



HAL
open science

Segmentation d'images de microscopie électronique par apprentissage profond pour l'analyse quantitative de l'ultrastructure cellulaire

Cyril Meyer

► **To cite this version:**

Cyril Meyer. Segmentation d'images de microscopie électronique par apprentissage profond pour l'analyse quantitative de l'ultrastructure cellulaire. Apprentissage [cs.LG]. Université de Strasbourg, 2023. Français. NNT : 2023STRAD045 . tel-04473581

HAL Id: tel-04473581

<https://theses.hal.science/tel-04473581>

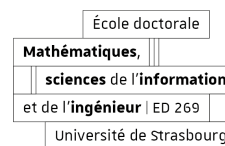
Submitted on 22 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université de Strasbourg



École doctorale : **Mathématiques, Sciences de l'Information et de l'Ingénieur**
ED 269

Unité de recherche : **ICube - CNRS UMR 7357** et **IGBMC - CNRS UMR 7104**
- Inserm U 1258

THÈSE

Présentée par : **Cyril Meyer**

Soutenue le : **17 novembre 2023**

Pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/S spécialité : **Informatique**

Segmentation d'images de microscopie électronique par apprentissage profond pour l'analyse quantitative de l'ultrastructure cellulaire

Thèse dirigée par :

NAEGEL Benoît Professeur, Université de Strasbourg
SCHULTZ Patrick Directeur de recherche, CNRS

Encadrant :

BAUDRIER Etienne Maître de conférences HDR, Université de Strasbourg

Rapporteurs :

PERRET Benjamin Professeur, ESIEE Paris, Université Gustave Eiffel
RUAN Su Professeure, Université de Rouen

Examineur :

PASSAT Nicolas Professeur, Université de Reims Champagne-Ardenne

Remerciement

Je remercie mon directeur de thèse, Benoît Naegel, pour la confiance qu'il m'a accordée, sa disponibilité, ses conseils avisés ainsi que son soutien tout au long de ce travail. Merci également à Patrick Schultz, mon co-directeur pour m'avoir épauler tous le long de la thèse, pour ses encouragements, sa bonne humeur communicative et pour m'avoir fait découvrir un nouveau domaine au travers de la microscopie électronique cellulaire. Merci à mon encadrant, Etienne Baudrier, pour son investissement, sa franchise, ses remarques pertinentes sur nos travaux et pour avoir toujours été à l'écoute afin de m'aiguiller au mieux. Je n'oublie pas Véronique Mallouh qui m'a permis d'appréhender nos problématiques au plus proche du monde réel, ainsi que d'avoir passé de très nombreuses heures assise derrière sa tablette graphique pour nous fournir des segmentations de référence de qualité.

Je remercie également les membres du jury : Benjamin Perret, Su Ruan et Nicolas Passat qui ont accepté de juger mon travail.

Merci à tous mes collègues de l'équipe IMAGEs du laboratoire ICube, ainsi qu'à ceux du département informatique de l'IUT Robert Schuman d'Illkirch, qui m'ont offert l'opportunité de travailler dans une superbe ambiance et dans d'excellentes conditions. En particulier un grand merci à Etienne Le Quentrec, Guillaume Decor et Eleonore Dufresne.

Je remercie l'Université de Strasbourg et en particulier le programme doctoral international pour avoir financé ce travail.

Mes remerciements vont également à Franck Velikonia, mon professeur de mathématiques au lycée, qui a réussi à me motiver et à me donner confiance en mes capacités en exploitant au mieux ma passion pour l'informatique et la programmation.

Merci à tous mes amis, Thibaut Peyret, Kevin Masson, Yann Guth, Tom Hammer, Victor Beaucourt, Julien Hetzlen et tous ceux que je ne cite pas, mais que je n'oublie pas néanmoins. En particulier, je tiens à remercier Paul Viville, avec qui nous avons même collaboré sur un outil présenté dans cette thèse.

Je remercie ma famille, pour leur encouragement, en particulier mes parents, pour m'avoir permis d'étudier pendant toutes ces années dans les meilleures conditions possibles.

Enfin, je remercie Aurélie, ma compagne, pour son soutien, son affection et pour m'avoir trouvé un compagnon d'insomnie, Yuki, aussi appelé Monsieur le Chat.

Table des matières

Sigles	ix
Table des figures	xi
Liste des tableaux	xv
Résumé	xvii
1 Introduction	1
1.1 Contexte	2
1.1.1 Imagerie biomédicale cellulaire et microscopie électronique FIB-SEM	2
1.1.2 Analyse d'images de microscopie électronique cellulaire	6
1.1.3 Apprentissage profond et méthodes supervisées	8
1.2 Problématique	11
1.3 Objectifs et contributions	12
1.4 Organisation du manuscrit	13
2 État de l'art	15
2.1 Segmentation d'images par apprentissage profond	15
2.1.1 Biomédical et microscopie électronique cellulaire	18
2.1.2 Segmentation interactive d'images	25
2.2 Apprentissage profond et morphologie mathématique	30
2.2.1 Les réseaux de neurones morphologiques	30
2.2.2 Application de la morphologie mathématique en apprentissage profond	32
3 Données et évaluation	37
3.1 Données	37
3.1.1 Données internes	41
3.1.2 Données publiques	50
3.2 Évaluation	58

3.2.1	Mesure de recouvrement	58
3.2.2	Mesures de distance aux contours	61
3.2.3	Mesure de composantes connexes	62
3.2.4	Comparaison des mesures	62
4	Apprentissage profond et segmentation	69
4.1	Protocole expérimental	70
4.2	Segmentation automatique avec annotations denses	73
4.2.1	Modèles multi-classes et binaires	73
4.2.2	Modèles 2D et 3D	76
4.2.3	Variations du modèle U-Net	80
4.2.4	Taille de patch 2D pour l'entraînement	84
4.2.5	Taille du jeu d'entraînement	86
4.2.6	Fonction de perte	87
4.2.7	Modèle pré-entraîné	90
4.2.8	Adaptation de domaine et CycleGAN	90
4.2.9	Premières conclusions	94
4.3	Segmentation interactive et annotations faibles	95
4.3.1	Segmentation interactive et simulation d'interaction	95
4.3.2	Interactions via interface homme machine	104
4.3.3	Annotations faibles	105
4.3.4	Résultats de segmentation avec annotations faibles	108
4.3.5	Conclusion	109
4.4	Morphologie mathématique et apprentissage profond	111
4.4.1	Opérateurs morphologiques	111
4.4.2	Pré-traitements des images	113
4.4.3	Réseaux de neurones et couches morphologiques	117
4.4.4	Structures hiérarchiques et opérateurs connexes	119
4.4.5	Pré-traitement des images par opérateurs connexes	123
4.4.6	Pré-traitement des images par extraction d'attributs	125
4.4.7	Conclusion	132
5	Conclusion	133
5.1	Bilan	133
5.2	Perspectives	135
5.2.1	Annotations faibles et apprentissage profond	135
5.2.2	Représentation hiérarchique et apprentissage machine	136
5.2.3	Applications aux images non FIB-SEM	136
5.2.4	OpenData et base de données d'imagerie électronique cellulaire	137
5.2.5	Analyse morphologique de l'ultrastructure cellulaire	137

A Annexe	139
A.1 Productions logicielles	140
A.1.1 Segmentation d'images de microscopie électronique	140
A.1.2 Segmentation interactive et annotations faibles	143
A.1.3 Morphologie mathématique	145
A.1.4 Visualisation	147
A.1.5 Divers	148
A.1.6 Résumé des applications	151
A.2 Jeux de données	152
A.2.1 I3-ISBI et LW4-ISBI	152
A.2.2 IGBMC-S-multi	152
A.2.3 IGBMC-620-MITO	152
A.2.4 I3-ISO	153
A.2.5 LW4-ISO	153
A.2.6 JG1-Mito 5x384	153
A.2.7 MitoEM-H	154
A.3 Images FIB-SEM IGBMC	155
A.4 Figures	161
Bibliographie	163

Sigles

CNN	Réseau de neurones convolutif (Convolutional Neural Network)
FCN	Réseau entièrement convolutif (Fully Convolutional Network)
GAN	Réseau antagoniste génératif (Generative Adversarial Network)
SGD	Algorithme du gradient stochastique (Stochastic Gradient Descent)
ASSD	Distance de surface symétrique moyenne (Average Symmetric Surface Distance)
DSC	Indice de Sørensen-Dice / Coefficient de Sørensen-Dice (Dice Similarity Coefficient)
EM	Microscopie électronique (Electron Microscopy)
SEM / MEB	Microscopie électronique à balayage (Scanning Electron Microscopy)
TEM / MET	Microscopie électronique en transmission (Transmission Electron Microscopy)
FIB-SEM	Microscopie électronique à balayage et faisceau d'ions focalisé (Focused Ion Beam Scanning Electron Microscopy)
ER	Réticulum endoplasmique (Endoplasmic Reticulum)

Table des figures

1.1	Comparaison schématique simplifiée des résolutions.	2
1.2	Images de microscopie électronique.	4
1.3	Principe de fonctionnement d'un microscope FIB-SEM.	5
1.4	Exemples d'images FIB-SEM cellulaires.	5
1.5	Différents types de segmentations.	6
1.6	Exemple d'images FIB-SEM cellulaires avec segmentation sémantique.	7
1.7	Imbrication du domaine de l'apprentissage profond.	8
1.8	Fréquence d'utilisation des différents noms, Google Books Ngram Viewer.	9
1.9	Comparaison des résultats de la compétition ILSVRC.	10
2.1	Un modèle FCN [Long et al., 2015].	16
2.2	Convolution et convolution à trous [Dumoulin and Visin, 2016].	17
2.3	Chronologie des architectures de segmentation d'images.	18
2.4	Le modèle U-Net [Ronneberger et al., 2015].	19
2.5	Le modèle présenté par [Xiao et al., 2018].	20
2.6	Données d'entraînements du projet OpenOrganelle	21
2.7	Les architectures U-Net comparées par [Kugelman et al., 2022].	22
2.8	Exemple des différents types d'interactions [Ramadan et al., 2020].	26
2.9	Framework proposé par [Xu et al., 2016].	26
2.10	Résultats de segmentation de la méthode DEXTR [Maninis et al., 2018].	27
2.11	DistMap-BRS [Jang and Kim, 2019] et f-BRS [Sofiuk et al., 2020].	28
2.12	Aperçu de la méthode interactive proposée par [Wang et al., 2019].	30
2.13	Exemple d'apprentissage d'un opérateur de fermeture et d'ouverture.	33
2.14	Éléments structurants appris pour des couches PConv, \mathcal{L} Morph et \mathcal{S} Morph.	34
2.15	Framework proposé par Farfan Cabrera et al. [Farfan Cabrera et al., 2021b].	35
3.1	Représentation schématique des différentes méthodes d'acquisition.	39
3.2	Exemples d'images de coupes issues de l'image $I3$ et $LW4-1$	42
3.3	Exemples d'images de coupes issues de l'échantillon $LW4$	43
3.4	Exemples d'images de coupes issues de l'échantillon $LW2$	43

3.5	Exemple d’annotations pour les images $I3$ et $LW4-1$	44
3.6	Représentation 3D des mitochondries pour l’image $I3$	45
3.7	Répartitions des tranches annotées pour $I3$ et $LW4-1$	45
3.8	Exemple d’annotations pour l’image $LW4-2$	46
3.9	Exemple d’une tranche de l’image $JG1$	47
3.10	Exemples de tranches des images GoetzLab.	48
3.11	Représentation 3D du cinquième bloc annoté de l’image $JG1$	50
3.12	Outil de visualisation en ligne des données OpenOrganelle.	51
3.13	Exemple d’un bloc d’entraînements OpenOrganelle [Heinrich et al., 2020].	52
3.14	Visualisation des jeux de données MitoEM [Wei et al., 2020].	53
3.15	Exemple d’image TEM ISBI 2012 et sa segmentation correspondante.	54
3.16	Exemple d’une tranche d’entraînement du jeu de données EPFL Lucchi.	54
3.17	Exemple d’une tranche d’entraînement du jeu de données Kasthuri 2015.	55
3.18	Exemple d’une tranche du jeu de données CREMI.	55
3.19	Exemple d’une tranche d’entraînement du jeu de données 3D Platelet EM.	56
3.20	Représentation de l’utilisation des métriques pour la segmentation.	59
3.21	Résumé schématique des notions d’erreurs des mesures de recouvrement.	59
3.22	Exemple de travail : F1-score similaire.	63
3.23	Exemple travail : ASSD similaire.	63
3.24	Transformations de la segmentation de référence par érosion.	64
3.25	Transformations de la segmentation de référence par dilatation.	64
3.26	Transformations de la segmentation de référence par décalage.	65
3.27	Transformations élastiques de la segmentation de référence.	65
3.28	Exemple des transformations appliquées.	66
4.1	Résultats de l’expérience binaire et multi-classes.	75
4.2	Résultats de l’expérience 2D et 3D, mitochondrie, L_{Dice}	77
4.3	Résultats de l’expérience 2D et 3D, mitochondrie, L_{Dist}	78
4.4	Résultats de l’expérience 2D et 3D, réticulum endoplasmique, L_{Dice}	78
4.5	Résultats de l’expérience 2D et 3D, réticulum endoplasmique, L_{Dist}	79
4.6	Résultats de l’expérience 2D et 3D, bilan des scores médians.	79
4.7	Représentation schématique de l’architecture U-Net.	80
4.8	Représentation schématique des blocs composant l’architecture U-Net.	81
4.9	Surfaces de loss avec/sans connexions résiduelles [Li et al., 2018a].	82
4.10	Résultats de l’expérience de variations du modèle U-Net, ASSD.	82
4.11	Résultats de l’expérience de variations du modèle U-Net, F1 Score.	83
4.12	Résultats de l’expérience sur les tailles de patch, exactitude maximale.	85
4.13	Résultat de l’expérience sur la taille du jeu d’entraînement $I3-ISO$	86
4.14	Résultats de l’expérience sur la taille du jeu d’entraînement $JG1-Mito$	87
4.15	Résultats de l’expérience de choix de loss en 2D, F1 Score.	88
4.16	Résultats de l’expérience de choix de loss en 3D, F1 Score.	88
4.17	Comparaisons des fonctions de pertes pour modèles 2D et 3D sur 4 classes.	89
4.18	Résultats de l’expérience sur les modèles pré-entraînés.	91

4.19	Architecture de la méthode CycleGAN [Zhu et al., 2017].	92
4.20	Expérimentation CycleGAN.	93
4.21	Représentation du processus itératif de segmentation interactive.	97
4.22	Différents types d'interactions pour la classe des mitochondries.	98
4.23	Résultat de l'expérience de segmentation interactive, <i>I3</i> (1 à 16).	98
4.24	Résultat de l'expérience de segmentation interactive, <i>I3</i> (32 à 200).	99
4.25	Exemples de coupes des blocs d'entraînement et de test pour l'image <i>JG1</i>	100
4.26	Résultats de l'expérience de segmentation interactive, <i>JG1</i> , automatique.	101
4.27	Résultats de l'expérience de segmentation interactive, <i>JG1</i> , « DEXTR ».	101
4.28	Résultats de l'expérience de segmentation interactive, <i>JG1</i> , « Fill ».	102
4.29	Résultats de l'expérience de segmentation interactive, <i>JG1</i> , « Fill » dense.	102
4.30	Modèles automatiques, interactifs et entraînés sur des résultats interactifs.	103
4.31	Démonstration de l'outil de segmentation cutter	104
4.32	Exemple d'annotations pour l'image <i>LW4-2</i>	105
4.33	Segmentation utilisant un CNN entraîné sur des annotations faibles.	106
4.34	Simulation des annotations faibles par squelettisation.	107
4.35	Comparaisons entre modèles entraînés par annotations denses ou faibles.	107
4.36	F1-Score sur les 5 coupes de tests pour les 5 classes étudiées.	108
4.37	Une coupe de segmentation de référence de l'image <i>LW4-2</i>	109
4.38	Prédiction d'une coupe de l'image <i>LW4-2</i>	109
4.39	Exemples de coupes segmentées en utilisant SAM [Kirillov et al., 2023].	110
4.40	Exemple de pré-traitement d'une coupe de l'image <i>I3</i>	114
4.41	Exactitude avec un pré-traitement d'ouverture ou de fermeture.	115
4.42	Exactitude avec une combinaison image / pré-traitement.	116
4.43	Exemple d'entraînements avec combinaisons images / pré-traitements.	116
4.44	Image d'entraînement synthétique utilisée pour l'entraînement.	117
4.45	Résultats de l'apprentissage des éléments structurants <i>U</i>	118
4.46	Les blocs bleus représentent l'opération de calcul classique de la couche.	120
4.47	Résultats de l'expérience utilisant U-Net et couches morphologiques.	120
4.48	Une image et ses représentations hiérarchiques [Carlinet and Géraud, 2015].	122
4.49	Opérateurs connexes et représentation hiérarchique.	123
4.50	Résultat de l'expérience de pré-traitement avec l'ouverture d'aire.	124
4.51	Pré-traitements de l'image <i>I3</i> , ouverture d'aire sur l'arbre des formes.	125
4.52	Pré-traitements d'un patch, ouverture d'aire sur l'arbre des formes.	126
4.53	Expérience de pré-traitement par opérateurs connexes, mitochondries.	126
4.54	Expérience de pré-traitement par opérateurs connexes, réticulum.	127
4.55	Exemple d'utilisation de GUIlibTIM pour analyser une image.	128
4.56	Exemples des pré-traitements appliqués sur une coupe de l'image <i>I3</i>	130
4.57	Résultats de l'expérience avec extraction d'attributs, mitochondries.	131
4.58	Résultats de l'expérience avec extraction d'attributs, réticulum.	131
A.1	Exemple d'importation de modules NeNIST.	140
A.2	Exemple d'utilisation de l'application Cutter.	141

A.3	Page de présentation de la bibliothèque TISM.	142
A.4	Exemple d’affichage de la bibliothèque LII.	143
A.5	Exemples d’interactions simulées par ISISIM.	144
A.6	Exemples d’annotations faibles simulées par SegmentationToScribble.	144
A.7	Exemple d’analyse de l’image $I3$ avec l’interface graphique GUILibTIM.	145
A.8	Exemple d’analyse de l’image $LW4$ avec l’interface graphique GUILibTIM.	146
A.9	Exemple de visualisation d’une prédiction générée par VREMS.	147
A.10	Exemples de visualisation générée par VREMS.	148
A.11	Exemples d’images aléatoires générées avec NumPyRandomShapes3D.	149
A.12	Exemple de comparaison faite avec SeSeCoCo.	150
A.13	Exemple de calcul de carte de distance avec DistanceMap.	150
A.14	Image $I3$, coupe 001.	155
A.15	Image $I3$ avec segmentation de référence, coupe 180.	155
A.16	Image $LW4-1$, coupe 001.	156
A.17	Image $LW4-1$ avec segmentation de référence, coupe 001.	156
A.18	Image $LW4-2$, coupe 116.	157
A.19	Image $LW4-2$ avec segmentation de référence, coupe 013.	157
A.20	Image $LW4-3$, coupe 074.	158
A.21	Image $LW4-4$, coupe 001.	158
A.22	Image $LW2-1$, coupe 059.	159
A.23	Image $LW2-4$, coupe 521.	159
A.24	Image $LW2-5$, coupe 096.	160
A.25	Représentation schématique de l’exploration scientifique de la thèse.	161

Liste des tableaux

3.1	Résolutions typiques des méthodes d'imagerie [Briggman and Bock, 2012].	39
3.2	Résolutions et tailles des images IGBMC.	44
3.3	Annotations disponibles pour les images IGBMC en nombre de tranches. .	46
3.4	Résolutions et tailles des images GoetzLab.	49
3.5	Informations pour les données accessibles publiquement.	57
4.1	Résumé des noms donnés aux jeux de données utilisés.	72
4.2	Résumé des caractéristiques des expériences de la section 4.2.	74
4.3	Nombre de paramètres et temps d'entraînement minimaux par architecture.	84
4.4	Exemples d'attributs pour l'élagage des structures hiérarchiques.	123
4.5	Liste des extracteurs d'attributs utilisés pour l'expérience.	129
4.6	Liste des opérateurs connexes utilisés pour l'expérience.	129
A.1	Liste des productions logicielles open source principales.	151

Résumé

Les travaux de thèse s’inscrivent dans un projet de collaboration entre le laboratoire ICube (équipe IMAGEs) et l’IGBMC (équipe Schultz). La thèse est financée par un contrat doctoral IDEX interdisciplinaire de l’université de Strasbourg.

