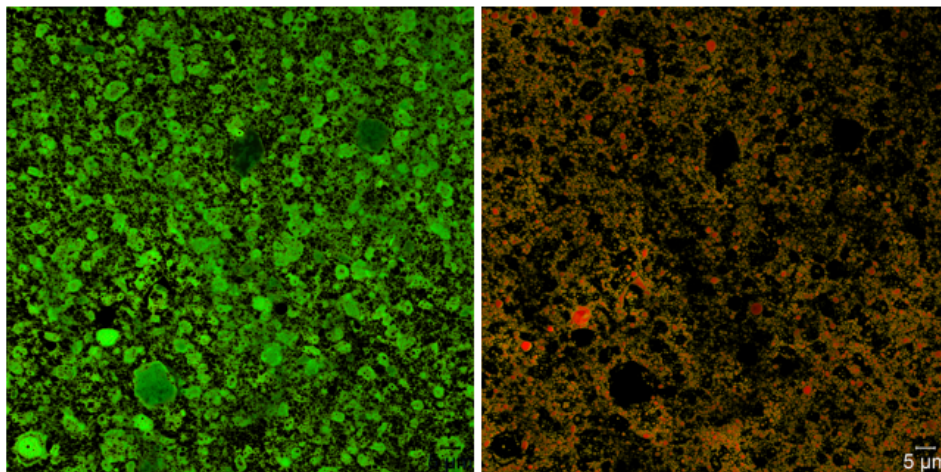
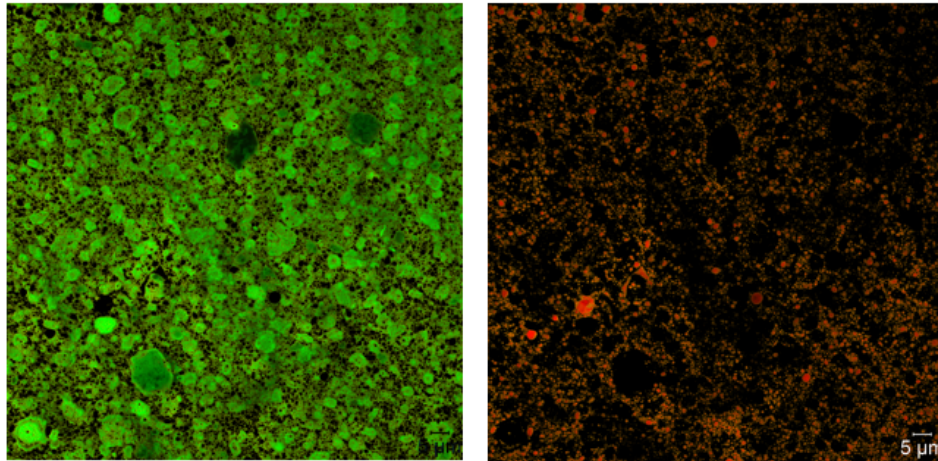


Figure 5.4 – *Threshold35*

Enfin, nous avons généré toutes les images FP et FN avec des seuils de 10 à 35, les avons comparées et avons finalement décidé de binariser les images avec un seuil de 25. Le résultat est dans le figures 5-5 :

### 2.3 Traitement des données

Une fois l'image binarisée, nous obtenons une image de particules segmentées, et l'objectif suivant est d'identifier ces particules noires et de compter leurs informations de caractéristiques.

Figure 5.5 – *Threshold25*

Nous traitons l'image binarisée via l'algorithme d'extraction de contour, obtenons les informations de contour de chaque particule et comptons sa taille de zone, qui est représentée par des pixels. Selon les informations de contour, nous dessinons une boîte englobante dont la longueur et la largeur sont  $w$  et  $h$ , puis nous utilisons  $(h+w)/2$  pour représenter le rayon de la particule. En même temps, nous calculons son centre de masse comme les coordonnées de cette particule. De cette façon, nous obtenons les informations de chaque particule, y compris leur nombre, leur rayon, leur aire et leurs coordonnées.

À partir des informations de toutes les particules dans une seule image, nous pouvons calculer la surface totale des particules, la surface moyenne, l'écart type de la surface, le diamètre moyen, l'écart type du diamètre et le pourcentage de toutes les particules lipidiques dans le image totale.

Enfin, sur la base de ces informations, nous pouvons dessiner un histogramme statistique des particules lipidiques dans une seule image par diamètre ou par zone pour afficher plus intuitivement la teneur en lipides dans l'image.

De même, lorsqu'il s'agit de plusieurs images, nous pouvons calculer toutes les informations ci-dessus.

## 2.4 Interface de logiciel

Afin de faciliter l'utilisation des algorithmes de segmentation et des informations statistiques par les collaborateurs de Bel, nous avons développé une interface logicielle simple pour mettre en œuvre les opérations ci-dessus.

L'interface principale est comme indiqué dans le figures 5-6 :

Les utilisateurs peuvent cliquer sur Importer pour importer un dossier contenant des images à traiter, ce qui importera toutes les images TIFF dans ce dossier. Grâce à "«" et "»", chaque image peut être parcourue. En fonction de la largeur de l'image d'entrée et de la surface minimale des particules que vous souhaitez conserver, "Traiter" peut traiter l'image actuelle et "Exporter" peut exporter l'image actuelle. Les informations sur les particules sont importées dans un fichier csv. Dans le même temps, l'utilisateur peut également mettre en œuvre ces fonctions sur les touches correspondantes de la barre de menus. Avec "ProcessAllImages" et "ExportAllImages", l'utilisateur peut traiter toutes les images et exporter toutes les informations sur les particules.

Lorsque l'image actuelle est traitée, les histogrammes de diamètre et de surface peuvent être dessinés respectivement via "DrawHistOfDias" et "DrawHistOfArea", et différentes plages de taille peuvent être saisies pour le réglage. Les résultats sont dans la figure 5.7.