Le cadre général de la thèse est l’analyse d’images de microscopie électronique cellulaire, et plus précisément la segmentation sémantique de ces images.

L’imagerie cellulaire par microscopie électronique est un sous-domaine de l’imagerie biomédicale. L’objectif de la discipline est d’imager des cellules avec la meilleure résolution possible, afin de mettre en évidence l’ultrastructure cellulaire. En particulier, les microscopes FIB-SEM permettent d’obtenir des images volumiques quasi isotropes. Les images 3D résultantes sont de résolution nanométrique et de très grande taille. La morphologie des cellules imagées en microscopie électronique fournit une grande quantité d’informations sur la taille, la forme et la distribution des organelles. Un atlas des paramètres de morphologie cellulaire normaux et pathologiques pourrait, à l’avenir, améliorer le diagnostic ou guider les chercheurs afin de mieux comprendre les pathologies. Jusqu’à récemment, ces informations étaient essentiellement descriptives, mais le développement de modes d’imagerie, tels que l’imagerie FIB-SEM, ouvre la perspective d’extraire des informations 3D quantitatives.

Une solution pour analyser ces images consiste à identifier les composantes cellulaires et à leur donner une étiquette qui correspondrait à un type de composant cellulaire ; cette tâche s’appelle la segmentation. Un expert humain peut identifier et annoter les caractéristiques cellulaires de manière manuelle, mais cette tâche est fastidieuse et nécessite plusieurs jours pour chaque cellule. De plus, dans un contexte de production massive de données (plusieurs dizaines de cellules et plusieurs dizaines de patients), le traitement manuel de telles quantités d’images est extrêmement long et coûteux. Cependant, une grande quantité d’images segmentées est nécessaire afin d’extraire des caractéristiques statistiquement significatives dans le cas d’un atlas morphologique.

Les méthodes de segmentation à base de réseaux de neurones convolutifs ont prouvé leur intérêt dans le cadre de l’imagerie biomédicale. Dans le contexte de la microscopie électronique, des travaux récents ont proposé des approches de segmentation utilisant ces techniques. En un peu plus de 10 ans, les techniques basées sur l’apprentissage profond

sont devenues la référence pour la segmentation des images. L'apprentissage profond et plus particulièrement les réseaux de neurones convolutifs sont particulièrement adaptés aux tâches de classification ou encore de segmentation d'images.

Du point de vue des biologistes, le problème est d'analyser les images cellulaires pour mieux comprendre le fonctionnement de la cellule, et pour comparer les phénotypes cellulaires. On parle de l'analyse morphologique des cellules, et d'analyses comparatives.

Pour permettre ces comparaisons, il est nécessaire d'obtenir une segmentation de qualité suffisante pour permettre une analyse morphologique des composantes cellulaires. Différents éléments rendent difficile la segmentation d'objets biologiques en microscopie électronique :

- Tout d'abord, les jeux de données annotées sont peu nombreux et la quantité de données est loin d'égaliser d'autres domaines comme la classification et la segmentation d'images de la vie courante. De plus, la création de telles données est plus complexe. Les microscopes électroniques sont bien moins courants que les appareils photo et seuls des experts qualifiés peuvent annoter ces images.
- Ensuite, chaque image contient de nombreux objets complexes et les images sont de nature bruitée (ratio signal/bruit et contraste faibles).
- La variation entre les images rend difficile la création de méthodes robustes. Les techniques de fixation, de coloration et la résolution varient fortement entre différentes images. Il en résulte des images phénoménologiquement différentes, avec des formes, textures et contrastes variés, où certains objets changent radicalement d'aspect.
- Enfin, une dérive lors de l'acquisition de l'image peut entraîner un décalage sur l'axe Z et des défauts d'isotropie. Le contraste peut ainsi varier d'une coupe à l'autre.

Les méthodes actuelles manquent généralement de robustesse, utilisent des jeux de données très importants ou se focalisent uniquement sur certains organites. La problématique est donc la suivante : produire une segmentation sémantique multi-classes sur des images de microscopie électronique volumique, robuste au bruit et aux variations d'acquisition, et son exploitation à des fins biomédicales.

L'objectif de la thèse est de développer des méthodes de segmentation robustes dédiées aux images de microscopie électronique d'objets biologiques dans le but de décrire quantitativement l'organisation tridimensionnelle des cellules et de leurs composants cellulaires.

Une première partie de la thèse consiste à développer et améliorer des méthodes de segmentation à base de réseaux de neurones convolutifs adaptées au contexte de l'imagerie FIB-SEM. L'objectif est de segmenter plusieurs types d'objets, mitochondrie, réticulum endoplasmique et, dans une moindre mesure, membrane cellulaire, noyau, endosome et appareil de Golgi.

Dans un second temps, deux axes principaux de recherche ont été explorés. L'amélioration de la qualité des segmentations en combinant apprentissage profond et morphologie mathématique et la réduction du temps d'annotation nécessaire, à l'aide de méthode interactive ou d'annotations faibles.

L'objectif du premier axe est d'évaluer l'utilisation des opérateurs connexes dans un cadre de simplification d'image afin d'améliorer, de faciliter ou d'accélérer l'apprentissage des réseaux convolutifs.

L'objectif du second axe de recherche est de réduire la durée d'annotations nécessaire à l'apprentissage d'un modèle. Pour ce faire, nous évaluons les capacités de modèles en réduisant la quantité ou la qualité d'annotation. Nous évaluons dans ce but les méthodes de segmentation interactive ainsi que les méthodes basées sur les annotations faibles ou clairsemées.

Les travaux de thèse ont permis de proposer de nouvelles méthodes de segmentation sémantique efficaces et adaptées aux problématiques des images de microscopie électronique cellulaire. Les travaux ont également permis d'évaluer les différentes méthodes issues de l'état de l'art afin d'en faire une synthèse et de mettre en perspectives les différents travaux récents.

Contributions

Les travaux effectués au cours de la thèse ont fait l'objet de communications en conférences internationales avec actes ou de communications lors de différents évènements scientifiques.

Conférences internationales avec actes

- [Meyer et al., 2021] C. Meyer, V. Mallouh, D. Spehner, E. Baudrier, P. Schultz, B. Naegel : "Automatic Multi Class Organelle Segmentation for Cellular FIB-SEM Images", ISBI 2021
- [Meyer et al., 2023] C. Meyer, E. Baudrier, P. Schultz, B. Naegel : "Combining max-tree and CNN for segmentation of cellular FIB-SEM images", RRPR, DGMM 2022

Communications orales

- 01/2020 Présentation des travaux de thèse aux étudiants du cursus master ingénierie (CMI)
- 09/2020 Présentation d'un poster pour la journée poster de l'école doctorale 269
- 11/2020 Présentation de la thèse pour les journées du Programme Doctoral International
- 04/2021 Présentation des travaux de thèse "Automatic multi class organelle segmentation for cellular FIB-SEM images" pour le Data Science and Artificial Intelligence Workshop
- 03/2022 "Flash Presentation" pour la Spring Session du Programme Doctoral International
- 05/2022 Présentation de l'équipe et de la thèse pour la journée des nouveaux entrants de l'ICube
- 10/2022 Présentation des travaux de thèse aux étudiants ingénieurs des mines de 3e année
- 11/2022 Présentation des travaux publiés à DGMM RRPR 2022 pour le séminaire de l'équipe IMAGeS
- 11/2022 Présentation des travaux publiés à DGMM RRPR 2022 pour le GT Géométrie Discrète et Morphologie Mathématique 2022

Introduction

1.1	Contexte	2
1.1.1	Imagerie biomédicale cellulaire et microscopie électronique FIB-SEM	2
1.1.2	Analyse d'images de microscopie électronique cellulaire	6
1.1.3	Apprentissage profond et méthodes supervisées	8
1.2	Problématique	11
1.3	Objectifs et contributions	12
1.4	Organisation du manuscrit	13

Cette thèse se place à l'intersection de deux domaines, d'un côté l'informatique, plus précisément l'analyse d'image par apprentissage profond, de l'autre, l'imagerie biomédicale, en particulier, les images de microscopie électronique FIB-SEM cellulaire.

Trois acteurs institutionnels ont pris part au projet.

- L'équipe IMAGEs, du laboratoire ICube pour le traitement informatique des images.
- L'équipe Schultz, du laboratoire IGBMC pour l'imagerie de microscopie électronique FIB-SEM cellulaire et les annotations des images.
- L'équipe Goetz, du CRBS pour l'imagerie de microscopie électronique FIB-SEM cellulaire in-vivo ainsi que la recherche fondamentale et appliquée en cancérologie.

Dans ce chapitre d'introduction, nous verrons tout d'abord le contexte dans lequel nous travaillons, les images que nous traitons ainsi que les méthodes utilisées pour les analyser. Nous présenterons ensuite la problématique, en prenant en compte le point de vue des biologistes et les verrous scientifiques. Enfin, nous définirons clairement les objectifs et le sujet d'étude.

1.1 Contexte

L'imagerie cellulaire par microscopie électronique est un sous-domaine de l'imagerie biomédicale. L'objectif de la discipline est d'imager des cellules avec la meilleure résolution possible, afin de mettre en évidence les composantes des cellules. En particulier, les microscopes FIB-SEM permettent d'obtenir des images volumiques quasi isotropes. La figure 1.1 représente les résolutions approximatives praticables entre différentes méthodes par rapport à la taille des objets d'intérêt.

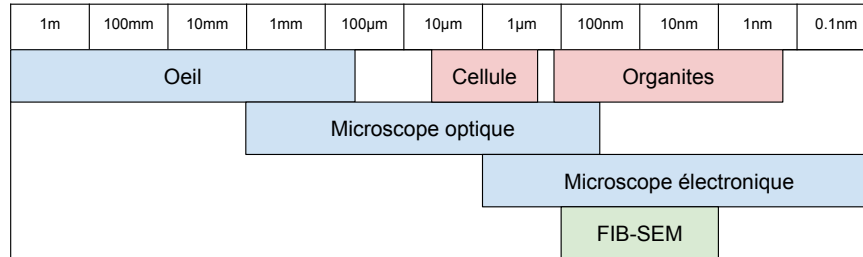


FIGURE 1.1 – Comparaison schématique simplifiée des résolutions.

Les images 3D résultantes sont de résolution nanométrique et de très grande taille. L'observation visuelle de ces images 3D est informative, mais limitée et il est souhaitable d'identifier les composants cellulaires automatiquement afin de permettre des comparaisons quantitatives entre différentes cellules, mais également pour visualiser en trois dimensions les cellules de manière à mieux appréhender leur fonctionnement. Nous détaillons les particularités de l'imagerie de microscopie électronique FIB-SEM cellulaire dans la section 1.1.1.

Une solution pour analyser ces images consiste à identifier les composantes cellulaires et à leur donner une étiquette qui correspondrait à un type de composant cellulaire, cette tâche s'appelle la segmentation. En un peu plus de 10 ans, les techniques basées sur l'apprentissage profond sont devenues la référence pour la segmentation des images. L'apprentissage profond et plus particulièrement les réseaux de neurones convolutifs sont particulièrement adaptés aux tâches de classification ou encore de segmentation d'images. Nous présentons les différents types de segmentation et méthodes d'apprentissage pour l'analyse d'images dans les sections 1.1.2 et 1.1.3.

1.1.1 Imagerie biomédicale cellulaire et microscopie électronique FIB-SEM

L'imagerie biomédicale concerne toutes données issues d'imageurs biologiques ou médicaux. Historiquement, la discipline est née de la volonté de mieux comprendre le vivant avec les premières tentatives de visualisation de l'intérieur des cellules à l'aide de la microscopie optique (microscope photonique) au milieu du XVIIe siècle. L'imagerie médicale a pour but de restituer des images du corps humain et remonte à la fin du XIXe siècle avec les travaux sur les rayons X. Les premières images biomédicales concernaient tout d'abord

l'extérieur du corps ou l'intérieur de cadavres, mais l'utilisation de nouvelles techniques a permis de voir l'intérieur d'un corps sans pour autant devoir l'ouvrir, notamment grâce à l'utilisation de rayonnement électromagnétique avec des longueurs d'onde non visible comme les rayons X. L'évolution des techniques a également permis d'observer des objets de plus en plus petits. Aujourd'hui, l'imagerie biomédicale concerne par exemple l'imagerie par résonance magnétique, l'imagerie de médecine nucléaire, l'imagerie par ultrasons, les scanners à rayons X, ou encore la microscopie électronique.

L'imagerie cellulaire est un sous-domaine de l'imagerie biomédicale, qui se concentre sur les images comprenant une ou plusieurs cellules. L'objectif de ces images est de visualiser une ou plusieurs cellules et les différentes parties composant les cellules.

En particulier, les cellules sont composées de structures spécialisées appelées organelles (aussi nommées organites). Comme le nom le suggère, les organelles microscopiques sont les « organes » de la cellule. Il existe beaucoup de types d'organelles différentes, chacun remplissant une ou plusieurs tâches particulières. Dans notre étude, nous ne nous intéresserons qu'à certaines organelles. Voici une liste de différentes organelles étudiées avec une explication succincte de leur fonction au sein de la cellule.

- La **mitochondrie** convertit l'énergie des molécules organiques issues de la digestion en énergie directement utilisable par la cellule, l'adénosine triphosphate (ATP). C'est la « centrale énergétique » de la cellule.
- Le **réticulum endoplasmique** (RE) synthétise les protéines, produit des macromolécules et transfère des substances vers l'appareil de Golgi.
- Le **noyau** stocke le génome ainsi que la machinerie nécessaire à la réplication des chromosomes et à l'expression de l'information contenue dans les gènes.
- La **membrane cellulaire** (également appelée membrane plasmique) délimite l'intérieur et l'extérieur de la cellule.
- L'**appareil de Golgi** permet le transfert et le tri, il synthétise la plupart des sécrétions cellulaires. C'est un intermédiaire entre le réticulum endoplasmique et la membrane cellulaire.
- L'**endosome** trie la matière avant qu'elle n'atteigne les lysosomes (organelles de digestion intracellulaire) pour y être dégradée.

Un microscope électronique est un microscope qui utilise un faisceau d'électrons accélérés comme source d'illumination. Les microscopes électroniques permettent d'obtenir des images avec une résolution plus élevée que les microscopes optiques (aussi appelés microscopes photoniques) et peuvent ainsi révéler la structure d'objets plus petits comme les organelles d'une cellule. Il existe différents types de microscopes électroniques, les plus communs sont les microscopes électroniques en transmission (MET, ou TEM pour Transmission Electron Microscopy) et les microscopes électroniques à balayage (MEB, ou SEM pour Scanning Electron Microscopy). La microscopie électronique a permis une grande augmentation des résolutions d'imagerie, atteignant aujourd'hui l'ordre de 0,5 Å (0,05 nm) en MET.

La **microscopie électronique à balayage** consiste en un faisceau d'électrons balayant la surface d'un échantillon qui réémet des particules. Un système de détection analyse ces particules afin de reconstruire une image de la surface.

La microscopie par transmission est une technique de microscopie où un faisceau d'électrons traverse un échantillon. Le résultat des interactions entre les électrons et l'échantillon permet de créer une image. La MET permet d'obtenir de meilleures résolutions dans le plan focal que la MEB, mais l'objet imagé n'est alors pas la surface d'un échantillon, c'est la projection d'une coupe. La figure 1.2 représente des images obtenues avec les deux techniques.

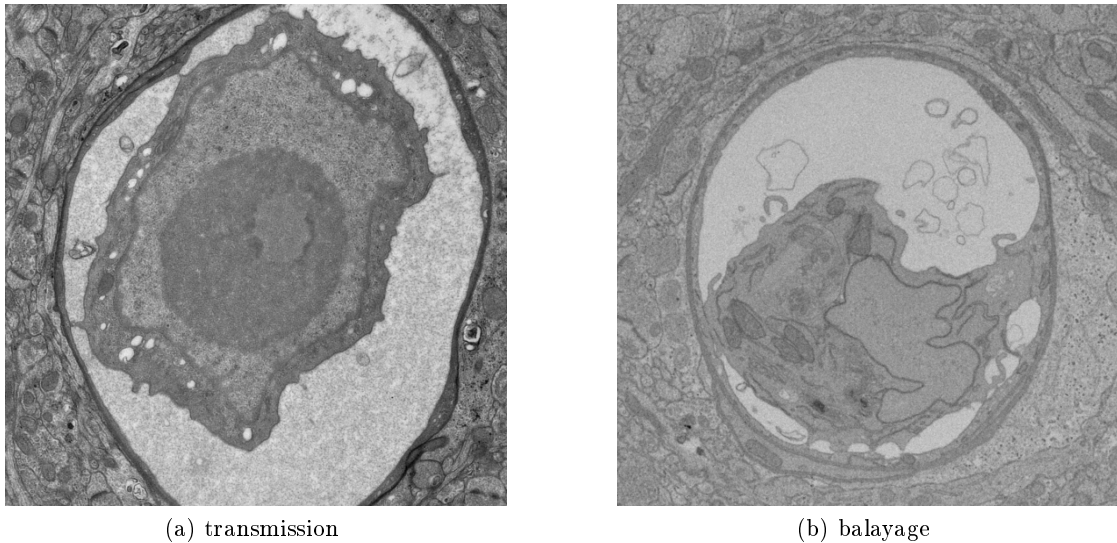


FIGURE 1.2 – Images de microscopie électronique : par transmission (serial TEM) 1.2a et par balayage (FIB-SEM) 1.2b.

La technologie de microscopie FIB-SEM [Heymann et al., 2006, Knott et al., 2008] permet d'acquérir des images de cellules entières avec une haute définition spatiale et fournit des images de très grande taille (de l'ordre de 2000 pixels au cube, soit plusieurs milliards de pixels), qui contiennent de nombreux objets biologiques complexes. Le principe de fonctionnement du FIB-SEM est le suivant : un faisceau ionique focalisé (FIB) découpe une couche mince à la surface de l'échantillon, tandis qu'un microscope électronique à balayage image la face du bloc nouvellement créée. La figure 1.3 représente le principe de fonctionnement du FIB-SEM. L'itération de ces deux processus offre la possibilité d'imager des zones de plusieurs dizaines de μm avec une résolution spatiale quasi isotrope pouvant aller jusqu'à 3 nm. Dans la pratique, les images sont rarement isotropes et les résolutions varient entre 3 et 30 nm. L'isotropie est un point fort de la technique FIB-SEM.

La microscopie FIB-SEM n'est pas réservée aux images biomédicales et cellulaires, mais dans ce domaine, la haute définition permet de visualiser les composantes cellu-

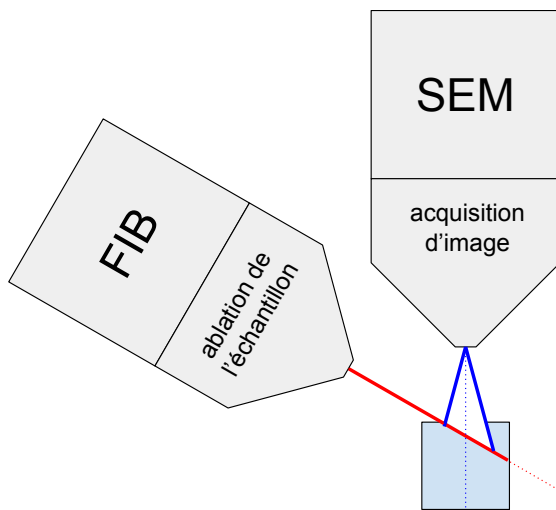


FIGURE 1.3 – Principe de fonctionnement d'un microscope FIB-SEM.

laire en 3 dimensions. La figure 1.4 représente 3 exemples d'images FIB-SEM cellulaire auxquels nous nous sommes intéressés. Visuellement, ces 3 images sont très différentes, ce qui s'explique par des variations dans le processus de préparation de l'échantillon ou d'acquisition de l'image. Ces différences seront détaillées pour chacune des images que nous utiliserons dans le chapitre 3.

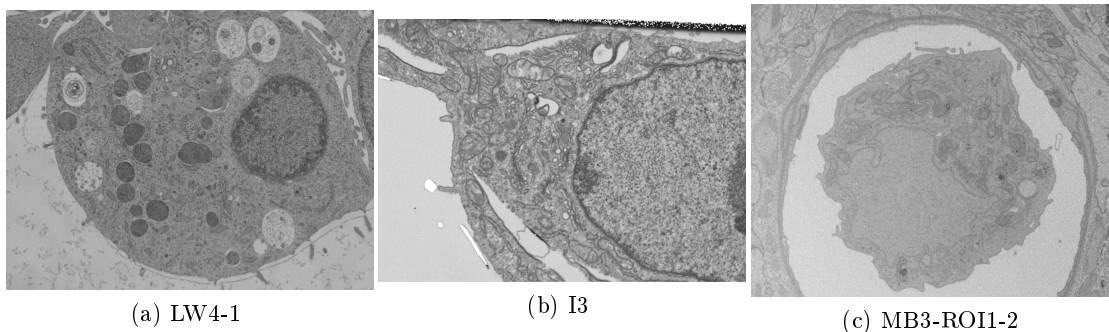


FIGURE 1.4 – Exemples d'images FIB-SEM cellulaires. Les noms de codes utilisés dans la thèse sont constants.

La morphologie des cellules imagées en microscopie électronique fournit une grande quantité d'informations sur la taille, la forme et la distribution des organelles. Un atlas des paramètres de morphologie cellulaire normaux et pathologiques pourrait, à l'avenir, améliorer le diagnostic ou guider les chercheurs afin de mieux comprendre les pathologies [Santucci and Franchi, 2008, Franchi and Santucci, 2013]. Jusqu'à récemment, ces informations étaient essentiellement descriptives, mais le développement de modes d'ima-

gerie, tels que l'imagerie FIB-SEM, ouvre la perspective d'extraire des informations 3D quantitatives.

1.1.2 Analyse d'images de microscopie électronique cellulaire

L'**analyse d'images** correspond à la reconnaissance des éléments et des informations contenus dans une image. Elle peut être manuelle ou automatique pour les images numériques. Une image est numérique quand elle est acquise puis stockée sous forme binaire ; une image numérique correspond alors à un tableau de pixels. L'analyse d'images numériques est un domaine vaste qui comporte beaucoup de tâches différentes, tant par leurs natures que leurs applications. Ces applications vont de la lecture de codes-barres, la classification d'images de la vie courante ou de photos aériennes jusqu'à la reconnaissance automatisée d'objets dans des images biomédicales, en passant par l'utilisation industrielle pour le contrôle qualité.

L'informatique a joué un rôle clé dans l'évolution de l'analyse d'images numériques, en permettant de stocker, de traiter et de visualiser de grandes quantités de données de manière rapide et efficace. En particulier, l'utilisation de l'apprentissage automatique, ou machine learning, a permis de développer des outils capables de détecter et de caractériser automatiquement les éléments présents dans les images, ce qui a considérablement simplifié et accéléré l'analyse des données.

Pour pouvoir effectuer une **analyse quantitative** des caractéristiques morphologiques (taille, distribution, formes) des différentes composantes cellulaires, une première étape de segmentation est nécessaire. Cette étape de segmentation correspond à l'attribution d'une classe à chaque voxel, ce qu'on appelle une segmentation sémantique. Le problème de **segmentation sémantique** se différencie de la détection d'objet ou de la segmentation d'instance pour laquelle deux objets différents de même classe sont annotés différemment. Un exemple de ces différents types de segmentation est représentée dans la figure 1.5.

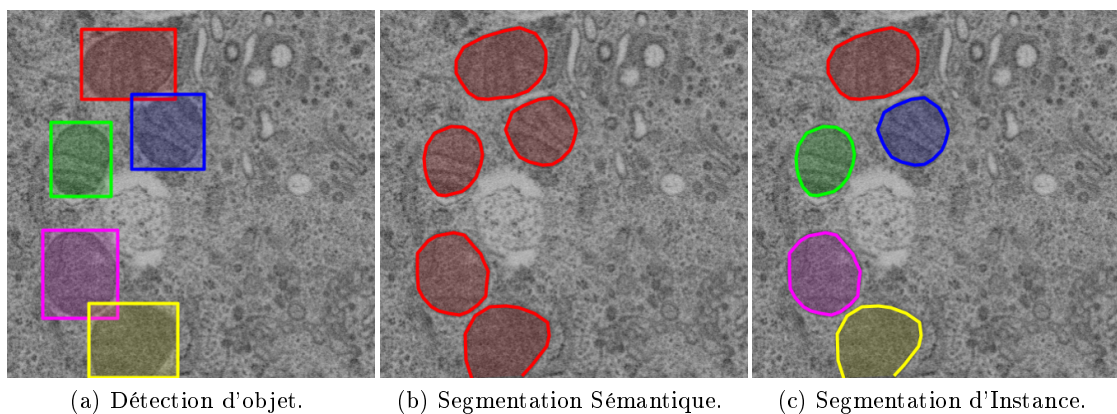


FIGURE 1.5 – Différents types de segmentations.

Une segmentation peut être binaire ou multi-classes. Dans le premier cas, on segmente les objets d'intérêt par rapport au fond. Dans le second, une classe différente est affectée à chaque voxel. Une segmentation multi-classes peut être encodée de différentes manières ; on peut affecter une valeur différente à chaque classe (par exemple mitochondries = 1, noyau = 2) ou utiliser un vecteur contenant autant de valeurs que de classes pour chaque voxel. Ce vecteur peut être « one-hot » (une seule valeur à 1) ou permettre à un voxel d'être dans plusieurs classes simultanément.

Un expert humain peut identifier et annoter les caractéristiques cellulaires de manière manuelle, mais cette tâche est fastidieuse et nécessite plusieurs jours pour chaque cellule. De plus, dans un contexte de production massive de données (plusieurs dizaines de cellules et plusieurs dizaines de patients), le traitement manuel de telles quantités d'images est extrêmement long et coûteux. Cependant, une grande quantité d'images segmentées est nécessaire afin d'extraire des caractéristiques statistiquement significatives dans le cas d'un atlas morphologique. Les méthodes de segmentation à base de réseaux de neurones convolutifs ont prouvé leur intérêt dans le cadre de l'imagerie biomédicale. Dans le contexte de la microscopie électronique, des travaux récents ont proposé des approches de segmentation utilisant ces techniques [Ronneberger et al., 2015, Çiçek et al., 2016, Zeng et al., 2017, Haberl et al., 2018, Xiao et al., 2018, Moen et al., 2019, Liu et al., 2020, Heinrich et al., 2021]. La figure 1.6 représente un exemple d'images FIB-SEM cellulaire segmentée manuellement.

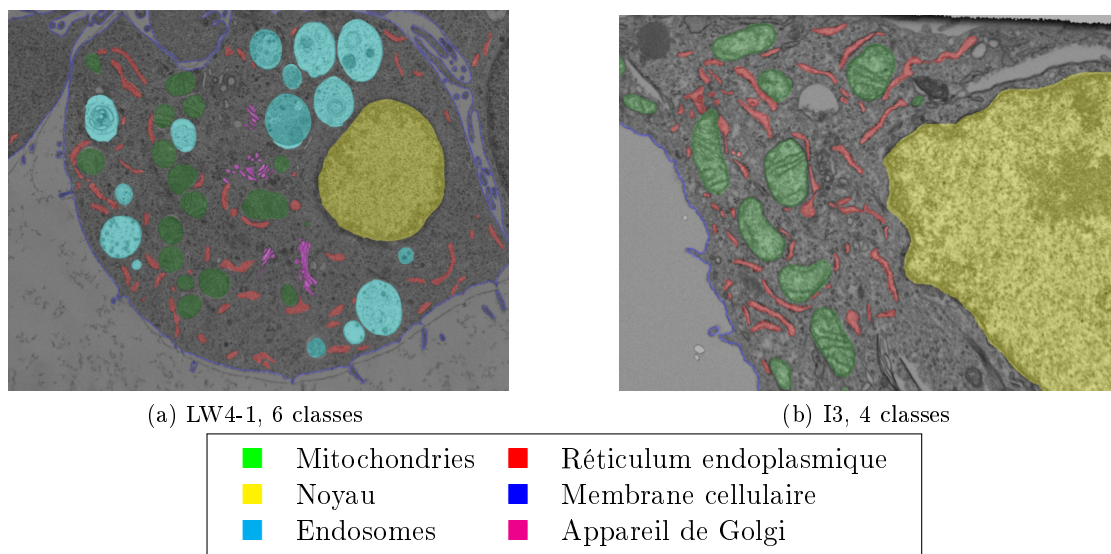


FIGURE 1.6 – Exemple d'images FIB-SEM cellulaires avec segmentation sémantique de différentes organelles. Mitochondries (vert), réticulum endoplasmique (rouge), membrane cellulaire (bleu), noyau (jaune), appareil de Golgi (magenta) et endosomes (cyan).

1.1.3 Apprentissage profond et méthodes supervisées

L'apprentissage automatique (machine learning) est un sous-domaine de l'intelligence artificielle qui s'intéresse à l'élaboration de méthodes permettant de produire des modèles prédictifs à partir de données. Les algorithmes d'apprentissage peuvent être supervisés ou non supervisés.

L'apprentissage supervisé consiste à entraîner un modèle prédictif en utilisant une base de données d'exemples appelée base d'apprentissage. On distingue deux types de tâches supervisées, la régression et la classification. Dans le cas de la régression, la sortie est une valeur dans un intervalle réel. En classification, la sortie est une valeur dans un ensemble fini, ce qui revient à attribuer une étiquette à chaque entrée.

L'apprentissage profond (deep learning) est un ensemble de méthodes d'apprentissage automatique utilisant des architectures articulées autour de couches successives de traitement non linéaires. Ces couches de traitement permettent l'extraction et la transformation de caractéristiques issues des données avec un haut niveau d'abstraction. Chaque couche prend en entrée la sortie de la précédente. Dans la suite de la thèse, nous nous concentrerons sur les méthodes d'**apprentissage profond supervisé**.

La figure 1.7 représente l'imbrication schématique des domaines autour de l'apprentissage profond.

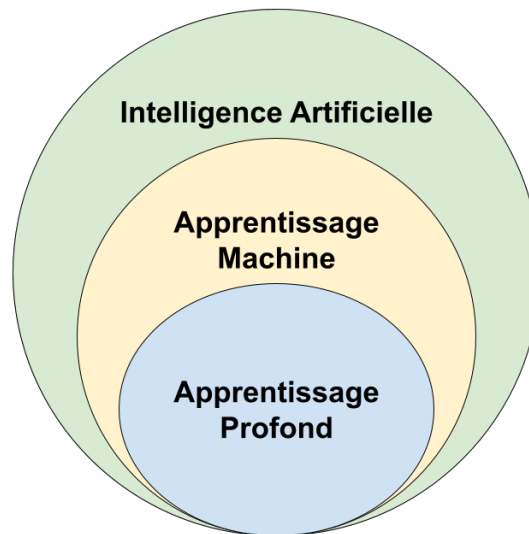


FIGURE 1.7 – Imbrication du domaine de l'apprentissage profond.

Afin d'éviter toute confusion, nous souhaitons définir les termes suivants.

- Un **modèle (de réseau de neurones)** est un réseau de neurones. Il est composé d'un agencement de couches de traitement et des poids correspondants. Un modèle peut être dans un état entraîné ou non.

- Une **architecture** est un « plan de construction » de réseau de neurones, (par exemple l'architecture U-Net). L'architecture représente l'organisation des différentes couches de traitements entre elles.
- Une **brique de construction** est une composition de couches élémentaires, généralement répétée dans une architecture. Le terme **backbone** fait généralement référence à un modèle de classification (par exemple ResNet50) auquel on retire la dernière couche afin d'utiliser le réseau dans la partie contractante, aussi appelée « encodeur », mais il peut aussi faire référence à un ensemble de briques de constructions.

Historiquement, l'apprentissage profond a porté plusieurs noms et a connu plusieurs phases de développement successives, entrecoupées de périodes de creux, parfois appelées « hiver ». La cybernétique dans les années 1940 - 1960, le connexionnisme dans les années 1980 - 1990 et depuis les années 2000, l'apprentissage profond [Goodfellow et al., 2016]. La figure 1.8 représente la fréquence d'utilisation de ces différents noms en fonction des années.

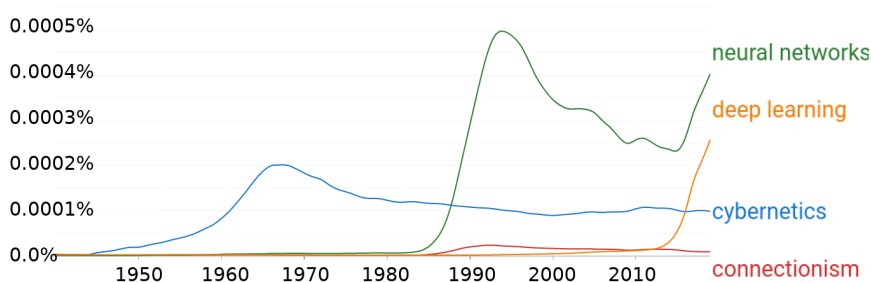


FIGURE 1.8 – Fréquence d'utilisation des différents noms. Source : Google Books Ngram Viewer.

Sans détailler l'historique complet du domaine, on peut noter plusieurs étapes importantes. Tout d'abord, les travaux sur le neurone formel [McCulloch and Pitts, 1943], une version simplifiée du neurone biologique, et le perceptron [Rosenblatt, 1958] qui propose le premier système artificiel capable d'apprendre à partir d'une base d'apprentissage. Ensuite, le perceptron est décliné dans une version multicouche et entraîné grâce à l'algorithme de rétropropagation du gradient [Rumelhart et al., 1986]. Quelques années plus tard, le réseau LeNet, propose l'utilisation d'un des premiers réseaux de neurones convolutifs pour la reconnaissance de codes postaux manuscrits [LeCun et al., 1989].