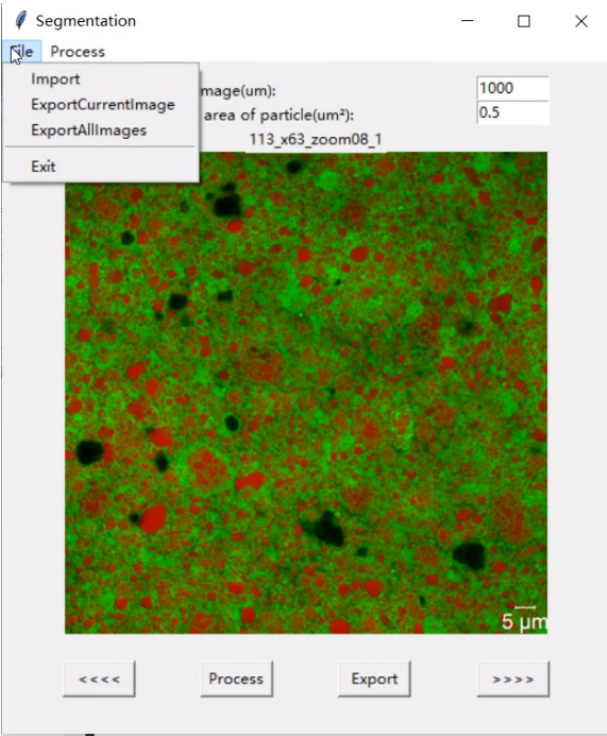


Figure 5.6 – *ViewMain*

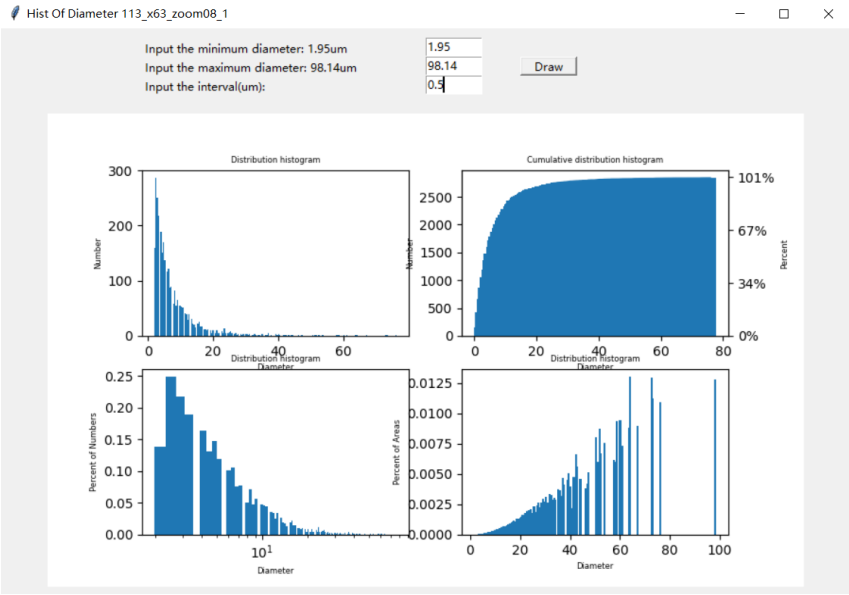


Figure 5.7 – *Histogramme*

# 6

## Bilan et conclusion

### 1 Bilan du semestre 9

Les tâches suivantes ont été accomplies :

- Analyse des besoins
- Lecture et compréhension d'articles liés au projet
- Écrire la partie de la spécification
- Écrire À la pointe de la technologie
- Préparation de la présentation à mi-parcours

En général, le projet s'est bien déroulé au semestre 9. Bien que le temps ait été retardé, les tâches nécessaires ont été accomplies avec succès. En lisant les articles, j'ai une meilleure compréhension de l'algorithme qui doit être implémenté et clarifié le cadre logiciel à implémenter. Au cours du travail, j'ai rencontré le professeur à plusieurs reprises, il m'a donné des indications sur l'orientation de mes recherches et a en même temps répondu à certains de mes doutes.

Dans les jours suivants, je poursuivrai la lecture de l'article, et terminerai la construction de l'algorithme de segmentation, la réalisation de l'interface utilisateur, et compléterai la présentation finale.

### 2 Bilan du semestre 10

Au cours de ce semestre, j'ai essentiellement atteint les objectifs de chaque petite étape. Moi, l'enseignant et le client rencontrerons les équipes toutes les deux semaines pour rendre compte de l'achèvement des tâches de l'étape précédente et formuler les objectifs de l'étape suivante en fonction des besoins du client. Enfin, les fonctions de base requises par ce projet sont réalisées, et les images fournies peuvent être segmentées et les particules lipidiques peuvent être extraites. Cependant, en raison du manque de temps, il existe encore des lacunes dans le logiciel et une optimisation supplémentaire est nécessaire.

De plus, je tiens également à remercier mon mentor, Monsieur Thierry, qui organisera soigneusement pour moi les objectifs de la prochaine étape en anglais lorsque je ne comprendrai pas les attentes du client. En même temps, il me prodiguera patiemment conseils et aide lorsque je rencontrerai des difficultés.

### 3 Bilan sur la qualité

Au cours du semestre S9, grâce à la lecture de littérature, j'ai une meilleure compréhension de la segmentation d'image, je comprends les méthodes de traitement courantes actuelles et j'élargis ma réflexion. Au semestre S10, le traitement des images et le stockage des données sont réalisés en construisant la classe `ProcessImage`, et l'interface interactive est construite via `ViewMain`, qui peut réaliser les fonctions de base. Cependant, en termes de traitement des images, le code peut encore être optimisé pour améliorer la vitesse de traitement. Lors de la mise en œuvre de la segmentation d'image, les particules lipidiques denses ne peuvent pas être divisées de manière très précise. Dans le même temps, lorsque plusieurs images sont comptées, toutes les images sont à nouveau traitées, ce qui réduit l'efficacité.

### 4 Bilan auto-critique

Grâce à ce projet, j'ai d'abord amélioré ma capacité à trouver et à lire la littérature, et en affinant les idées principales de la thèse, j'en ai compris l'essence. Dans la dernière étape du processus de développement, je dois rester en contact constant avec le client, identifier les besoins du client et rendre compte de mon travail, ce qui a exercé mes compétences en communication et en planification.

Mais dans le projet, j'ai aussi découvert mes lacunes. Lors de la lecture de littérature, il est souvent difficile de lire à cause de l'ennui, ce qui distrait et ne sait pas quoi dire. Deuxièmement, mon français doit encore s'améliorer et je ne comprends souvent pas en raison de problèmes de langue lors du processus de communication. Heureusement, avec l'aide de Monsieur Thierry, je peux clairement comprendre les objectifs de la tâche.

# Annexes

# A

# Planification, gestion de projet

## 1 Evolution du projet

Le diagramme de Gantt Initial pour la planification de ce projet

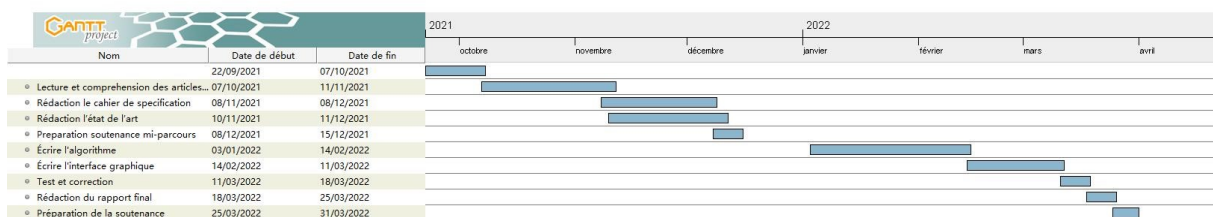


Figure A.1 – Le diagramme de Gantt Final

Le diagramme de Gantt Final de ce projet est comme Figure A.2.

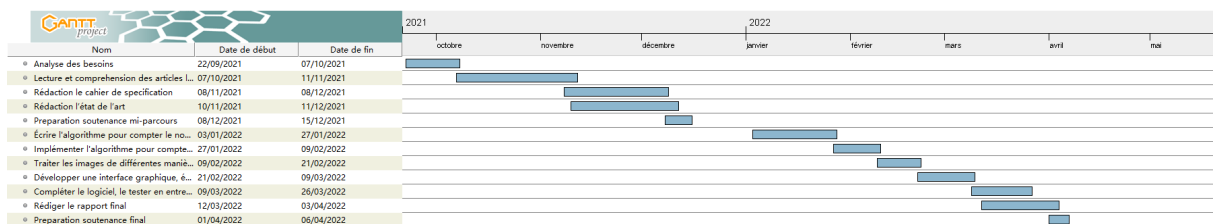


Figure A.2 – Le diagramme de Gantt Final

## 2 Description des tâches

### Tâche 1 : Analyse des besoins

- Date de début : 22/09/2021
- Date de fin : 07/10/2021
- Durée : 16 jours
- Description : Cette tâche comprend le contact avec mon MOA Thierry BROUARD et les clients. Nous avons visité le laboratoire BEL et acquis une compréhension plus intuitive des objectifs du projet. Ensuite, j'ai mené des analyses approfondies et des échanges

avec l'enseignant. Nous pouvons comprendre avec précision les besoins spécifiques des utilisateurs et des projets en termes de fonction, de performance, de fiabilité et d'autres aspects, transformer l'expression des besoins non formalisés des utilisateurs en une définition complète de la demande et déterminer les fonctions que le système doit réaliser.

#### **Tâche 2 : Lecture et compréhension d'articles liés au projet**

- Date de début : 07/10/2021
- Date de fin : 11/11/2021
- Durée : 36 jours
- Description : Ce contenu comprend la lecture d'articles sur la segmentation d'images, le comptage de cellules et d'autres contenus connexes, la compréhension de plusieurs méthodes couramment utilisées, la compréhension de leurs idées et leur mise en œuvre.

#### **Tâche 3 : Rédiger le cahier des charges**

- Date de début : 08/11/2021
- Date de fin : 08/12/2021
- Durée : 31 jours
- Description : Cette tâche comprenait la rédaction d'un cahier des charges. Il comprend la définition du contexte, des objectifs et de sa fonctionnalité.

#### **Tâche 4 : Rédiger l'état d'art**

- Date de début : 10/11/2021
- Date de fin : 11/12/2021
- Durée : 32 jours
- Description : Cette tâche comprenait la rédaction d'un état de l'art basé sur les articles lus précédemment.

#### **Tâche 5 : Préparation de la soutenance à mi-parcours**

- Date de début : 08/12/2021
- Date de fin : 15/12/2021
- Durée : 8 jours
- Description : La tâche consiste à préparer la soutenance à mi-parcours. Cela comprend la finition du rapport, la préparation des diapositives et le tri des idées.

#### **Tâche 6 : Écrire l'algorithme pour compter le nombre de particules dans l'image segmentée**

- Date de début : 03/01/2022
- Date de fin : 25/01/2022
- Durée : 25 jours
- Description : Cette tâche extrait les contours noirs dans les images binarisées fournies et compte le diamètre, les coordonnées de zone et d'autres informations de ces points noirs.

#### **Tâche 7 : Implémenter l'algorithme pour compter plusieurs graphiques en même temps et traiter les données**

- Date de début : 27/01/2022
- Date de fin : 09/02/2022
- Durée : 14 jours
- Description : Cette tâche consiste à traiter plusieurs images en même temps et à dessiner un histogramme basé sur les informations obtenues sur les particules.

### **Tâche 8 : Traiter les images de différentes manières et implémenter la binarisation**

- Date de début : 09/02/2022
- Date de fin : 21/02/2022
- Durée : 13 jours
- Description : Cette tâche implique le traitement des images de différentes manières et l'utilisation de la binarisation pour extraire les particules lipidiques.

### **Tâche 9 : Développer une interface graphique, évaluer l'extraction**

- Date de début : 21/02/2022
- Date de fin : 09/03/2022
- Durée : 17 jours
- Description : Cette tâche comprend le développement d'interfaces graphiques interactives pour l'utilisateur et la proposition et le test de méthodes pour évaluer les résultats des tests.

### **Tâche 10 : Compléter le logiciel, le tester en entreprise**

- Date de début : 09/03/2022
- Date de fin : 26/03/2022
- Durée : 18 jours
- Description : Cette tâche comprenait la mise au point du logiciel et sa livraison à Bel pour tester son bon fonctionnement sur les équipements de l'entreprise.

### **Tâche 11 : Rédiger le rapport final**

- Date de début : 12/03/2022
- Date de fin : 03/04/2022
- Durée : 23 jours
- Description : Cette tâche comprenait la rédaction du rapport final et l'ajout de la section manquante dans S9.

### **Tâche 12 : Préparer la soutenance final**

- Date de début : 01/04/2022
- Date de fin : 06/04/2022
- Durée : 6 jours
- Description : Cette tâche comprenait la préparation d'une présentation finale. Il comprend les diapositives et les résultats de la mise en œuvre.

# B

## Description des interfaces

### 1 Interfaces matérielles/logicielles

Le développement est effectué sous l'environnement Windows et la plate-forme pycharm est utilisée pour le développement python. La partie modèle est réalisée par la bibliothèque opencv, et l'interface graphique est réalisée par la bibliothèque Tkinter.

### 2 Interfaces homme/machine

Pour le client, il peut :

#### Définir la taille des particules de lipides et la largeur de l'image

Cette interface peut permettre à l'utilisateur de définir la taille des particules de lipides qu'il souhaite et la largeur de l'image pour calculer l'échelle.

#### Exporter la liste des particules lipidiques

Cette interface peut permettre l'utilisateur d'exporter la liste des particules lipidiques dans un fichier csv

#### Dessiner un histogramme statistique

Cette interface peut permettre l'utilisateur de dessiner un histogramme statistique basé sur les informations sur les particules



# Cahier de Spécifications

## 1 spécifications Fonctionnelles

### 1.1 Définition de la fonction 1 : ImageProcessing

#### Description de la fonction 1 :

Pour traiter les images :

##### **sharpness\_ImageEnhance**

Entrée : chemin de l'image

Sortie : image

Préconditions : L'image est au format TIFF

Postconditions : L'image est plus nette

##### **process**

Entrée : threshold

Sortie : None

Préconditions : None

Postconditions : L'image est binarisée selon le seuil donné

##### **extract**

Entrée : threshold

Sortie : None

Préconditions : None

Postconditions : Les particules sont marquées.

##### **evaluate**

Entrée : type

Sortie : None

Préconditions : None

Postconditions : Générer les images de FP et FN.

##### **calculateFeatures**

Entrée : None

Sortie : None

Préconditions : L'image est déjà traité.

Postconditions : Les informations sur les particules sont calculées

**export**

Entrée : None

Sortie : None

Préconditions : L'image est déjà traité.

Postconditions : Le fichier est exporté.

**1.2 Définition de la fonction 2 :Interface de logiciel****Description de la fonction 2 :**

Pour générer l'interface interactive

**ImportImage**

Entrée : None

Sortie : None

Préconditions : Ouvrir un dossier

Postconditions : Les images sont importés

**processCurrentImage**

Entrée : threshold

Sortie : None

Préconditions : L'images est importé

Postconditions : L'image est segmenté

**processAllImages**

Entrée : threshold

Sortie : None

Préconditions : Tous les images est importés.