En 2009, la base de données ImageNet [Deng et al., 2009] est présentée pour la première fois et en 2010, la compétition « ImageNet Large Scale Visual Recognition Challenge » (ILSVRC) compétition annuelle de traitement d'images, est organisée pour la première fois. En 2012, le réseau de neurones convolutifs AlexNet [Krizhevsky et al., 2012a] bat largement tous les records de classification de la compétition. La figure 1.9 montre l'erreur de classification des premières années du challenge ILSVRC.

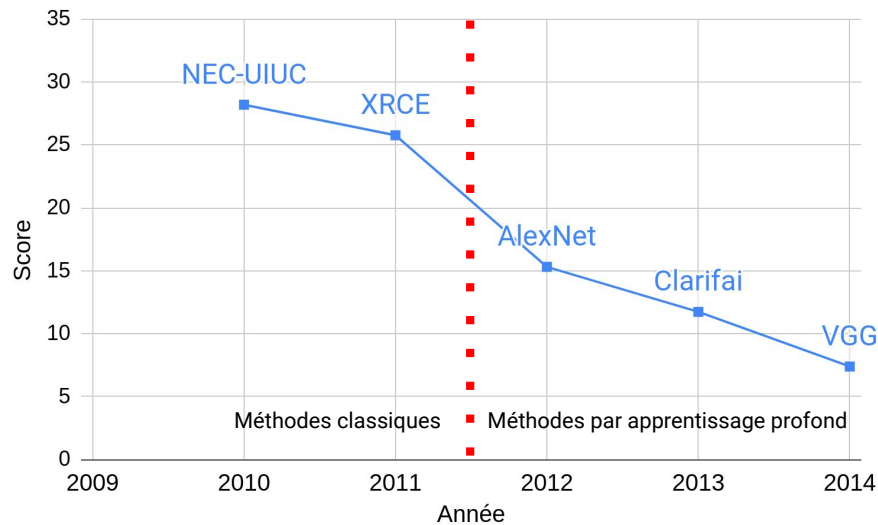


FIGURE 1.9 – Comparaison des résultats des 5 premières années de la compétition ILS-VRC. Les pointillés représentent le passage des méthodes classiques aux méthodes par apprentissage profond.

C'est durant cette période que les réseaux de neurones gagnent en popularité, les frameworks se multiplient (Torch en 2002, Theano en 2007, Caffe en 2013, Keras et TensorFlow en 2015 puis PyTorch en 2016) et les utilisations se diversifient. Aujourd'hui, l'apprentissage profond est l'un des domaines de recherche en informatique les plus actifs.

L'analyse d'image par apprentissage profond est l'un des sous-domaines les plus importants dans le secteur de l'analyse d'images. Un des outils importants est la couche de convolution, qui applique une opération de convolution sur une entrée donnée. Cette opération, particulièrement adaptée aux images, quelles que soient leurs dimensions, permet de traiter des images beaucoup plus efficacement qu'en utilisant des couches classiques dites denses. Les couches de convolution ne sont pas densément interconnectées, ce qui signifie que la sortie d'un neurone ne dépend pas de l'entièreté de la couche précédente, et n'a pas un impact sur tous les neurones de la couche suivante. Cela donne aux couches de convolution plus de flexibilité, et permet de réduire le nombre de paramètres.

Si les premiers travaux d'analyse d'images se concentraient principalement sur de la classification [LeCun et al., 1989, Krizhevsky et al., 2012a], en 2015, les réseaux « entièrement convolutifs » (FCN pour Fully Convolutional Networks) [Long et al., 2015] sont proposés afin de permettre la segmentation sémantique, entraînée pixel à pixel. Ces modèles permettent de traiter des images de taille arbitraire avec des tailles de sortie correspondant aux entrées. Ces travaux adaptent les réseaux de classification classique, AlexNet, VGG ou encore LeNet en réseau FCN pour la segmentation.

Les techniques d'apprentissage profond basées sur les **réseaux de neurones convolutifs** (CNN pour Convolutional Neural Network) sont particulièrement performantes.

Nous considérerons les FCN comme une sous-classe des CNN, qui utilisent des couches de convolutions.

Dans le contexte de l'analyse d'images de microscopie électronique cellulaire, l'apprentissage profond a permis de développer des outils capables de détecter et de caractériser les différents éléments constitutifs de l'intérieur des cellules [Ronneberger et al., 2015, Xiao et al., 2018, Liu et al., 2020, Heinrich et al., 2021]. Cependant, les méthodes basées sur l'apprentissage profond ont souvent des difficultés à traiter des images complexes et à gérer les variations de l'apparence des éléments d'intérêt. Nous détaillerons plus précisément ces méthodes dans le chapitre 2.

1.2 Problématique

Du point de vue d'un biologiste, le problème est d'analyser les images cellulaires pour mieux comprendre le fonctionnement de la cellule, et pour comparer les phénotypes cellulaires. On parle de l'analyse morphologique des cellules, et d'analyses comparatives.

Pour permettre ces comparaisons, il est nécessaire d'obtenir une segmentation de qualité suffisante pour permettre une analyse morphologique des composantes cellulaires.

Différents éléments rendent difficile la segmentation d'objets biologiques en microscopie électronique.

Tout d'abord, les jeux de données annotées sont peu nombreux et la quantité de données est loin d'égaliser d'autres domaines comme la classification et la segmentation d'images de la vie courante.

De plus, la création de telles données est plus complexe. Les microscopes électroniques sont bien moins courants que les appareils photo et seuls des experts qualifiés peuvent annoter ces images.

Ensuite, chaque image contient de nombreux objets complexes et les images sont de nature bruitée (ratio signal/bruit et contraste faibles).

Mais encore, la variation entre les images rend difficile la création de méthodes robustes. Les techniques de fixation, de coloration et la résolution varient fortement entre différentes images. Il en résulte des images phénoménologiquement différentes, avec des formes, textures et contrastes variés, où certains objets changent radicalement d'aspect (figure 1.4).

Enfin, une dérive lors de l'acquisition de l'image peut entraîner un décalage sur l'axe Z et des défauts d'isotropie. Le contraste peut ainsi varier d'une coupe à l'autre.

Les méthodes actuelles manquent généralement de robustesse, utilisent des jeux de données très importants ou se focalisent uniquement sur certaines organelles.

La problématique est donc la suivante : produire une segmentation sémantique multi-classes sur des images de microscopie électronique volumique, robuste au bruit et aux variations d'acquisition, et son exploitation à des fins biomédicales.

Le public ciblé par les résultats de nos contributions est constitué principalement des ingénieurs de recherche ou techniciens travaillant dans des laboratoires de recherche biomédicale. Nos méthodes ont donc également comme prérequis l'utilisation de matériel courant et l'utilisation d'une quantité d'annotation limitée. Nous avons défini la notion de

« matériel courant » à partir des stations de travail « haut de gamme » disponibles sur le marché, la configuration de référence est la suivante : Processeur Intel Xeon W-2135, carte graphique NVIDIA GeForce RTX 2080 Ti (11G de VRAM), 64G de mémoire RAM et un stockage sur SSD.

1.3 Objectifs et contributions

L’objectif de la thèse est de développer des méthodes de segmentation robustes dédiées aux images de microscopie électronique d’objets biologiques dans le but de décrire quantitativement l’organisation tridimensionnelle des cellules et de leurs composants cellulaires.

Une première partie de cette thèse consiste à développer et améliorer des méthodes de segmentation à base de réseaux de neurones convolutifs adaptées au contexte de l’imagerie FIB-SEM. L’objectif est de segmenter plusieurs types d’objets, mitochondrie, réticulum endoplasmique et, dans une moindre mesure, membrane cellulaire, noyau, endosome et appareil de Golgi.

Dans un second temps, l’objectif est d’évaluer l’utilisation des opérateurs connexes dans un cadre de simplification d’image afin d’améliorer, de faciliter ou d’accélérer l’apprentissage des réseaux convolutifs.

Enfin, un objectif est de réduire la durée d’annotations nécessaires à l’apprentissage d’un modèle. Pour ce faire, nous évaluerons les capacités de modèles en réduisant la quantité ou la qualité d’annotation. Nous évaluerons dans ce but les méthodes de segmentation interactive ainsi que les méthodes basées sur les annotations faibles ou clairsemées.

Les principales contributions de la thèse s’articulent autour des techniques de segmentation utilisant l’apprentissage profond. Nous avons tout d’abord validé nos méthodes développées à partir de l’état de l’art, en expérimentant différents paramètres d’apprentissage comme la taille des patches, les modèles 2D et 3D, les fonctions de pertes, la modification des architectures, la taille des jeux d’entraînement ou l’utilisation de modèles pré-entraînés. Nous avons validé l’utilisation de ces méthodes dans le cas de la segmentation binaire et multi-classes sur nos images [Meyer et al., 2021]. Nous avons également contribué aux méthodes combinant apprentissage profond et morphologie mathématique. Ces travaux ont fait l’objet de communications en conférences internationales. Une première communication porte sur la segmentation automatique multi-classes d’organelles pour les images de microscopie FIB-SEM cellulaire et l’utilisation de filtres connexes [Meyer et al., 2021]. Une seconde communication concerne la combinaison de max-tree et CNN pour la segmentation des images FIB-SEM cellulaires [Meyer et al., 2023]. Enfin, nous avons étudié l’utilisation de méthodes interactives ou utilisant des annotations faibles afin d’améliorer la robustesse de nos modèles pour lesquels le transfert de domaine est difficile. Toutes ces contributions ont nécessité un développement logiciel qui a été rendu accessible et libre. Les données produites et utilisées durant la thèse ont également été rendues accessibles (plus de détails dans la section A.1).

1.4 Organisation du manuscrit

Le chapitre 2, présente l'état de l'art correspondant à notre problématique. Dans le chapitre 3, nous verrons l'ensemble des données que nous avons utilisées dans le cadre de nos expérimentations ainsi que les méthodes d'évaluation mises en œuvre pour évaluer nos modèles prédictifs. Nous nous intéressons ensuite dans le chapitre 4 aux méthodes mises en œuvre et aux résultats de nos expériences. Enfin, nous présentons une synthèse et concluons dans le chapitre 5.

État de l'art

2.1	Segmentation d'images par apprentissage profond	15
2.1.1	Biomédical et microscopie électronique cellulaire	18
2.1.2	Segmentation interactive d'images	25
2.2	Apprentissage profond et morphologie mathématique	30
2.2.1	Les réseaux de neurones morphologiques	30
2.2.2	Application de la morphologie mathématique en apprentissage profond	32

Dans ce chapitre, nous ferons un état des lieux des méthodes actuelles de segmentation par apprentissage profond. Nous commencerons par les méthodes d'applications générales puis nous détaillerons celles adaptées ou développées pour la microscopie électronique dont les caractéristiques sont différentes des bases d'images naturelles ou médicales. Nous nous concentrerons en particulier sur les méthodes liées à la microscopie électronique cellulaire FIB-SEM. Nous nous intéresserons également aux méthodes basées sur les interactions utilisateur qui permettent une réduction du temps nécessaire à la segmentation manuelle des images. Enfin, nous verrons les différents travaux mêlant apprentissage profond et morphologie mathématique afin d'étudier les possibilités de mélange des deux disciplines dans un objectif d'amélioration de la qualité des segmentations.

2.1 Segmentation d'images par apprentissage profond

Comme nous l'avons déjà précisé dans le chapitre 1, l'analyse d'image par apprentissage profond est l'un des sous-domaines les plus importants dans le secteur de l'analyse d'images. Ceci entraîne une amélioration des méthodes et un grand nombre de contributions scientifiques chaque année. L'état de l'art dans le domaine de la segmentation d'images cellulaires a évolué rapidement entre le début de la thèse en hiver 2019 et au-

jourd'hui, et continuera probablement dans les années à venir. Cet état de l'art prend uniquement en compte les contributions ayant eu lieu avant mars 2023.

En 2015, Long et al. proposent le modèle de réseaux entièrement convolutifs (FCN pour Fully Convolutional Networks) [Long et al., 2015]. C'est une étape importante qui permet d'utiliser les CNN pour la segmentation et non plus uniquement pour la classification. Ils proposent des versions modifiées des modèles AlexNet [Krizhevsky et al., 2012b], VGG16 [Simonyan and Zisserman, 2015] et GoogLeNet [Szegedy et al., 2015] afin de les utiliser pour la segmentation d'images (figure 2.1).

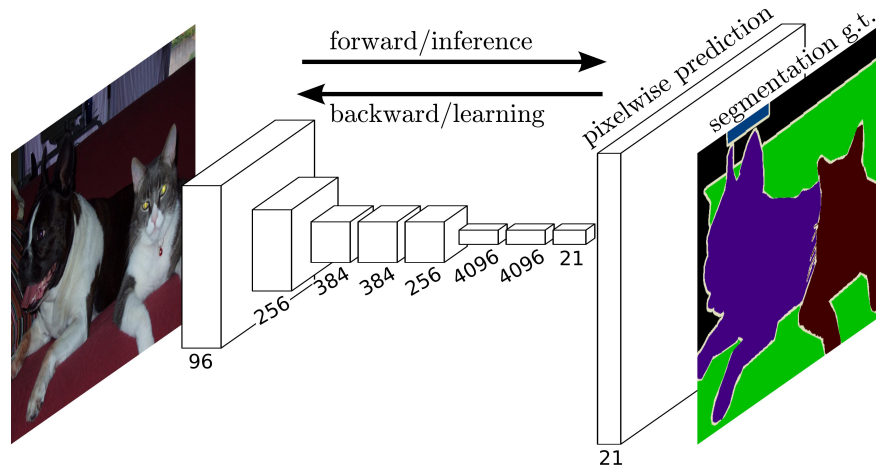


FIGURE 2.1 – Un modèle FCN [Long et al., 2015].

Depuis, de nombreuses architectures ont été proposées afin d'améliorer les performances. On peut regrouper les approches les plus répandues en plusieurs catégories.

Les **architectures encodeur / décodeur**, qui utilisent des réseaux en deux parties comme DeConvNet [Noh et al., 2015], SegNet [Badrinarayanan et al., 2017] ou U-Net [Ronneberger et al., 2015]. Ces architectures utilisent un encodeur, comme les modèles FCN, afin de transformer l'entrée du réseau en vecteur de caractéristiques (ou carte de caractéristiques, feature maps) concentrant les informations de l'image utiles à la segmentation et un décodeur pour générer une carte de probabilité, une prédiction pour chaque pixel. Le décodeur utilise des opérations de sur-échantillonnage ou de convolution transposée dans le but d'augmenter la dimension de la sortie.

Les **architectures à convolutions à trous**, qui utilisent les convolutions à trous [Holschneider et al., 1990, Shensa, 1992] (convolution dilatée, figure 2.2b) afin d'augmenter la taille du champ récepteur sans utiliser d'opération de sous-échantillonnage. Les modèles les plus notables sont ceux de la famille de l'architecture DeepLab [Chen et al., 2018a, Chen et al., 2018b],

Il existe d'autres types d'architectures, par exemple les architectures multi-échelles et pyramidales comme FPN (Feature Pyramid Networks) [Lin et al., 2017] ou PSPNet (Pyramid Scene Parsing Network) [Zhao et al., 2017] qui mélangent les cartes de caractéristiques à différents niveaux de résolution.

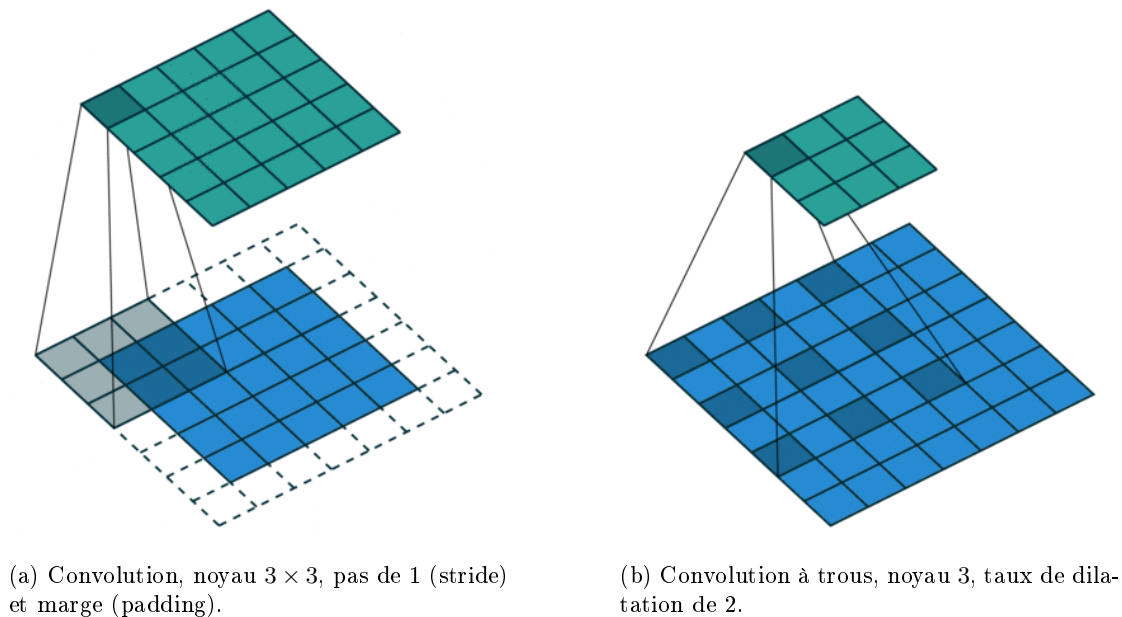


FIGURE 2.2 – Convolution et convolution à trous [Dumoulin and Visin, 2016].

Enfin, depuis le début des années 2020, les Vision Transformers (ViT) [Cordonnier et al., 2020, Dosovitskiy et al., 2023], basés sur les Transformers [Vaswani et al., 2017] ont surpassé les CNN pour les tâches de classification d’images. Pour les tâches de segmentation et de détection d’objet, les Transformers classiques rencontrent des difficultés. Une nouvelle classe dite « hiérarchique » a été introduite comme les Swin Transformers [Liu et al., 2021]. Il est cependant important de noter que cette approche a été critiquée, en particulier en démontrant qu’un réseau « classique » uniquement basé sur des modules de réseaux de neurones convolutifs peut tout à fait surpasser un ViTs pour des tâches de segmentation : « *However, the effectiveness of such hybrid approaches is still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions* » [Liu et al., 2022].