Postconditions : Tous les images sont segmentés.

**exportCurrentImage**

Entrée : type

Sortie : None

Préconditions : L'image est segmenté

Postconditions : Le fichier est exporté.

**exportAllImages**

Entrée : None

Sortie : None

Préconditions : Tous les images sont segmentés.

Postconditions : Le fichier est exporté

**call\_previous**

Entrée : None

Sortie : None

Préconditions : Tous les images est importés.

Postconditions : L'image précédente a été montrée

**call\_next**

Entrée : None

Sortie : None

Préconditions : Tous les images est importés.

Postconditions : L'image suivante est montrée

**DisplayItem**

Entrée : None

Sortie : None

Préconditions : Tous les images est importés.

Postconditions : L'image actuelle s'affiche.

**DrawDia**

Entrée : None

Sortie : None

Préconditions : L'image est déjà traité.

Postconditions : Un histogramme des diamètres des particules de l'image actuelle est tracé.

**DrawArea**

Entrée : None

Sortie : None

Préconditions : L'image est déjà traité.

Postconditions : Un histogramme des surfaces des particules de l'image actuelle est tracé.

## 2 Spécifications non fonctionnelles

### 2.1 Contraintes de développement et conception

Comme il n'y a pas suffisamment de données étiquetées, la segmentation ne peut pas être mise en œuvre à l'aide de l'apprentissage automatique et de l'apprentissage en profondeur. Dans le même temps, pour les résultats d'extraction de particules, en raison du problème de précision après le traitement d'image traditionnel, il ne peut être garanti que l'identification est très précise.

### 2.2 Contraintes de fonctionnement et d'exploitation

#### 2.2.1 Performances

Le résultat attendu est qu'après avoir donné une image, la position, la taille et la quantité des particules lipidiques (points rouges) peuvent être efficacement identifiées. Plus la précision est élevée, plus meilleur.

#### 2.2.2 Capacités

La fonction de segmentation peut être utilisée par n'importe quel utilisateur. Il n'y a pas de limite au nombre d'utilisateurs.

#### 2.2.3 Contrôlabilité

Les utilisateurs peuvent comparer les résultats générés pour déterminer leur exactitude.

#### 2.2.4 Sécurité

Le logiciel n'aura pas accès à d'autres informations, seules les données d'image importées seront traitées.

# D

## Cahier du développeur

### 1 Introduction

Pour ce logiciel, nous avons utilisé deux classes pour l'implémentation. Ce sont respectivement ProcessImage et ViewMain. Parmi eux, ProcessImage est utilisé pour traiter les informations d'image, y compris l'accentuation, le débruitage, la binarisation de l'image et le comptage des informations sur les particules qu'elle contient. ViewMain est utilisé pour construire une interface graphique, compléter les fonctions de base requises en appelant les méthodes appropriées dans la classe ProcessImage, dessiner des histogrammes et exporter des fichiers csv en même temps.

### 2 Diagrammes architecturaux et UML

Le diagramme de classes est le suivant

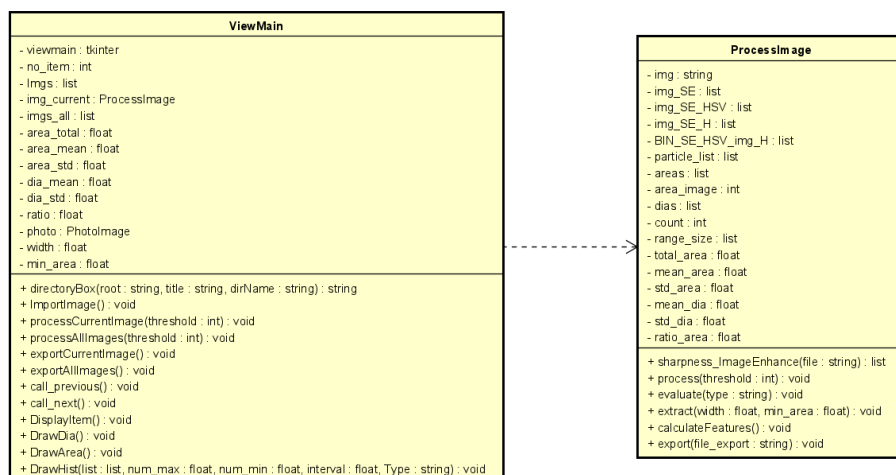


Figure D.1 – class diagram

Les capacités de l'utilisateur est le suivant :

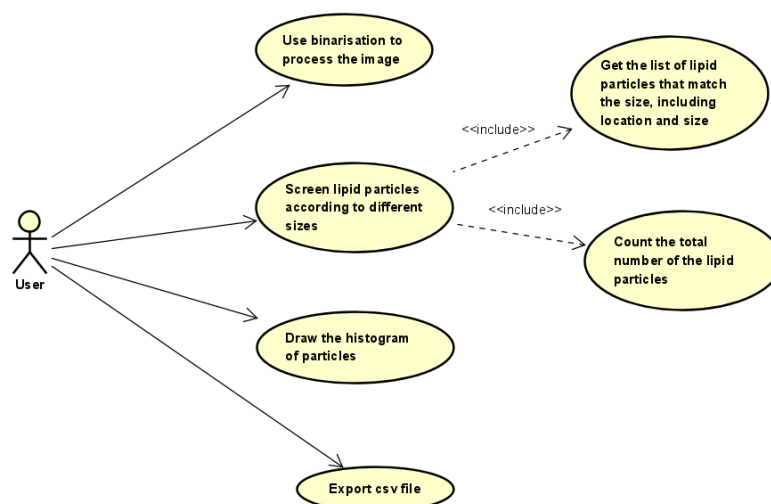


Figure D.2 – Cas d'utilisation

Le flux de son utilisation du logiciel est le suivant :

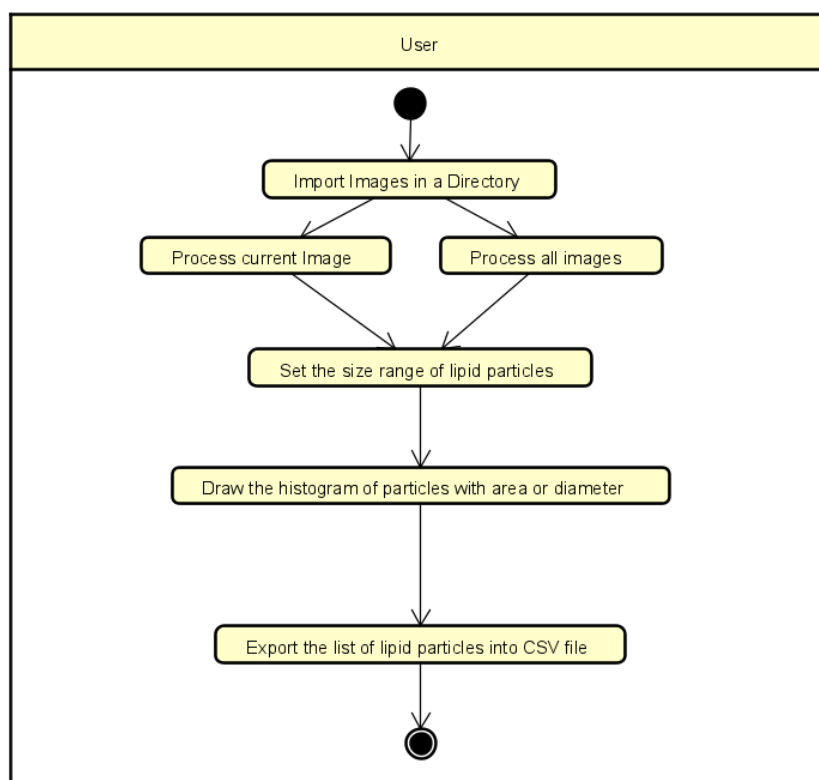


Figure D.3 – Activity

### 3 Descriptions détaillées de données exploitées

L'analyse repose sur des images. D'une taille de  $2048 \times 2048$  pixels, au format TIFF, elles permettent d'identifier les éléments étudiés selon la fluorescence de leur constituants en réponse

à une excitation procurée par un laser. Les deux couleurs dominantes sont le rouge, associé aux matières grasses, et le vert, associé aux protéines.

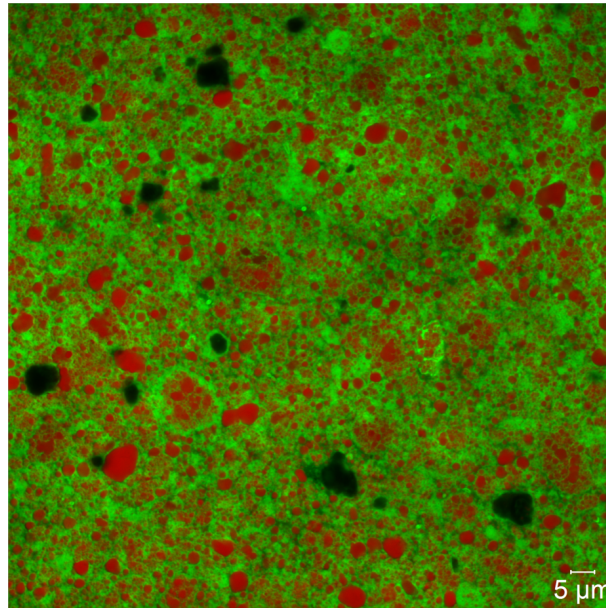


Figure D.4 – Données Exploitées

## 4 Descriptions détaillées des classes, modules, réalisations

### 4.1 Classe "ProcessImage"

#### 4.1.1 Propriété

- `img` : Le chemin de l'image entrante
- `img_SE` : Image améliorée de netteté
- `img_SE_HSV` : Image de la chaîne HSV
- `img_SE_HSV_H` : Image du canal H
- `BIN_SE_HSV_img_H` : image après binarisation
- `threshold` : Seuil de binarisation
- `particle_list` : Liste des informations sur les particules dans l'image
- `areas` : Une liste des aires de chaque particule
- `area_image` : Surface de l'image
- `dias` : Liste des diamètres de particules
- `count` : Nombre de photos
- `range_size` : Liste des gammes de diamètres et de surfaces
- `total_area` : Surface totale des particules
- `mean_area` : Valeur moyenne de la surface des particules dans l'image
- `std_area` : Écart type de la surface des particules dans l'image
- `mean_dia` : Valeur moyenne des diamètres de particules dans l'image
- `std_dia` : Écart type des diamètres de particules dans l'image
- `ratio_area` : Surface totale des particules en pourcentage de la surface de l'image

### 4.1.2 Méthode

- `sharpness_ImageEnhance()` : Méthodes pour améliorer la netteté
- `process()` : Binariser l'image
- `evaluate()` : Évaluer les résultats de la binarisation avec FN et FP
- `extract()` : Extraire les informations sur les particules dans l'image binarisée
- `calculateFeatures()` : Calculer les caractéristiques globales des particules dans l'image
- `export()` : Exporter les informations sur les particules dans l'image vers un fichier csv

## 4.2 Classe "ViewMain"

### 4.2.1 Propriété

- `viewmain` : Fenêtre principale
- `no_item` : Le numéro de l'image actuelle
- `Imgs` : Toutes les images du répertoire
- `img_current` : Classe `ProcessImage` pour l'image courante
- `imgs_all` : Liste de classe `ProcessImage` de toutes les images
- `area_total` : Superficie totale des particules dans toutes les images
- `area_mean` : Surface moyenne des particules dans toutes les images
- `area_std` : Écart type de la surface des particules sur toutes les images
- `dia_mean` : Moyenne des diamètres de particules dans toutes les images
- `dia_std` : Écart type des diamètres de particules sur toutes les images
- `ratio` : Surface totale des particules en pourcentage de la surface de toutes les images
- `photo` : Image affichée dans Canvas
- `width` : La largeur réelle de l'image
- `min_area` : la zone de la plus petite particule que vous souhaitez conserver

### 4.2.2 Méthode

- `directoryBox()` : Obtenir le chemin du répertoire cible
- `ImportImage()` : Importer les images dans le répertoire
- `processCurrentImage()` : Traiter l'image actuelle, le binariseur et calculer les caractéristiques
- `processAllImages()` : Traiter toutes les images du répertoire courant
- `exportCurrentImage()` : Exporter les informations de l'image courante vers le fichier csv avec le nom de l'image courante
- `exportAllImages()` : Exporter les informations de toutes les images vers le fichier csv avec le nom de l'image actuelle
- `call_previous()` : Afficher l'image précédente dans le répertoire
- `call_next()` : Afficher l'image suivante dans le répertoire
- `DisplayItem()` : Afficher l'image en cours dans le répertoire
- `DrawDia()` : Dessine l'histogramme du diamètre de l'image courante dans une nouvelle fenêtre
- `DrawArea()` : Dessine l'histogramme du diamètre de l'image courante dans une nouvelle fenêtre
- `DrawHist()` : dessiner un histogramme

# E

## Document d'installation

L'installation du logiciel est simple. Au moment de la livraison, nous utilisons pyinstaller pour emballer le projet et générer un fichier exe exécutable. Pour les utilisateurs, il suffit de télécharger le fichier exe, et les fonctions associées du programme peuvent être utilisées normalement sous Windows.

# F

## Document d'utilisation

Lorsque l'utilisateur ouvre le logiciel, l'interface principale est la suivante, par « Importer » sous la barre de menu « Fichier », l'utilisateur peut importer le dossier contenant les images à traiter. image actuelle.