Les représentants les plus connus des deux premières catégories sont les architectures U-Net et DeepLab.

Il existe encore beaucoup d’autres catégories moins répandues, mais également de petites variations dans les architectures proposées. La figure 2.3 représente chronologiquement quelques architectures proposées entre 2014 et 2019.

Dans cette première section, nous avons vu un rapide historique et les architectures principales des méthodes de segmentation d’images par apprentissage profond. Il serait possible d’aller beaucoup plus dans le détail des différentes méthodes développées ces dernières années, mais concentrons-nous plutôt sur notre problématique. Plusieurs articles récents passent en revue de manière détaillée les technologies pour la segmentation basée sur l’apprentissage profond [Minaee et al., 2022, Mo et al., 2022].

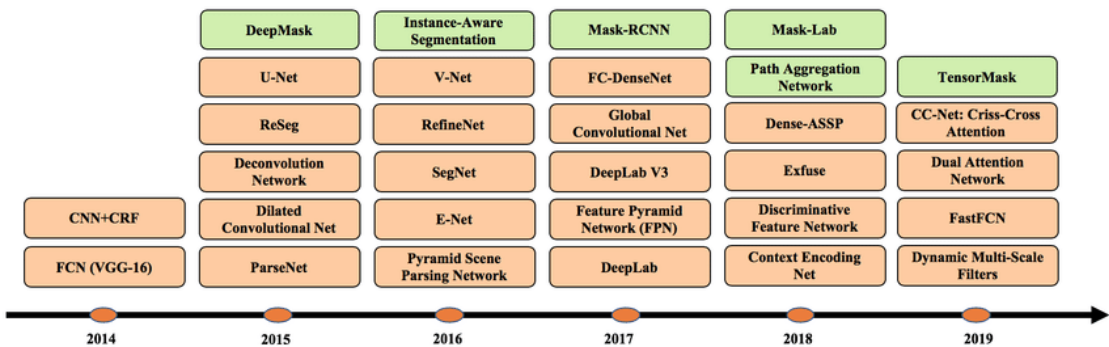


FIGURE 2.3 – Chronologie des architectures de segmentation d’images. Les blocs orange et verts indiquent respectivement les algorithmes de segmentation sémantique et d’instance [Minaee et al., 2022].

2.1.1 Biomédical et microscopie électronique cellulaire

Dans le domaine biomédical, les architectures les plus courantes sont l’architecture U-Net [Ronneberger et al., 2015] et DeepLab (et ses évolutions, DeepLabV3+) [Chen et al., 2018a, Chen et al., 2018b]. L’architecture U-Net est rapidement devenue populaire pour plusieurs raisons :

- la tâche : la tâche de segmentation est l’une des tâches les plus communes en traitement d’images ;
- la simplicité : le modèle est facile à comprendre, à réimplémenter et à modifier ;
- les performances : l’article original présente un réseau performant avec peu de données d’entraînement grâce aux techniques d’augmentation de données ;
- le timing : l’article est sorti à un moment où les méthodes basées sur apprentissage profond devenaient de plus en plus utilisées.

Le principe de fonctionnement du U-Net est d’utiliser une architecture encodeur / décodeur avec des connexions (concaténations) entre les blocs de convolution d’un même niveau appelées « long skip connections ». Beaucoup de variantes du modèle U-Net ont été proposées, modifiant par exemple les blocs de traitement [He et al., 2016a, He et al., 2016b] ou les types de connexions [Chaurasia and Culurciello, 2017]. L’architecture générale reste la même, un modèle à encodeur / décodeur avec des connexions entre l’encodeur et le décodeur. La figure 2.4 représente schématiquement le réseau U-Net original.

Si l’on se réfère aux articles récents comparant architectures U-Net et DeepLab, les résultats sont similaires. Dans certains cas, l’un des deux modèles donne de meilleurs résultats mais il n’y a pas pour l’instant de consensus sur le choix définitif de l’une ou l’autre des architectures [Ahmed et al., 2020, Khan et al., 2020, Su et al., 2021, Guay et al., 2021a]. En particulier, l’architecture U-Net est très facilement modifiable ; l’encodeur comme le décodeur classique peuvent être aisément remplacés par des blocs plus complexes s’inspirant des avancées du domaine. Les articles mettant en avant DeepLabv3+ le font généralement par rapport au modèle U-Net classique de 2015 [Ronneberger et al.,

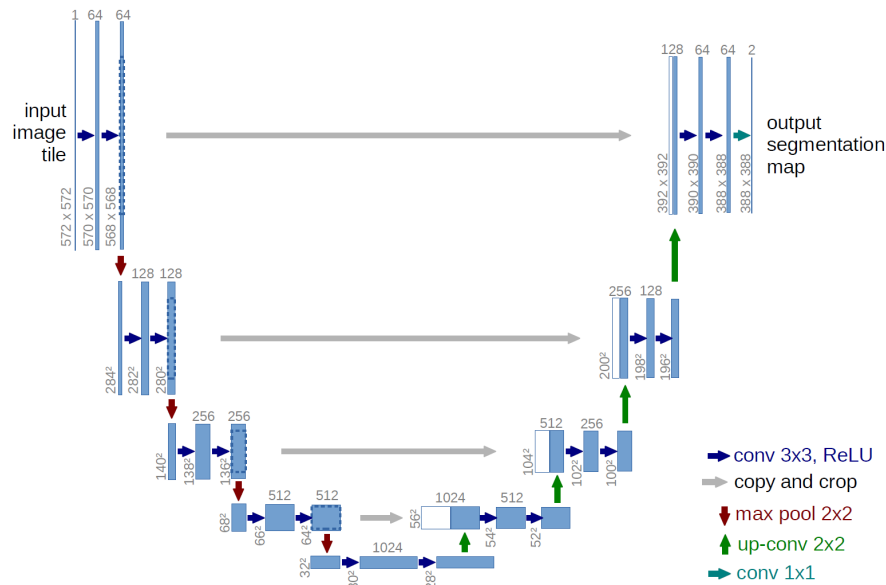


FIGURE 2.4 – Le modèle U-Net [Ronneberger et al., 2015].

2015]. Inversement, les articles mettant en avant un modèle U-Net par rapport à DeepLabv3+ présentent en général leurs propres améliorations.

Dans le domaine des **images biologiques de microscopie électronique**, le constat est le même [Moen et al., 2019, Heinrich et al., 2021] et les travaux effectués dans le domaine proposent différentes améliorations.

Le modèle U-Net [Ronneberger et al., 2015] propose originalement une application de segmentation comprenant des structures neuronales imagées par microscopie électronique. En particulier, les concaténations, aussi appelées « long skip connections » permettent de détecter des détails fins dans ces images. La version étendue en 3D [Çiçek et al., 2016] appelée 3D U-Net remplace les opérations 2D par des opérations 3D. Cette extension permet la segmentation de volumes en prenant en compte le contexte de toutes les dimensions spatiales. En plus de cela, les auteurs de [Çiçek et al., 2016] proposent d'utiliser des annotations faibles (sparse annotation). Les annotations faibles (annotations clairsemées) sont des annotations pour lesquelles seule une partie des pixels est annotée.

Les travaux de Zeng et al. sont centrés autour du challenge SNEMI3D, un problème de segmentation 3D des neurites sur des images de microscopie électronique [Zeng et al., 2017]. Les résultats des travaux permettent d'obtenir des scores très proches des performances humaines d'experts. La méthode se concentre sur la nature anisotrope des images. Le projet CDeep3M [Haberl et al., 2018] propose un système « plug and play » pour la segmentation automatique d'images, y compris le noyau, la membrane et les mitochondries dans des images de microscopie électronique en reprenant les travaux de [Zeng et al., 2017].

Xiao et al. segmentent automatiquement les mitochondries dans des images FIB-SEM et ATUM-SEM [Briggman and Bock, 2012] en utilisant une variation de U-Net 3D utilisant des modules de convolutions résiduelles [Xiao et al., 2018]. Les images FIB-SEM concernées sont isotropes, mais la segmentation se focalise uniquement sur les mitochondries. L'architecture utilisée ainsi que les variations sont représentées dans la figure 2.5. Ce furent les travaux les plus proches de notre problématique lorsque la thèse commença.

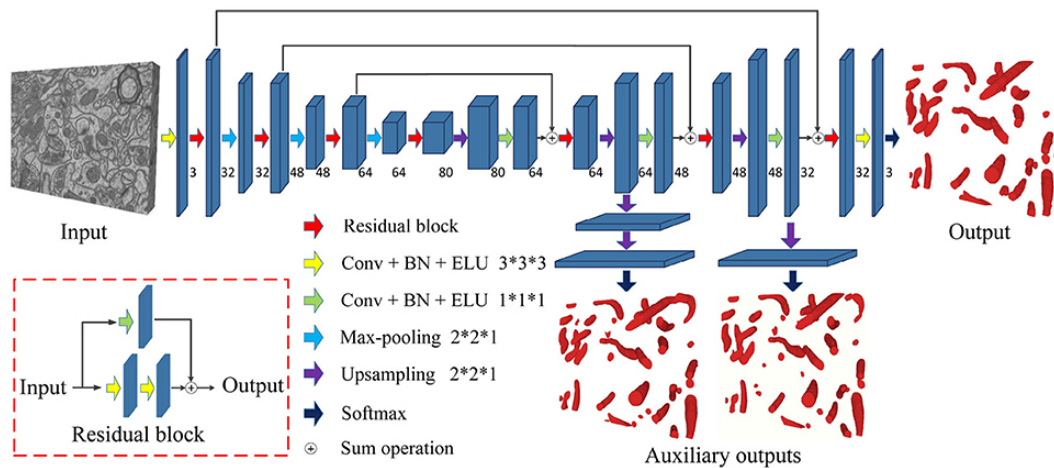


FIGURE 2.5 – Le modèle présenté par [Xiao et al., 2018].

Les travaux de thèse ont commencé fin 2019. Depuis, de nouveaux travaux ont été publiés sur le sujet, se rapprochant encore plus de notre problématique. Liu et al. séparent les tâches de segmentation des différentes organelles [Liu et al., 2020]. Les mitochondries sont segmentées à l'aide d'une architecture Mask R-CNN [He et al., 2020] et d'un backbone FPN [Lin et al., 2017] alors que pour le réticulum endoplasmique, c'est une architecture U-Net avec backbone ResNet-50 [He et al., 2016a] qui est utilisé. Dans leur article [Heinrich et al., 2018], les auteurs proposent une modification du problème de classification en régression en utilisant des transformations de distances pour la segmentation de synapses en microscopie électronique. Heinrich et al. continuent ces travaux avec le projet OpenOrganelle, un projet pour la segmentation automatique de 35 organelles pour des images FIB-SEM (figure 2.6), basés sur des modèles U-Net 3D [Heinrich et al., 2020, Heinrich et al., 2021]. A l'heure actuelle, c'est cette dernière approche qui représente la meilleure méthode connue pour la segmentation sémantique d'organelles en microscopie électronique quasi isotrope.

Tous ces travaux montrent un intérêt certain pour la segmentation d'organelles en microscopie électronique, ainsi qu'une volonté d'amélioration des méthodes. De telles segmentations peuvent être produites de manière efficace, mais les travaux sur les organelles complexes comme le réticulum endoplasmique ou l'appareil de Golgi, restent peu nombreux et les problèmes de variations entre images peu évoqués. Dans la majorité des articles, les auteurs s'intéressent à une seule image, ou alors, lorsque plusieurs images sont concernées, elles sont toutes issues d'un même échantillon. Dans les faits, la

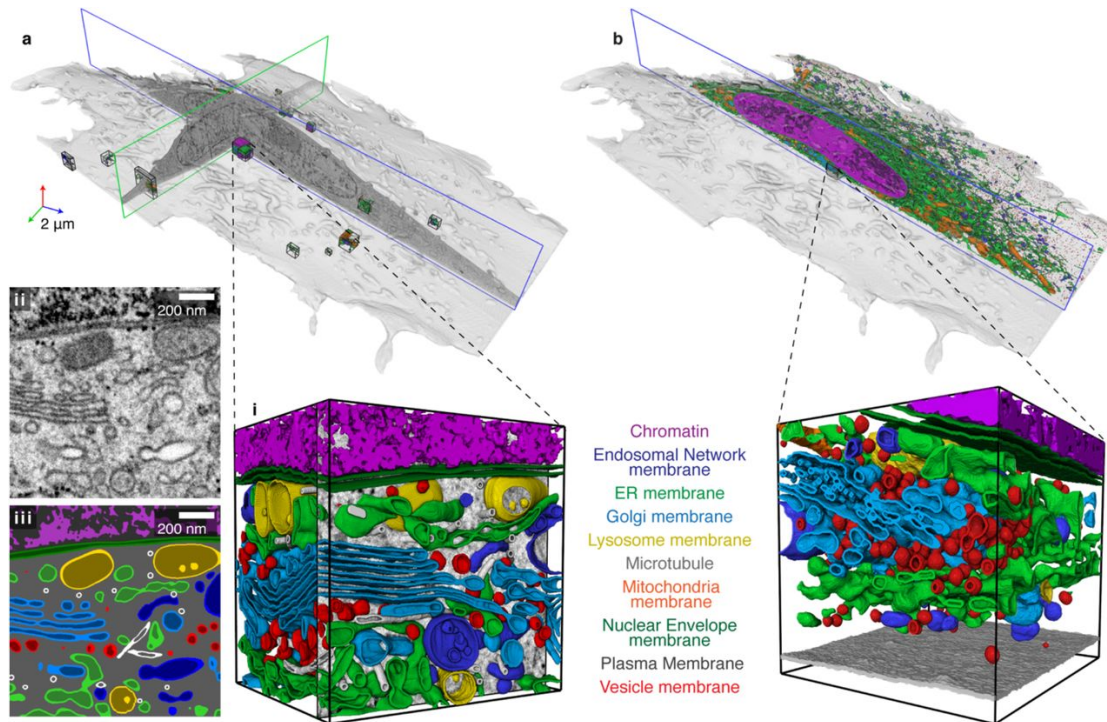


FIGURE 2.6 – Données d’entraînements (a) et prédiction (b) du projet OpenOrganelle [Heinrich et al., 2020].

variation entre les images est l’une des difficultés majeures si l’on souhaite développer des modèles robustes. Concernant les organelles segmentées, la majorité des travaux se concentrent sur les parois cellulaires (délimitation des cellules), les mitochondries ou les connexions synaptiques. De fait, si l’on s’intéresse à l’ultrastructure complète de la cellule, on remarque que la segmentation est plus difficile sur des objets d’apparence plus fine, complexe (composés de sous-éléments variés) ou des objets moins présents dans la cellule.

Si de nombreuses architectures sont présentes dans la littérature, l’architecture U-Net reste une référence proposant de bons résultats. Dans [Falk et al., 2019] les auteurs présentent un plugin ImageJ, livré avec des modèles U-Net pré-entraînés pour la segmentation de cellules, augmentant encore l’accessibilité de ces méthodes aux non-experts. Cette architecture a l’avantage d’être modulable, permettant des modifications sur les couches de traitements ou le changement de dimensions. Les auteurs de [Abdollahzadeh et al., 2021] ont écrit dans leur publication : « *We selected the U-Net architecture because it is widely used for the semantic segmentation of biomedical image volumes, resulting in precise segmentation and not requiring many annotated training* ». Cette architecture permet également d’envisager des améliorations de résultats grâce aux différentes modifications pouvant être apportées.

Kugelman et al. proposent une comparaison de plusieurs architectures U-Net [Kugelman et al., 2022] (U-Net [Ronneberger et al., 2015], Dense U-Net [Huang et al., 2017, Cai et al., 2020], Inception U-Net [Szegedy et al., 2015], Attention U-Net [Oktay et al., 2018], Residual U-Net [He et al., 2016a, He et al., 2016b], R2 U-Net [Alom et al., 2019], U-Net++ [Zhou et al., 2018], et SE U-Net [Hu et al., 2018, Roy et al., 2018, Roy et al., 2019]) pour la segmentation d'images de tomographie en cohérence optique (OCT pour Optical Coherence Tomography). Les différentes architectures sont représentées dans la figure 2.7. Cette étude propose une comparaison objective des différentes architectures, et montre que les améliorations des modèles plus complexes sont rarement significatives si on les rapporte aux augmentations de complexités des modèles « *State-of-the-art models do not appear to provide a clear benefit for this application while increasing the number of layers in each model resulted in small performance gains but, again, with a trade-off with respect to complexity and speed* » [Kugelman et al., 2022].

Nous avons décidé de sélectionner l'architecture U-Net (2D et 3D) pour nos travaux, tout en envisageant certaines améliorations n'impactant pas la complexité ou le temps d'apprentissage de manière significative.