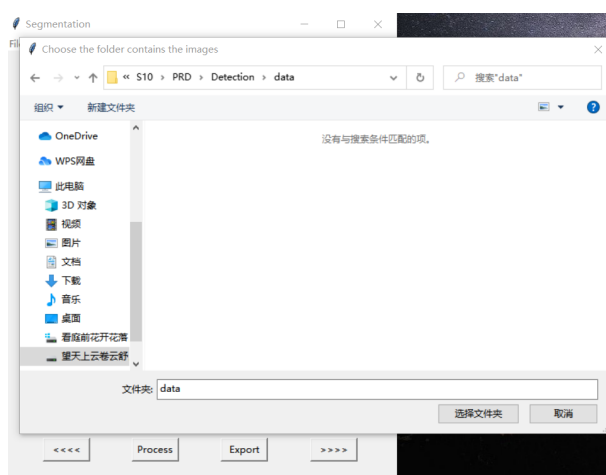


Figure F.1 – Importer

Sous l'image actuelle, nous pouvons entrer la largeur réelle de l'image pour déterminer l'échelle et entrer la zone de la plus petite particule que nous voulons conserver. Après avoir déterminé ces deux hyperparamètres, on peut cliquer sur "Process" pour segmenter l'image courante afin d'obtenir les particules lipidiques qu'elle contient. Une fois le traitement terminé, la boîte de dialogue suivante apparaîtra. Ensuite, cliquez sur "Export", nous pouvons exporter les informations de toutes les particules lipidiques de l'image actuelle sous forme de fichier csv sous le répertoire de l'image à enregistrer.

En parallèle, dans le menu déroulant "Process" de la barre de menu, on peut tracer des histogrammes statistiques pertinents grâce à « DrawHistOfDias » et « DrawHistOfAreas ». Voici un exemple de dessin d'un histogramme de diamètre (il en va de même pour l'aire). Cliquez sur ce bouton et une nouvelle fenêtre apparaîtra avec un titre indiquant quelle image est l'histogramme.

Selon l'invite, nous pouvons entrer la plage de taille des particules que nous voulons dessiner l'histogramme. Les diamètres minimum et maximum ont été donnés par défaut, et l'unité est um. En même temps, en entrant l'espacement, nous pouvons déterminer le nombre de classes dans

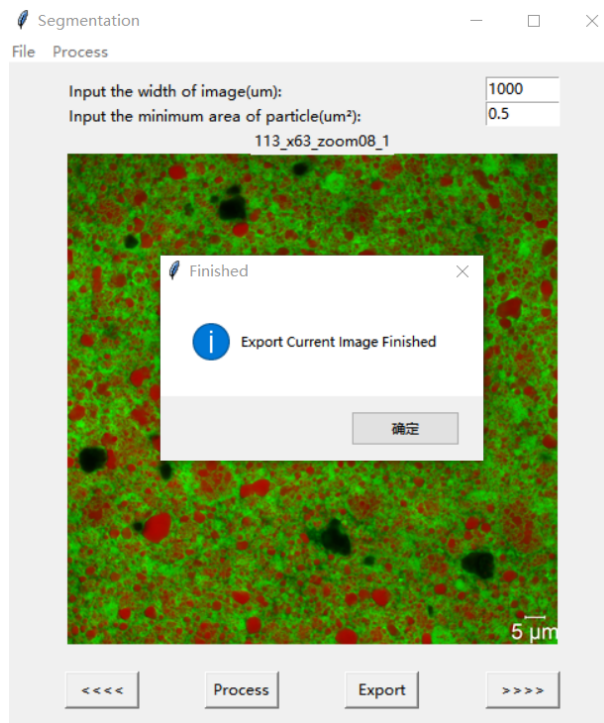


Figure F.2 – Exporter

l'histogramme.

Cliquez sur Dessiner pour dessiner comme indiqué sur la figure. À ce stade, vous pouvez dessiner différents histogrammes en modifiant les paramètres.

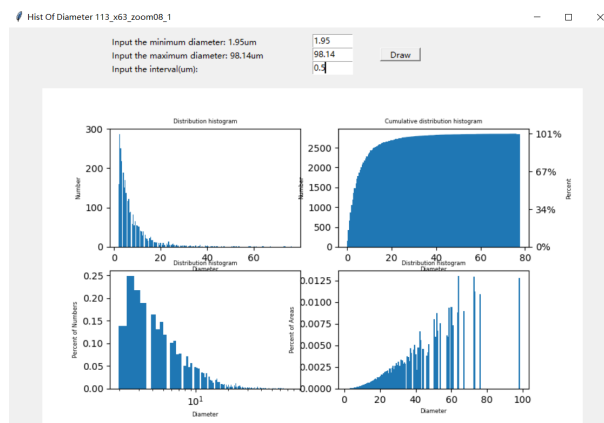


Figure F.3 – Dessiner l'histogramme

Outre le traitement de l'image actuelle, le logiciel offre également la fonction de traitement de toutes les images en même temps. En cliquant sur "Traiter toutes les images", vous pouvez effectuer une segmentation sur toutes les images importées, et en cliquant sur "Exporter toutes les images", vous pouvez calculer les informations pertinentes des particules de papier dans toutes les images et les exporter dans des fichiers csv. Une fois terminé, la boîte de dialogue suivante apparaîtra.

Si vous cliquez sur exporter ou dessinez un histogramme avant de traiter l'image, une boîte de message apparaîtra pour vous rappeler de le traiter en premier.

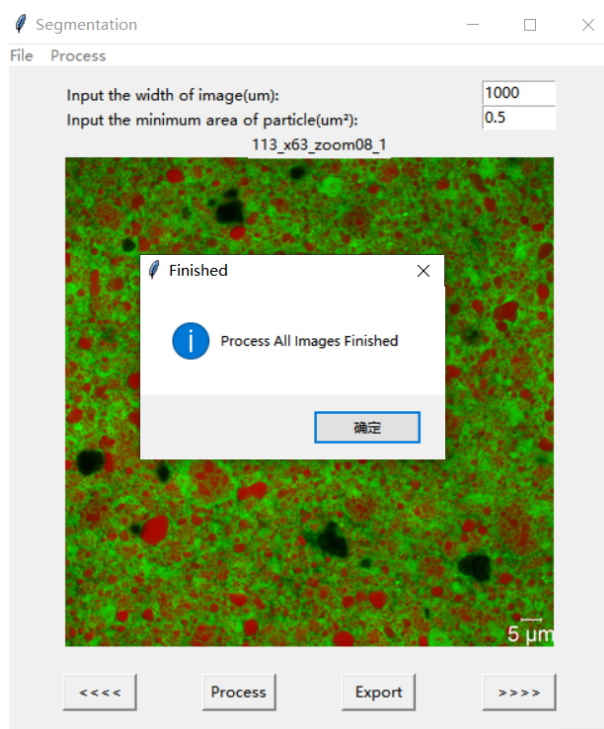


Figure F.4 – Traiter tous les images

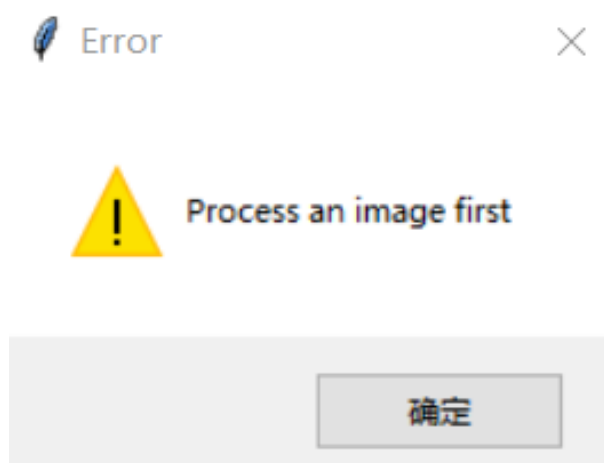


Figure F.5 – Erreur

Le logiciel traite principalement des images, il est donc testé en générant les images obtenues à chaque étape.

## 1 Tests fonctionnel

Les images obtenues après traitement sont les suivantes.

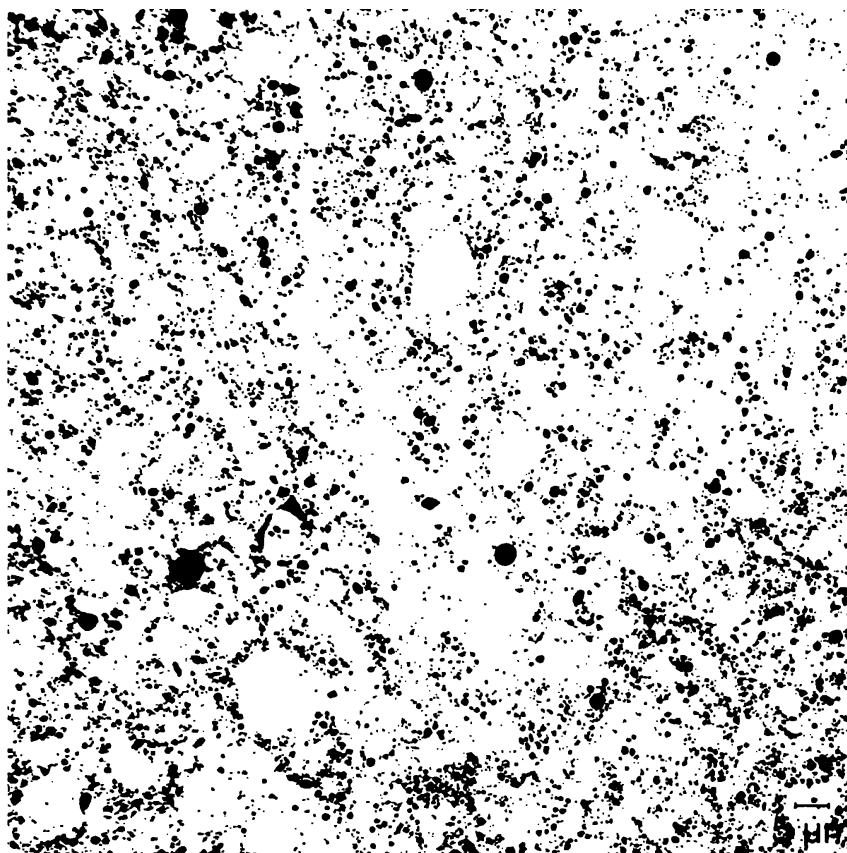


Figure G.1 – *Binarisation*

Les graphiques des résultats extraits sont les suivants.

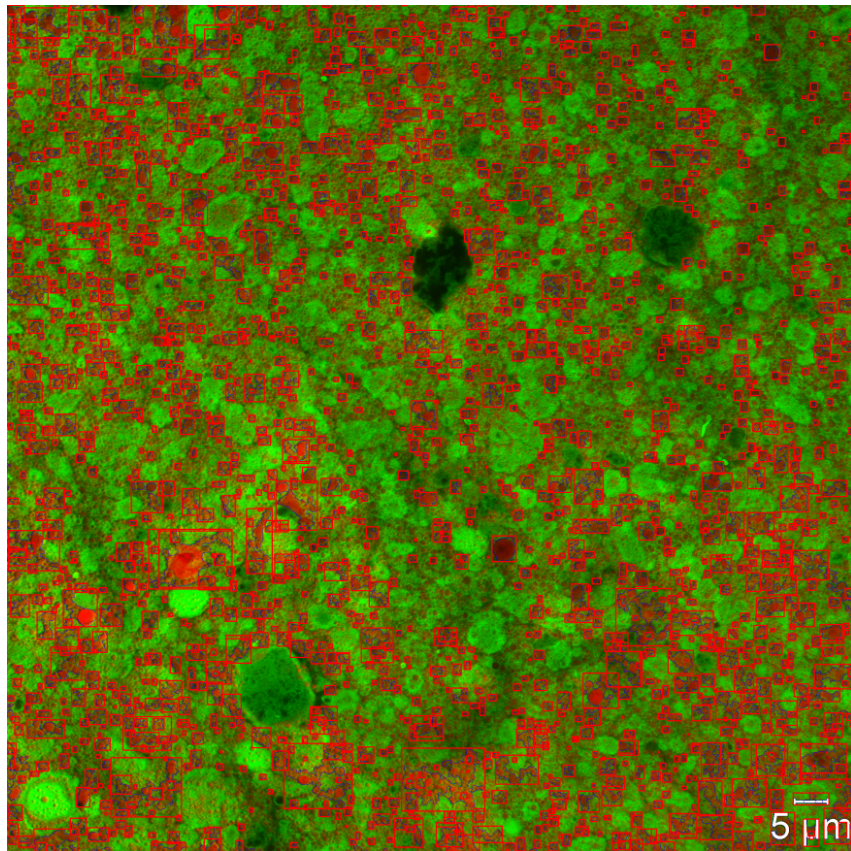


Figure G.2 – *Résultat*

- [1] Dev Kumar DAS, Madhumala GHOSH, Mallika PAL, Asok K MAITI et Chandan CHAKRABORTY. « Machine learning approach for automated screening of malaria parasite using light microscopic images ». In : *Micron* 45 (2013), p. 97-106.
- [2] Carlos X HERNÁNDEZ, Mohammad M SULTAN et Vijay S PANDE. « Using deep learning for segmentation and counting within microscopy data ». In : *arXiv preprint arXiv :1802.10548* (2018).
- [3] Gurbakash PHONSA et K. MANU. « A Survey : Image Segmentation Techniques : Theory and Applications, ICHSA 2018 ». In : jan. 2019, p. 1123-1140. ISBN : 978-981-13-0760-7. DOI : [10.1007/978-981-13-0761-4\\_105](https://doi.org/10.1007/978-981-13-0761-4_105).
- [4] J. SAUVOLA et M. PIETIKÄINEN. « Adaptive document image binarization ». In : *Pattern Recognition* 33.2 (2000), p. 225-236. ISSN : 0031-3203. DOI : [https://doi.org/10.1016/S0031-3203\(99\)00055-2](https://doi.org/10.1016/S0031-3203(99)00055-2). URL : <https://www.sciencedirect.com/science/article/pii/S0031320399000552>.
- [5] Yao XUE et Nilanjan RAY. « Cell detection in microscopy images with deep convolutional neural network and compressed sensing ». In : *arXiv preprint arXiv :1708.03307* (2017).