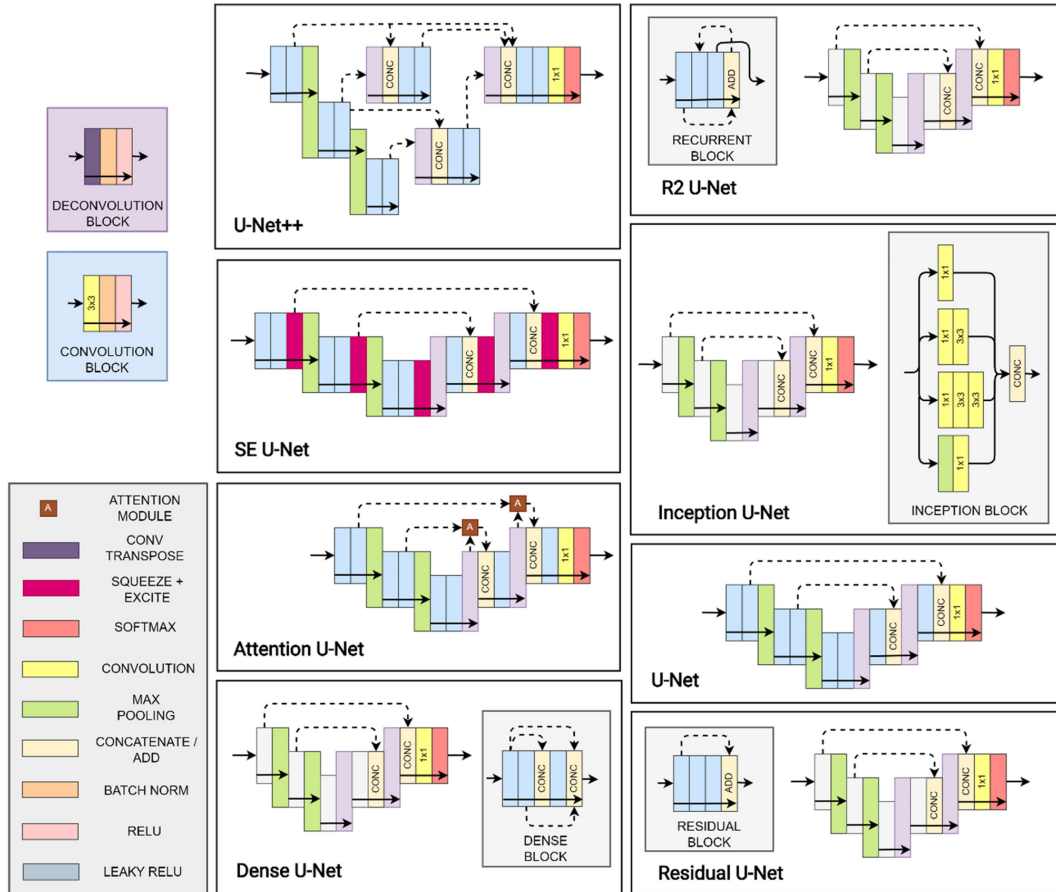


FIGURE 2.7 – Les architectures U-Net comparées par [Kugelman et al., 2022].

En plus des architectures, une composante importante permettant l'apprentissage d'un modèle est **la fonction de perte** qui est optimisée. Pour la segmentation d'images biomédicales, les approches les plus communes utilisent des mesures d'entropie croisée [Ronneberger et al., 2015, Çiçek et al., 2016, Xiao et al., 2018] ou des mesures de comparaison de région comme le coefficient de Dice [Sudre et al., 2017]. Il existe également des approches basées sur les distances aux contours [Heinrich et al., 2018] qui ont montré de bons résultats pour les tâches de segmentation de l'espace inter-synaptique ou des organelles [Heinrich et al., 2021]. La fonction proposée utilise l'erreur quadratique moyenne, mais transforme la carte binaire de segmentation en une carte de distance signée (< 0 pour l'extérieur, > 0 pour l'intérieur). La carte de distance est ensuite divisée par un facteur fixe puis on applique à ces valeurs la fonction \tanh sur la carte afin de saturer les résultats entre -1 et 1 . Cette fonction transforme le problème de segmentation en une régression de la distance aux contours.

Les auteurs de [Jadon, 2020] proposent une étude de différentes fonctions de perte pour la segmentation sémantique de lésion dans des images CT Scan [Puccio et al., 2016] et proposent une fonction basée sur le coefficient de Dice, la Log-Cosh Dice Loss. L'article présente une amélioration des performances, mais la différence est faible et la non-répétition des expériences ne montre pas une amélioration significative.

Définissons les différentes fonctions de perte utilisées dans la thèse. Premièrement, on définit une image de dimension n comme une fonction $f : E \rightarrow T$, où E est l'espace des points (\mathbb{Z}^n ou \mathbb{R}^n) et T l'échelle des niveaux de gris, inclus dans \bar{R} . On suppose que f est nulle en dehors d'une zone $E_f = \llbracket 1, w \rrbracket \times \llbracket 1, h \rrbracket \times \llbracket 1, d \rrbracket \subset \mathbb{Z}^3$ avec w , h et d des entiers positifs. On note $|E_f| = w \cdot h \cdot d$, le cardinal de l'image, le nombre d'éléments de l'image. Par souci de simplicité, on ne définit que le cas des images 3D ($n = 3$) et on identifie les images 2D au cas particulier où $d = 1$.

Dans la suite du chapitre, on utilisera $x : E \rightarrow [0, 1]$ pour représenter une prédiction, c'est-à-dire la sortie du réseau de neurones et $y : E \rightarrow \{0, 1\}$ la segmentation de référence correspondante. x et y sont de même taille et définis sur le même support E_0 , à noter que y est une image binaire.

Soit c le nombre de classes, la fonction vectorielle $\mathbf{x} : E_0 \rightarrow [0, 1]^c$ représente l'ensemble des prédictions pour les c classes. La fonction vectorielle $\mathbf{y} : E_0 \rightarrow \{0, 1\}^c$ représente l'ensemble des segmentations de référence pour les c classes. La $i^{\text{ème}}$ composante de \mathbf{x} est noté \mathbf{x}_i et celle de \mathbf{y} est noté \mathbf{y}_i .

Soit w un vecteur de longueur c dont les valeurs correspondent aux poids pour chaque classe. Si l'on ne souhaite pas pondérer les classes, les valeurs du vecteur w sont fixées à $\frac{1}{c}$. Soit ε une constante de valeur faible, par exemple 10^{-9} .

On note L_{CE} la fonction de perte de l'entropie croisée (CE pour cross-entropy) telle que :

$$L_{CE}(\mathbf{x}, \mathbf{y}, w) = \sum_{i=1}^c w_i \cdot L_{\log}(\mathbf{x}_i, \mathbf{y}_i) \quad (2.1)$$

avec

$$L_{\log}(x, y) = -\frac{1}{|E_0|} \sum_{p \in E_0} y(p) \log(x(p) + \varepsilon) \quad (2.2)$$

On note L_{Dice} la fonction de perte de Dice. L'équation (2.4) correspond au cas binaire et l'équation (2.3) au cas général où on calcule la moyenne pondérée des Dice binaires pour chaque classe. On définit l'intersection entre les images x et y comme le produit scalaire.

$$L_{Dice}(\mathbf{x}, \mathbf{y}, w) = \sum_{i=1}^c w_i \cdot L_{DiceBin}(\mathbf{x}_i, \mathbf{y}_i) \quad (2.3)$$

avec

$$L_{DiceBin}(x, y) = 1 - \frac{2 \cdot \sum_{p \in E_0} (x(p) \cdot y(p)) + \varepsilon}{\sum_{p \in E_0} (x(p) + y(p)) + \varepsilon} \quad (2.4)$$

On note L_{IoU} (équation (2.5)) la fonction de perte de l'intersection sur l'union.

$$L_{IoU}(\mathbf{x}, \mathbf{y}, w) = \sum_{i=1}^c w_i \cdot L_{IoUBin}(\mathbf{x}_i, \mathbf{y}_i) \quad (2.5)$$

avec

$$L_{IoUBin}(x, y) = 1 - \frac{\sum_{p \in E_0} (x(p) \cdot y(p)) + \varepsilon}{\sum_{p \in E_0} (x(p) + y(p)) - \sum_{p \in E_0} (x(p) \cdot y(p)) + \varepsilon} \quad (2.6)$$

L'erreur quadratique moyenne sur la transformation de distance signée précédemment détaillée [Heinrich et al., 2018] est notée L_{Dist} et définie dans l'équation (2.7). Pour utiliser cette fonction de perte, la fonction d'activation finale du réseau doit être telle que l'intervalle $[-1, 1]$ soit inclus dans son ensemble image. En pratique, on utilise la fonction \tanh . On redéfinit alors x et \mathbf{x} tel que $x : E \rightarrow [-1, 1]$ et $\mathbf{x} : E_0 \rightarrow [-1, 1]^c$.

$$L_{Dist}(\mathbf{x}, \mathbf{y}, w) = \sum_{i=1}^c w_i \cdot L_{MSE}(\mathbf{x}_i, DT(\mathbf{y}_i)) \quad (2.7)$$

avec L_{MSE} l'erreur quadratique moyenne et DT la transformation de distance de la segmentation de référence :

$$L_{MSE}(x, y) = \frac{1}{|E_0|} \sum_{p \in E_0} (x(p) - y(p))^2 \quad (2.8)$$

$$DT(y) = \begin{cases} \tanh(EDT(y)/s_{in}), & \text{pour } y = 1 \\ \tanh(-EDT(y)/s_{out}), & \text{pour } y \neq 1 \end{cases} \quad (2.9)$$

avec EDT la transformation de distance euclidienne [Danielsson, 1980] et s_{in} , s_{out} les paramètres de transformation de la pente intérieure et extérieure de la segmentation de référence.

2.1.2 Segmentation interactive d'images

La segmentation interactive a pour objectif de segmenter un objet dans une image à l'aide d'interactions utilisateurs. On parle également de séparation entre premier plan et arrière-plan. Dans notre problématique, nous avons expliqué que la variation entre images est un frein important à la segmentation automatique des images. Lorsque l'on ne possède pas d'annotation pour une image, les annotations issues d'autres images ne sont pas toujours suffisantes pour la traiter correctement. La segmentation interactive, guidée par l'utilisateur pourrait permettre de s'adapter plus facilement à ces changements entre images.

Les techniques de segmentation interactive peuvent être classées de deux manières différentes : en fonction du type d'interaction ou en fonction de la méthodologie utilisée. Une seconde définition plus large est parfois utilisée, la segmentation interactive regroupe toutes les méthodes de segmentation où des interactions de l'utilisateur permettent de guider une segmentation. Certaines méthodes utilisent des interactions afin d'améliorer d'autres résultats (par exemple ajustement de contours à la main) ou pour effectuer une segmentation sémantique complète de l'image.

Les types d'interactions les plus communes sont les suivantes :

(a) des points (seeds) sur les bords de l'objet (b) des points pour identifier le premier plan (+) et l'arrière-plan (-) (c) des traits (scribbles) pour identifier le premier plan (+) et l'arrière-plan (-) (d) région d'intérêt libre (on entoure l'objet) (e) région d'intérêt rectangulaire (on entoure l'objet) (f) région d'intérêt rectangulaire accolée (on place les bords de la zone d'intérêt sur les bords de l'objet)

La figure 2.8 représente ces différents types d'interactions.

Les familles de méthodes les plus communes sont basées sur les méthodologies suivantes : (a) coupe de graphe (graph cut) [Boykov and Jolly, 2001] (b) marche aléatoire (random walk) [Grady, 2006] (c) basées sur les contours [Liu et al., 2010, Nguyen et al., 2012] (d) fusion de région (region merging) et croissances de région (region growing) [Ning et al., 2010, Zhou et al., 2016] (e) apprentissage profond et CNN [Xu et al., 2016, Maninis et al., 2018]

Un article de revue [Ramadan et al., 2020] détaille ces méthodes, leurs avantages et inconvénients. Nous nous concentrerons sur les méthodes basées sur l'apprentissage profond, proposées à partir de 2016 [Xu et al., 2016].

Les jeux de données habituellement utilisés concernent des images 2D d'objets ou de scène de la vie courante. Il n'existe pas à notre connaissance de jeux de données sur des images 3D, des images de microscopie électronique ou biomédicales. Néanmoins, il est tout à fait possible de générer des interactions à partir d'un jeu de données de segmentation d'instance ou de segmentation sémantique si les objets sont correctement délimités (pas de fusion aux bords).

Les premières méthodes utilisant l'apprentissage profond à avoir été proposées sont celles de [Xu et al., 2016] et de [Lin et al., 2016].

Pour la première [Xu et al., 2016], il est proposé de transformer les points positifs et négatifs (interaction b) en carte de distance et de concaténer ces deux cartes de distance



FIGURE 2.8 – Exemple des différents types d'interactions [Ramadan et al., 2020].

au canal des couleurs (figure 2.9). Le résultat (carte de probabilité) est ensuite transformé en segmentation en utilisant une coupe de graphe. Les interactions de l'utilisateur sont générées avec différentes stratégies pour simuler différents comportements d'annotateurs.

La seconde méthode, [Lin et al., 2016] propose d'utiliser des scribbles (interaction c) afin d'annoter les différents éléments de l'image (multi-classes fixés à l'avance). Cette méthode s'éloigne de la première définition de la segmentation interactive, et de la séparation binaire entre un objet et le fond.

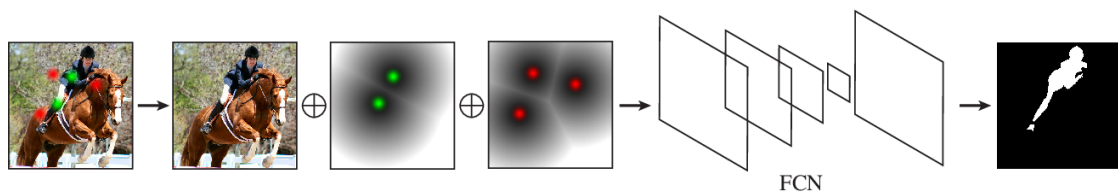


FIGURE 2.9 – Framework proposé par [Xu et al., 2016].

Par la suite, de plus en plus de travaux se sont intéressés aux méthodes utilisant l'apprentissage profond.

En 2017, Xu et al. modifient leur première méthode afin d'utiliser un rectangle comme annotation (interaction e) [Xu et al., 2017]. Ils gardent le même principe de fonctionne-

ment reposant sur les cartes de distances, mais suppriment leur post-processing utilisant une coupe de graphe.

Dans l'article [Liew et al., 2017] (interaction b), les auteurs appellent leur modèle RIS-Net. Ils proposent d'utiliser deux branches. Une première proposant une segmentation grossière sur toute l'image basée sur [Xu et al., 2016], et une seconde proposant des raffinements locaux autour des points de l'utilisateur. Les deux segmentations sont ensuite combinées et segmentées à l'aide d'un graph cut. Le modèle pour la segmentation grossière est appelé DeepLab-LargeFOV [Chen et al., 2018a].

Les auteurs de [Maninis et al., 2018] utilisent la méthode d'annotation de [Papadopoulos et al., 2017] qui utilise des points extrêmes (variation interaction f, figure 2.10). Seule la sous-partie de l'image dans la boîte englobante délimitée par les points extrêmes (avec un agrandissement de quelques pixels) est utilisée pour la segmentation. Le reste de la méthode est assez proche de [Xu et al., 2016] ; les points sont transformés en carte de chaleur avec des gaussiennes 2D et ajoutés aux canaux de couleur. Le CNN utilisé est un ResNet-101 modifié : suppression de la couche fully connected et des deux derniers max poolings, ajouts de convolutions à trous, (Figure 2.2b) et d'un module de « pyramid scene parsing » [Zhao et al., 2017] afin d'ajouter du contexte global.

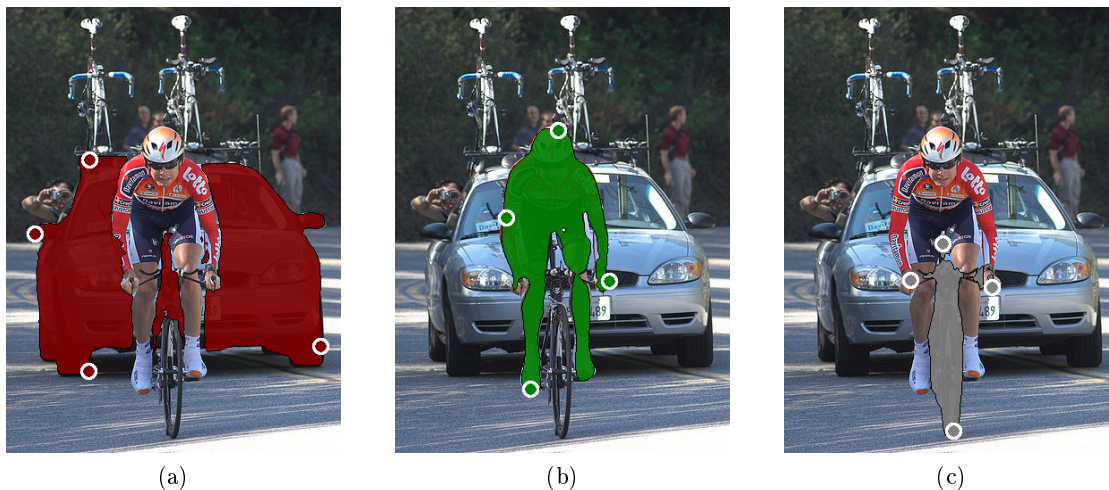


FIGURE 2.10 – Résultats de segmentation de la méthode DEXTR [Maninis et al., 2018].

Dans leurs travaux [Agustsson et al., 2019] souhaitent comme [Lin et al., 2016] segmenter tous les éléments de l'image. Une première étape utilise des points extrêmes [Papadopoulos et al., 2017, Maninis et al., 2018] afin de proposer une segmentation. L'utilisateur corrige les zones mal annotées (interaction c) de manière itérative, jusqu'à arriver à une segmentation satisfaisante. La méthode se base sur un modèle Mask RCNN [He et al., 2020] pour effectuer une segmentation à partir de région d'intérêt. Les résultats dans le cas spécifique à une seule région sont similaires à [Maninis et al., 2018].

Li et al. (interaction b) se concentrent sur le problème suivant : lorsque l'utilisateur clique sur une porte, souhaite-t-il une segmentation de la porte ou de toute la maison [Li

et al., 2018b]. Pour résoudre ce problème, ils proposent d'utiliser deux modèles. Un premier qui propose des segmentations possibles en fonction des interactions utilisateur et un second qui sélectionne la segmentation appropriée. Pour le premier, ils se basent sur la méthode de [Xu et al., 2016], en ajoutant en entrée les couches intermédiaires d'un VGG19 [Simonyan and Zisserman, 2015] pré-entraîné sur ImageNet [Deng et al., 2009]. Pour obtenir des propositions de segmentation diversifiées, une fonction de coût contraint les résultats. Le second modèle utilise l'image, les interactions et les résultats du précédent réseau en entrée. Il est entraîné à proposer la segmentation avec la plus petite distance de Jaccard à la vérité terrain. Les résultats montrent une amélioration importante par rapport aux méthodes précédentes.

Jang et Kim améliorent les travaux précédents en proposant une « Backpropagating Refinement Scheme » (BRS) qui vérifie et corrige les erreurs de segmentation sur des zones annotées [Jang and Kim, 2019]. Les BRS sont basés sur l'idée que l'on peut remonter à travers le réseau à l'envers et modifier les poids ou entrées pour arriver à un certain résultat. Le modèle est composé d'un encodeur DenseNet avec des « squeeze and excitation module » [Hu et al., 2018] et de deux décodeurs, un premier grossier et un second plus fin. Une fois la première segmentation obtenue (première passe en avant), la carte d'interactions est modifiée de façon à s'assurer que les points utilisateur sont correctement classifiés par le modèle.

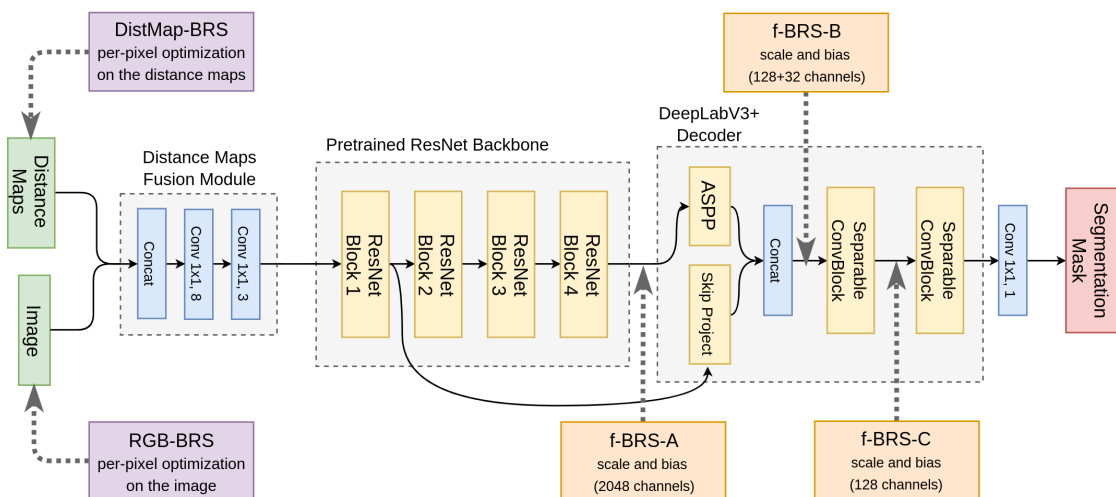


FIGURE 2.11 – DistMap-BRS [Jang and Kim, 2019] et f-BRS [Sofiuk et al., 2020].

Dans [Sofiuk et al., 2020], les auteurs modifient les travaux de [Jang and Kim, 2019] en proposant de ne pas remonter à travers tout le réseau, mais uniquement sur une petite partie (figure 2.11). Leur modèle est composé d'un ResNet pré-entraîné et d'un décodeur DeepLabV3+. C'est l'une des méthodes les plus efficaces actuellement. Les auteurs de [Sofiuk et al., 2022] critiquent les méthodes utilisant l'optimisation lors de l'inférence qui sont plus lourdes en temps de calcul. Ils proposent plusieurs modifications aux modèles de type [Xu et al., 2016] :

- remplacer l'encodage des clics qui utilise une transformée de distance par une transformation en disque à rayon fixe ;
- utiliser une « Distance Maps Fusion » (DMF) pour fusionner les différents canaux d'entrées dans un réseau pré-entraîné sur des images RGB (c'est une réaction au fait que la plupart des méthodes se basent sur des réseaux pré-entraînés sur ImageNet).

Ils utilisent ensuite une stratégie itérative pour la simulation des interactions. Une fonction de coût spécifique, la perte focale normalisée (Normalized Focal Loss) est proposée pour remplacer l'entropie croisée binaire utilisée habituellement. D'après leurs résultats, cette méthode semble être la plus performante à l'heure actuelle.

Une comparaison entre la plupart des méthodes citées est faite par [Sofiuk et al., 2020]. Il est important de noter que ces méthodes de segmentation interactive ne rentrent pas dans le cadre de l'apprentissage actif (active learning) où l'algorithme d'apprentissage peut interroger de manière interactive l'utilisateur. Également, il ne faut pas confondre segmentation interactive et la segmentation faiblement supervisée, qui est entraînée avec des données plus rapides à produire (bounding box, scribbles, etc.).

Toutes les méthodes précédemment citées se concentrent sur des problèmes de segmentation interactive d'objets ou de scène de la vie courante. Les techniques sont conçues pour segmenter des images 2D et non 3D. Dans la suite de cet état de l'art, on se concentre sur les articles avec une application sur des **images 3D, biologiques ou biomédicales**.

Les méthodes interactives ont déjà été appliquées aux images biomédicales, notamment les méthodes utilisant les graphes [Chen and Pan, 2018].

Wang et al. partent de l'idée suivante : les CNN fonctionnent très bien pour segmenter automatiquement des images biomédicales, mais ne sont pas suffisamment robustes et exactes pour des utilisations cliniques [Wang et al., 2019]. Ils souhaitent améliorer la robustesse et l'adaptation à de nouvelles images et de nouveaux objets. Ils proposent d'utiliser un premier CNN pour faire une segmentation automatique (appelé P-Net) et un second pour raffiner la segmentation à l'aide d'interaction utilisateur (R-Net) (interaction c). Les deux CNN (P-Net et R-Net, figure 2.12) sont connectés avec des champs aléatoires conditionnels (CRF pour conditional random field) (en réalité, CRF as RNN [Zheng et al., 2015]) pour « améliorer la cohérence spatiale et pour utiliser les annotations comme contrainte forte ». Les entrées du R-Net sont composées de l'image, de la segmentation initiale, ainsi que de deux cartes de distance. La transformation de distance géodésique est utilisée pour transformer les interactions en carte de distance.

Dans [Wang et al., 2018a] les auteurs utilisent une combinaison de scribbles et de bounding box (c + f) comme annotations. Trois applications sont proposées : segmentation multi-organe d'image 2D (fetal magnetic resonance slices) et deux tâches de segmentations 3D de tumeurs cérébrales. La différence mise en avant par rapport à [Wang et al., 2019] est l'adaptation à de nouveaux objets jamais vus auparavant. Un premier modèle (P-Net [Wang et al., 2019]) apprend à segmenter des objets à l'aide d'une bounding box (interaction f), et un second modèle de raffinement peut (optionnellement) utiliser des scribbles (interaction c) pour améliorer les résultats. Ils introduisent également une fonction de coût pondérée, qui utilise les interactions utilisateurs pour modifier l'importance de chaque pixel dans le calcul de la fonction de coût.

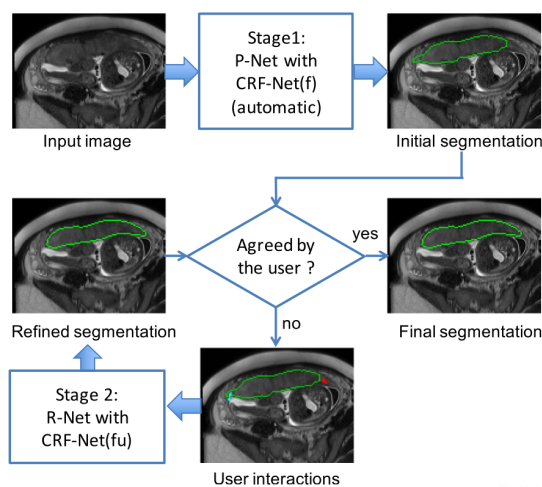


FIGURE 2.12 – Aperçu de la méthode interactive proposée par [Wang et al., 2019]. Le modèle P-Net propose une segmentation initiale raffinée par le modèle R-Net avec des interactions utilisateurs mettant en avant les erreurs de segmentation.

À notre connaissance, la segmentation interactive basée sur l'apprentissage profond ne semble pas avoir été utilisée en microscopie électronique cellulaire.

Dans nos expérimentations, nous évaluerons l'utilisation de segmentation interactive en commençant par les travaux de Xu et al. [Xu et al., 2016]. Cette méthode, bien que limitée et ne permettant certainement pas d'atteindre les meilleurs résultats possibles, nous permet néanmoins de comparer efficacement segmentation interactive et segmentation automatique entraînée sur des annotations faibles.

Dans cette section, nous nous sommes intéressés à la réduction du temps d'annotation. Intéressons-nous désormais à l'amélioration de la qualité et de la cohérence des segmentations produites.

2.2 Apprentissage profond et morphologie mathématique

La morphologie mathématique est une théorie pour l'analyse de structures basée sur la théorie des ensembles, la géométrie intégrale et la théorie des treillis [Soille, 2004]. L'origine de la discipline remonte au milieu des années 1960, suite aux travaux de Georges Matheron et Jean Serra. Depuis, de nombreux travaux théoriques et applicatifs ont permis le développement d'outils variés de traitement d'image.

2.2.1 Les réseaux de neurones morphologiques

Les réseaux de neurones traditionnels ont été utilisés avec succès pour résoudre différents problèmes de traitement d'image : classification, segmentation, détection d'objet, débruitage, etc. Ces réseaux traditionnels utilisent des unités de traitement linéaire, suivi

de fonctions d'activation non-linéaires. Ce sont ces fonctions d'activations qui donnent au réseau de neurones leurs propriétés non-linéaires. L'une des motivations des réseaux de neurones morphologiques est de remplacer les opérations linéaires de somme des multiplications par des opérations non-linéaires de maximum des additions.

Les réseaux de neurones morphologiques ont été décrits presque simultanément de deux manières différentes, par Wilson [Wilson, 1989] et par Davidson et Ritter [Davidson and Ritter, 1990].

Les réseaux morphologiques de Wilson [Wilson, 1989] sont formulés de manière générale autour des filtres de rangs, où les opérations maximum et minimum sont des cas particuliers. Cette vision a inspiré la création d'architecture hybride entre unité linéaire et filtres de rangs, les MRNNs (morphological / rank neural networks) [Pessoa and Maragos, 1996], MRL-filters (morphological / rank / linear) [Pessoa and Maragos, 1998] et MRL-NN (morphological / rank / linear neural networks) [Pessoa and Maragos, 2000]. Les MRNN sont des réseaux de neurones où chaque neurone est une unité de traitement morphologique. Les MRL-filters correspondent à une combinaison linéaire entre un filtre morphologique (ou filtre de rang) et un filtre linéaire. Les MRL-NN sont des réseaux composés de MRL-filters. Les MRL-NN sont très généraux, un MLP est un cas spécifique de MRL-NN et un MRNN est un cas spécifique de MRL-NN. Ces différents modèles et leurs filtres peuvent être optimisés par l'algorithme du gradient stochastique et la rétro-propagation du gradient, mais leur utilisation ne s'est pas démocratisée et peu de résultats sont disponibles en dehors des travaux de Pessoa et Maragos.

Les réseaux de neurones morphologiques de Davidson [Davidson and Ritter, 1990, Davidson and Hummer, 1993] utilisent des unités de traitement qui sont des dilations et des érosions. L'idée est de remplacer les opérations de base des neurones par des opérations non-linéaires. La somme est remplacée par une opération maximum et l'opération de multiplication des entrées par les poids, par une addition (mathématiques tropicales, algèbre max-plus [Maclagan and Sturmfels, 2021]). Contrairement aux modèles présentés par Wilson qui « *utilisent l'opération habituelle de multiplication et de somme à chaque nœud* », ces modèles « *utilisent l'opération d'addition et maximum à chaque nœud* » [Davidson and Ritter, 1990]. On utilise l'opération de maximum des additions pour produire les opérations d'érosion et de dilatation. On peut interpréter géométriquement le résultat de chaque neurone morphologique comme une boîte englobante dans l'espace à N dimension des features [Sussner and Esmi, 2011], là où un perceptron sépare linéairement l'espace.

Différents travaux [Ritter and Sussner, 1996, Sussner, 1998, Sussner and Esmi, 2011] s'intéressent aux capacités de ces réseaux. Ils montrent que leur entraînement est très rapide, grâce à l'utilisation d'heuristiques et leurs résultats sont bons sur les données d'entraînement. Cependant, peu d'intérêt a été apporté à la généralisation de ces modèles [Sussner and Esmi, 2011, Zhang et al., 2019] ce qui les rend difficilement applicables.

Une extension, **les réseaux de neurones morphologiques dendrites** [Ritter and Urcid, 2003, Ritter et al., 2003, Barmpoutis and Ritter, 2007, Sossa and Guevara, 2014] se basent sur les avancées récentes de la neurobiologie et ajoutent à chaque neurone des dendrites. Chaque dendrite correspond à une entrée du neurone, pouvant être dans un

état d'inhibition ou d'excitation. Une variation existe, les réseaux de neurones morphologiques dendrites ellipsoïdales, remplaçant les hyper-boîtes par des hyper-ellipses [Arce et al., 2017]. [Zamora and Sossa, 2016] proposent une solution pour entraîner les réseaux de neurones morphologiques dendrites avec l'algorithme du gradient stochastique, qui améliore significativement la généralisation et les résultats sur les jeux de tests. Une comparaison récente entre réseaux de neurones classique et réseaux de neurones morphologique dendrite montre de bons résultats sur une classe de problème spécifique [Hernández et al., 2017], la classification multi-classes de point de spirales. Cependant, l'apprentissage est plus lent qu'avec un perceptron multicouche, et le problème des spirales reste un exemple jouet où les réseaux de neurones morphologiques dendrites sont particulièrement adaptés.

Réseaux de neurones morphologiques et algèbre max-plus, min-plus Une version revisitée du perceptron morphologique de Davidson [Sussner and Esmi, 2011] et basée sur l'algèbre max-plus, min-plus est proposée [Charisopoulos and Maragos, 2017]. D'autres travaux [Zhang et al., 2019] s'intéressent à ces réseaux, aux possibilités d'élagage de modèles ainsi qu'à la proximité entre réseaux max-plus élagué et réseaux maxout [Goodfellow et al., 2013].

2.2.2 Application de la morphologie mathématique en apprentissage profond

L'utilisation des réseaux de neurones morphologiques reste encore assez restreinte, mais les réseaux de neurones et les méthodes d'apprentissage profond font aujourd'hui partie intégrante des outils de traitement d'images. Une combinaison entre méthode historique comme la morphologie mathématique avec l'apprentissage profond n'est pas à exclure, et une amélioration des méthodes d'apprentissage profond pourrait venir d'une combinaison des deux domaines. Certains travaux récents tentent d'utiliser les principes de morphologie mathématique sans s'éloigner de l'apprentissage profond plus traditionnel, en voici quelques exemples.

Masci et al. proposent une méthode basée sur des couches PConv qui effectuent des pseudo-érosion et pseudo-dilatation. Une couche PConv est une couche de convolution avec un filtre dont les poids ainsi qu'un paramètre P sont appris. Le paramètre P contrôle le type d'opération ($P = 0$: convolution standard, $P < 0$: pseudo-érosion, $P > 0$: pseudo-dilatation) [Masci et al., 2013]. Ces opérations sont effectuées en utilisant la moyenne contre-harmonique (CHM pour counter-harmonic mean) [Angulo, 2010] afin de pallier le problème de non-différentiation des opérations max et min. Ces couches permettent d'étendre les CNN en MPCNN (Morphological Convolutional Neural Network) [Masci et al., 2013] tout en restant compatible avec les méthodes et techniques habituelles d'apprentissage profond (optimisation par SGD et backprop). Les résultats présentés montrent un modèle capable d'apprendre à effectuer les opérations d'érosion, de dilatation, d'ouverture, de fermeture (exemple figure 2.13), des transformations seuil haut de forme et de débruitage. Les résultats ne sont pas comparés à un réseau de neurones convolutif classique.



FIGURE 2.13 – Exemple d’apprentissage d’un opérateur de fermeture (élément structurant ligne de longueur 10 orienté à 45°) et d’ouverture (élément structurant carré de côté 5). Les noyaux L1 et L2 représentent l’état des poids des couches 1 et 2. [Masci et al., 2013].

Avec leurs travaux [Kirszenberg et al., 2021, Hermary et al., 2022] les auteurs proposent d’utiliser des opérations \mathcal{L} Morph basées sur la moyenne de Lehmer [Bullen, 2003] et \mathcal{S} Morph (Smooth Morphological). Des exemples de résultats d’apprentissage d’opération de dilatation et d’érosion sont visibles en figure 2.14. De leur côté, dans [Shih et al., 2019, Shen et al., 2022] les auteurs utilisent le soft-maximum et soft-minimum plutôt que la moyenne contre harmonique afin d’apprendre les éléments structurants d’opérations morphologiques.

Nogueira [Nogueira et al., 2021] utilise des convolutions depthwise et pointwise ainsi que des poolings depthwise pour reproduire des opérations d’érosion, de dilatation, d’ouverture, de fermeture et des approximations de reconstruction géodésiques avec un modèle appelé DeepMorphNet. Franchi [Franchi et al., 2020] proposent d’utiliser des opérations morphologiques exactes, le tout dans un système d’apprentissage profond complet de bout en bout. Il remplace le max-pooling par un morphological-pooling.

Enfin, une autre approche peut être utilisée afin de combiner **morphologie mathématique et apprentissage profond** : utiliser des opérateurs morphologiques avant ou après l’utilisation des images par un réseau de neurones. On peut utiliser les opérations dans le but d’améliorer une image, d’extraire des caractéristiques intéressantes pour l’analyse ou d’améliorer un résultat comme une segmentation.

On peut citer les exemples suivants, représentant différents types d’opérations utilisées et d’applications.