# Segmentation/méetrologie d'images dans l'agroalimentaire : Détecter les particules lipidiques dans l'image

Jiyuan WANG

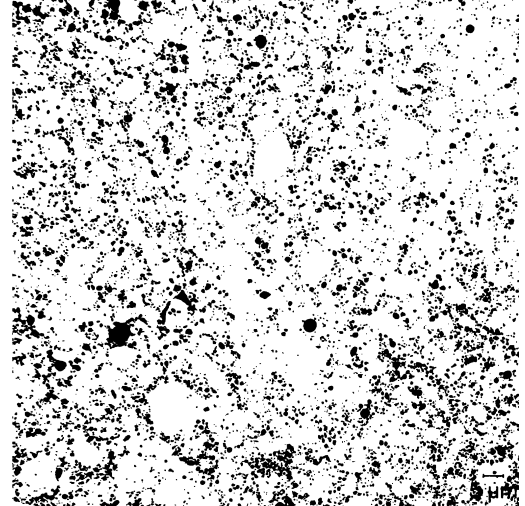
Encadrement : Thierry BROUARD



En collaboration avec Polytech

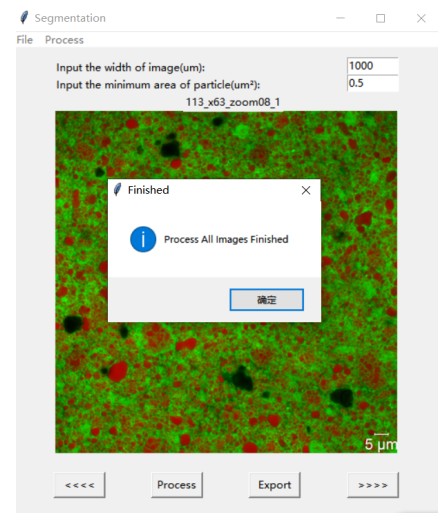
## Objectifs

- Segmentation
- Particules Lipidiques
- GUI



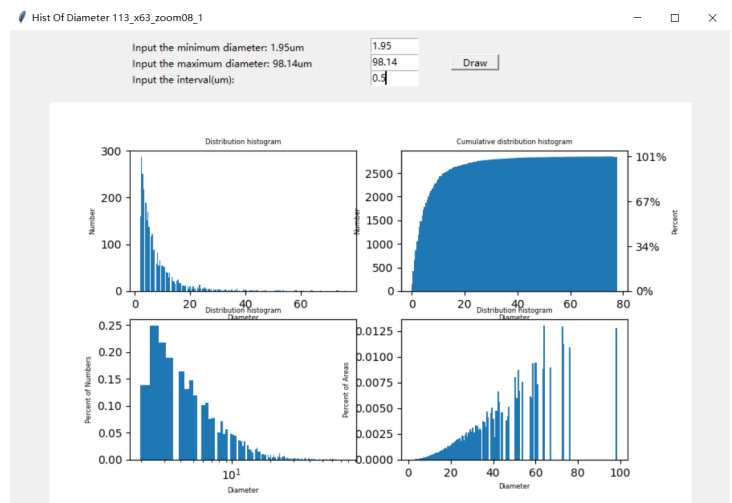
## Mise en œuvre

1. Binarisation
2. Interface
3. Histogramme



## Résultats attendus

Segmentation des particules lipidiques en images. Interface graphique simple et facile à utiliser.



# Segmentation/méetrologie d'images dans l'agroalimentaire : Détecter les particules lipidiques dans l'image

Jiuyan WANG

Encadrement : Thierry BROUARD



En collaboration avec Polytech

## Objectifs

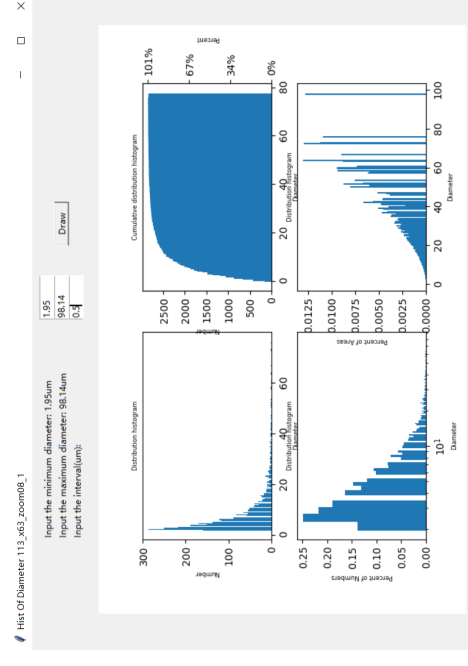
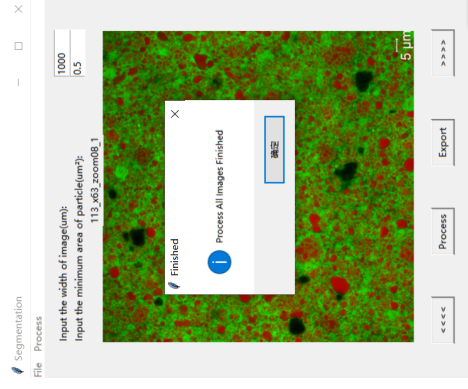
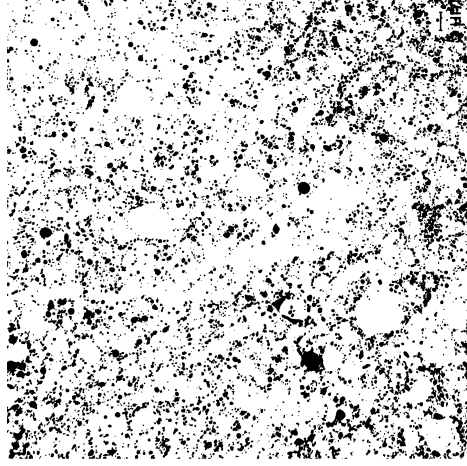
- Segmentation
- Particules Lipidiques
- GUI

## Mise en œuvre

1. Binarisation
2. Interface
3. Histogramme

## Résultats attendus

Segmentation des particules lipidiques en images. Interface graphique simple et facile à utiliser.



# Segmentation/métrologie d'images dans l'agroalimentaire

## Détecter les particules lipidiques dans l'image

### Résumé

Cet article, en collaboration avec BEL Corporation, a développé un logiciel pour détecter la teneur en particules lipidiques dans les images. Tout d'abord, les particules lipidiques de l'image sont extraites et segmentées par binarisation, puis les informations telles que leur nombre, leur taille et leur position sont comptées, et enfin un histogramme statistique est dessiné et un fichier csv est exporté.

### Mots-clés

particules lipidiques, binarisation, segmentation

### Abstract

This article, in collaboration with BEL Corporation, developed a software for detecting lipid particle content in images. Firstly, the lipid particles in the picture are extracted and segmented by binarization, and then the information such as their number, size, and position are counted, and finally a statistical histogram is drawn and a csv file is exported.

### Keywords

lipid particle, binarization, segmentation

Entreprise

Polytech



Tuteur entreprise

Hernan KENMOGNE

Étudiant

Jiyuan WANG (DI5)

Tuteur académique

Thierry BROUARD