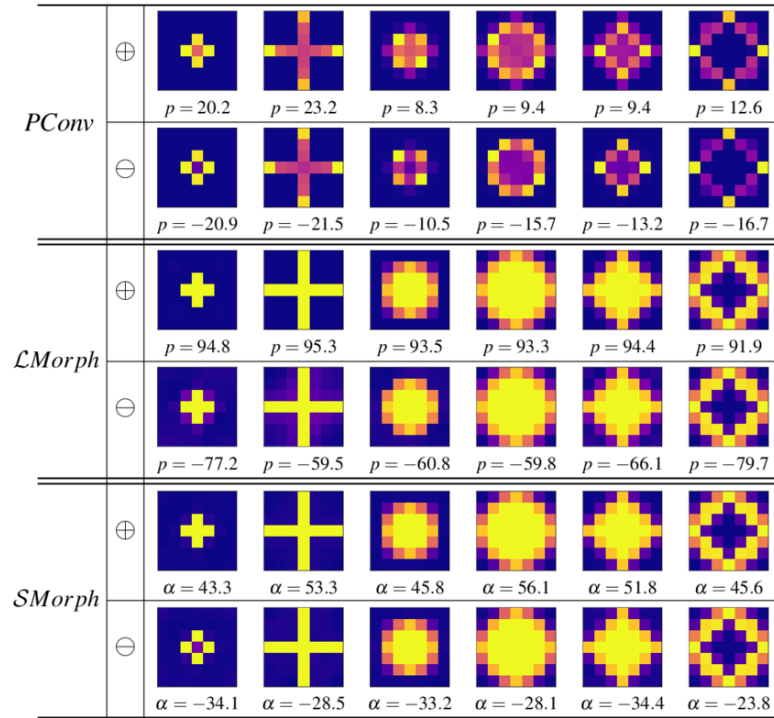


FIGURE 2.14 – Éléments structurants appris (avec p ou α correspondant après convergence) pour des couches *PConv*, *LMorph* et *SMorph* de dilatation \oplus et d'érosion \ominus [Kirszenberg et al., 2021].

- [Nasr-Esfahani et al., 2016] et [Selmi et al., 2017] utilisent des transformations chapeau haut de forme afin d'améliorer des images d'angiographie aux rayon-X [Nasr-Esfahani et al., 2016] et de plaque d'immatriculation [Selmi et al., 2017].
- [Nur-A-Alam et al., 2021] utilisent diverses opérations d'érosion et de dilatation pour améliorer des images d'empreintes digitales.
- [Wang et al., 2018b] utilisent une ouverture (érosion et dilatation) afin de supprimer les points isolés et de lisser les contours de segmentations de tumeurs rectales dans des images de résonance magnétique.

Dans leur article [Farfan Cabrera et al., 2021b] les auteurs effectuent une extraction de caractéristiques basée sur les arbres de composantes connexes [Salembier et al., 1998, Najman and Couprie, 2006] et développés initialement par Grossiord et al. [Grossiord et al., 2015, Grossiord et al., 2017]. Une représentation schématique du principe de fonctionnement proposé est disponible dans la figure 2.15.

C'est cette dernière approche, utilisant opération morphologique et opérateurs connexes pour extraire des caractéristiques et d'une certaine manière pré-traiter nos images que nous avons choisi d'expérimenter.

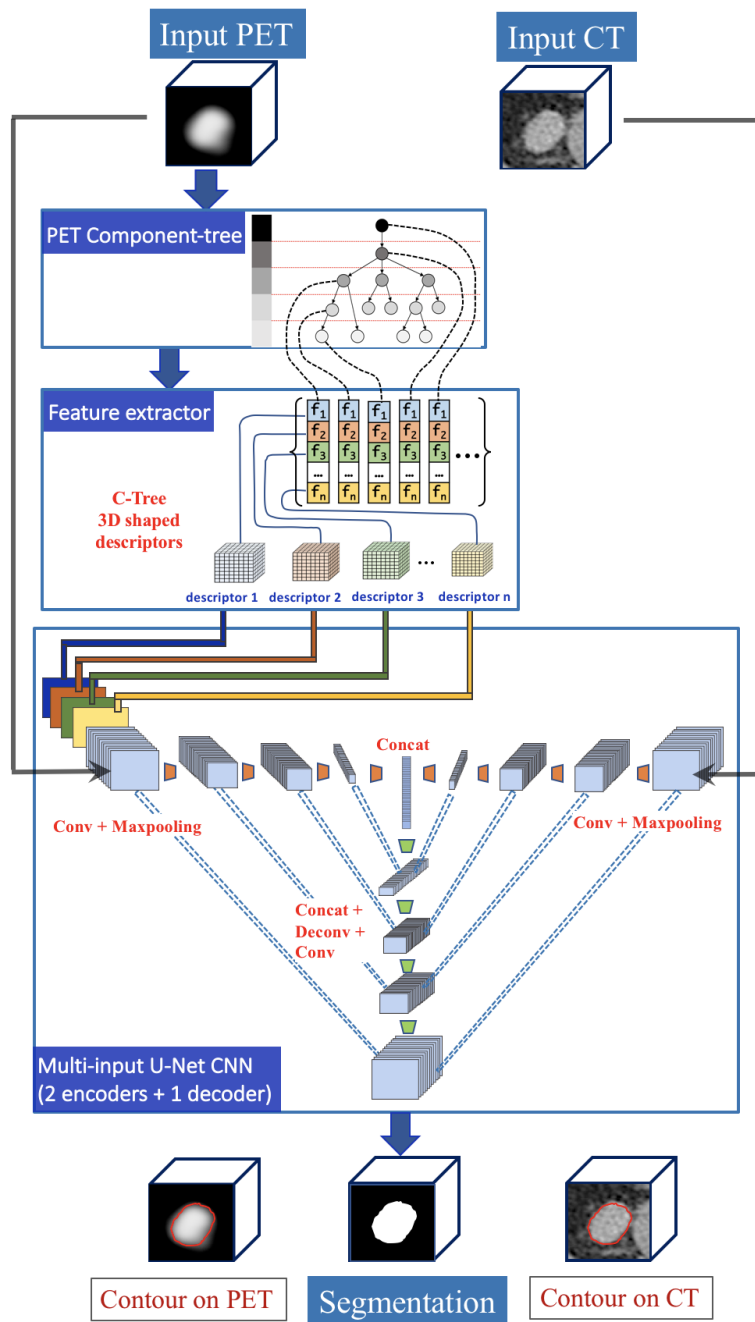


FIGURE 2.15 – Framework proposé par Farfan Cabrera et al. [Farfan Cabrera et al., 2021b].

Dans ce chapitre, nous avons fait un état des lieux des méthodes de segmentation utilisant l'apprentissage profond autour de notre problématique : la segmentation des organelles dans les images de microscopie électronique FIB-SEM cellulaires. Nous avons présenté les différentes méthodes existantes, leurs avantages, inconvénients. En conclusion de ce chapitre, voici un résumé des positionnements que nous prenons vis-à-vis de cet état de l'art.

- Premièrement, nous utiliserons l'architecture générale U-Net, tout en évaluant différentes modifications n'impactant pas la complexité ou le temps d'apprentissage de manière significative, le tout dans le cas binaire et multi-classes.
- Deuxièmement, nous évaluerons différents paramètres d'apprentissage.
 - Taille de patch pour l'entraînement.
 - Modèles 2D et 3D.
 - Fonctions de pertes.
 - Taille du jeu d'entraînement.
 - Modèles pré-entraînés.
- Ensuite, nous comparerons les approches interactives et utilisant des annotations faibles dans le but de réduire le temps utilisateur nécessaire à la création de données manuelles.
- Enfin, nous expérimenterons l'utilisation d'opérateurs morphologiques et d'opérateurs connexes afin de pré-traiter nos images dans le but d'améliorer la qualité de nos segmentations.

Données et évaluation

3.1	Données	37
3.1.1	Données internes	41
3.1.2	Données publiques	50
3.2	Évaluation	58
3.2.1	Mesure de recouvrement	58
3.2.2	Mesures de distance aux contours	61
3.2.3	Mesure de composantes connexes	62
3.2.4	Comparaison des mesures	62

Dans ce chapitre, nous présentons les différents jeux de données utilisés pour entraîner et évaluer nos modèles. Nous détaillons premièrement les images utilisées dans le cadre de la collaboration entre les laboratoires ICube, IGBMC et GoetzLab, puis nous listons les jeux de données publiques, proches de notre problématique.

Nous décrivons ensuite les différentes métriques utilisées pour évaluer la performance de nos modèles, et étudions comment elles reflètent les différents aspects de qualité des segmentations produites. Enfin, nous illustrons les propriétés des métriques sur de nombreux exemples.

3.1 Données

Dans cette première section, nous allons voir différents jeux de données. Un jeu de données est composé d'une ou plusieurs images et des annotations associées, aussi appelées segmentation de références.

Plusieurs informations sont importantes lorsque l'on s'intéresse aux images de microscopie électronique cellulaire :

- la **résolution de l'image** dans chacune des directions, qui correspond au nombre d'unités de distance entre chaque voxel ;

- le champ de vision correspond à la **taille** de la zone imagée. Pour l'axe Z c'est le nombre de tranches ou de sections 2D de l'image 3D. Pour simplifier, les valeurs de champ de vision sont données en voxels ;
- la méthode de préparation de l'échantillon, la **technique de fixation**, la **résine** utilisée, le type de **coloration** ;
- la **nature de l'échantillon**, l'objet qui a été imagé ;
- l'**instrument** utilisé, la méthode d'acquisition de l'image et ses paramètres.

Toutes les informations ne sont pas toujours disponibles, mais l'information la plus importante et requise est la résolution de l'image. Les objets d'intérêt ont une taille dans le monde réel qui correspond à un nombre de voxels pour une certaine résolution. Si la résolution change radicalement, le nombre de voxels représentant un objet changera tout autant. L'information sur l'instrument, la méthode d'acquisition et de préparation d'échantillon n'est pas une propriété directement utile pour le traitement des images, mais permet de mettre en relation ou d'éloigner des images de différents domaines.

Nous l'avons vu dans le chapitre 1, le fonctionnement du FIB-SEM consiste à découper de fines tranches de l'échantillon lors du processus de coupe sériée puis à imager la surface de la face révélée avec un microscope électronique à balayage. D'autres méthodes basées sur la microscopie électronique à balayage permettent d'obtenir des images volumiques, mais sans jamais atteindre la résolution Z du FIB-SEM [Briggman and Bock, 2012]. On peut citer la MEB de surface de bloc séquentielle (Serial Block-Face Scanning Electron Microscopy, SBEM) [Denk and Horstmann, 2004] qui consiste à utiliser un couteau en diamant afin d'enlever séquentiellement une couche d'environ 30 nm de l'échantillon puis d'imager la nouvelle surface ainsi créée, ou encore la MEB en série automatisée sur ruban (Automated Serial Sections to Tape Scanning Electron Microscopy, ATUM-SEM) [Hayworth et al., 2006, Schalek et al., 2012] qui place des tranches successives de l'échantillon, générées par un couteau en diamant, sur un ruban, permettant de ne pas détruire l'échantillon lors du processus de coupe sériée. Enfin, on peut également se reposer sur la microscopie électronique en transmission (MET) ou le faisceau d'électrons est transmis à travers l'échantillon. Cependant les images ne représentent plus la surface de l'échantillon à imager, mais une projection de la tranche. La méthode de MET par série de sections [Harris et al., 2006] (Serial Sectioning for Transmission Electron Microscopy, ssTEM) permet d'obtenir des images volumiques en découpant de fines tranches de manière séquentielle avant de les imager dans un MET.

Les résolutions typiques des différentes méthodes MEB sont listées dans la table 3.1. La résolution XY peut être bien meilleure en TEM (< 1 nm), mais la résolution utile, elle, est limitée par la méthode de préparation de l'échantillon. Même après cryo-substitution (définition page 40) des meilleures résolutions que 2 nm n'ont que peu de sens compte tenu de la conservation des structures présentes dans l'échantillon. La figure 3.1 présente les différents principes de fonctionnement des méthodes citées.

Beaucoup d'autres paramètres d'acquisition peuvent affecter la résolution et le rapport signal sur bruit des images en MEB. Par exemple, l'intensité du courant, le temps d'acquisition par pixel, la focalisation, le type de détecteur utilisé et le type d'électrons collectés sont tout un panel de paramètres affectant ces images.

Méthode	résolution X/Y	résolution Z
FIB-SEM	5	5
SBEM	10	25
ATUM-SEM	4	30

TABLE 3.1 – Résolutions typiques des méthodes d'imagerie volumique basées sur la MEB [Briggman and Bock, 2012].

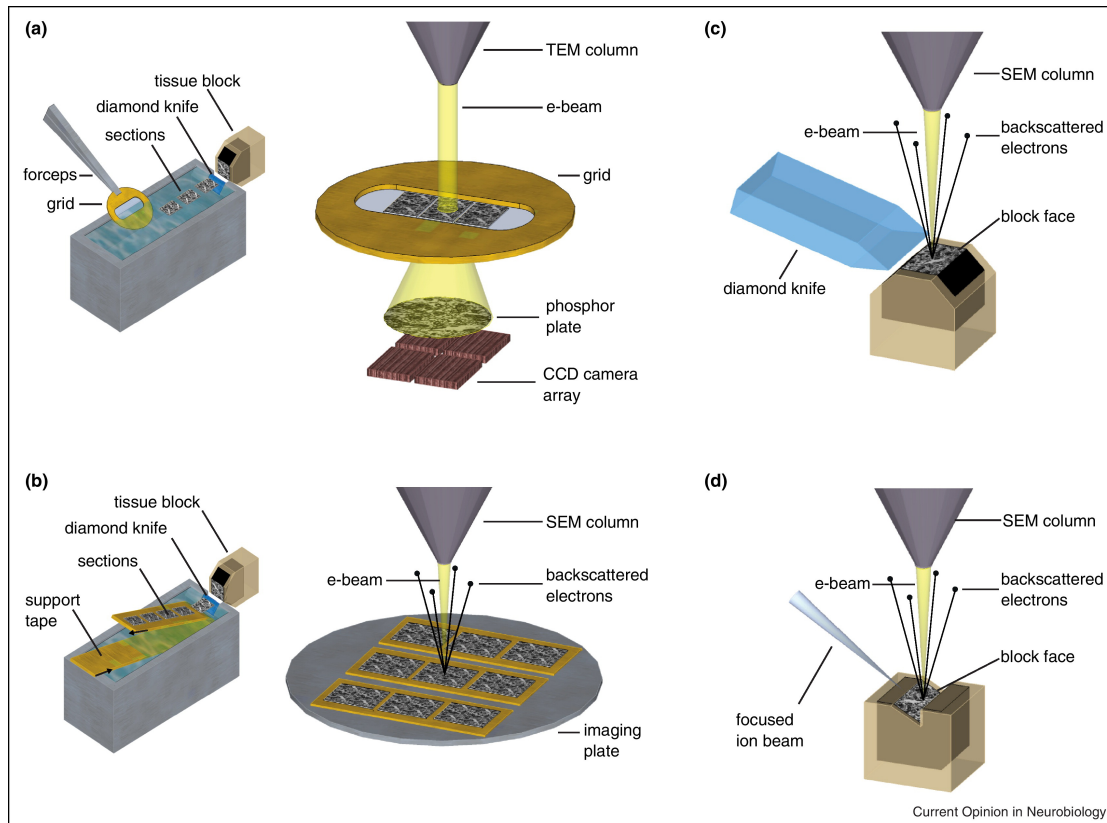


FIGURE 3.1 – Représentation schématique du fonctionnement des différentes méthodes d'acquisition citées : (a) ssTEM, (b) ATUM-SEM, (c) SBEM (d) FIB-SEM [Briggman and Bock, 2012].

Même lorsque la méthode d'acquisition est similaire, des **méthodes de fixation** différentes entraînent des différences fondamentales dans la représentation et la préservation des organelles.

Les méthodes de fixation utilisées sur les images que nous analysons sont la fixation chimique à température ambiante et la cryo-fixation par congélation haute pression et cryo-substitution.

Différentes **résines** peuvent être utilisées comme les lowicryl[®] (acrylate, plus communément acrylique) ou Epon[®] (polyépoxyde, plus communément epoxy). Ces résines permettent de fixer l'échantillon biologique afin de pouvoir le couper en fines tranches (coupes cellulaires) ou de décaper de fines tranches (FIB-SEM).

La **fixation chimique** à température ambiante utilise un réactif chimique (formaldéhyde, glutaraldéhyde) qui forme des liaisons covalentes entre les constituants cellulaires, et les agrègent les uns aux autres pour former une masse solide réticulée autour de laquelle on laisse diffuser une résine qui polymérise et devient dure. Exemple d'une image fixée chimiquement : figure 3.2a.

La **fixation par congélation à haute pression suivie d'une substitution à basse température** (cryo-fixation et cryo-substitution) consiste à immobiliser les structures cellulaires par le froid, puis de remplacer l'eau par un solvant (p. ex. acétone), une résine et un agent fixateur chimique (p. ex. tétroxyde d'osmium). L'objectif est d'immobiliser très rapidement les structures cellulaires afin d'éviter des réorganisations internes. Il faut toutefois éviter la formation de cristaux de glace lors de la congélation qui pourraient perturber l'organisation cellulaire. Pour cela, il faut que la vitesse de refroidissement soit très élevée ; l'échantillon est communément plongé dans un bain d'éthane liquide refroidi par de l'azote liquide à -196°C . Lorsque l'échantillon est trop épais (plus de $4\text{-}5\ \mu\text{m}$) la vitesse de refroidissement n'est pas suffisante et la baisse de température est accompagnée d'une augmentation de pression ($\pm 2000\ \text{bar}$) pour prévenir la formation de cristaux de glace. On peut alors atteindre des épaisseurs de vitrification de $100\ \text{à}\ 200\ \mu\text{m}$. Après cette étape, les étapes de fixation chimique, de coloration et de substitution sont faites à basse température (-20°C / -55°C) et sur un temps long, afin de préserver au mieux la structure de l'échantillon. Elles ont pour but d'inclure l'échantillon dans une résine dure, de ponter chimiquement les structures et de produire du contraste avec des agents de numéro atomique élevé diffusant beaucoup les électrons. Certains agents chimiques tels que le tetroxyde d'osmium sont non seulement des agents de contraste de numéro atomique élevé, mais également des agents fixateurs ayant la capacité de ponter chimiquement des protéines à l'intérieur des membranes biologiques. Dans certains cas que nous verrons dans la sous-section 3.1.1, la coloration tétroxyde d'osmium / acétate d'uranyl est utilisée et est mise en œuvre dès le début de l'opération de cryo-substitution avant même la polymérisation. Selon la nature et la concentration des agents de contraste et les protocoles de cryo-substitution utilisés, l'apparence et le contraste des images peuvent être radicalement différents. Par ailleurs, selon la distance de l'échantillon par rapport à la surface externe, la diffusion des agents de contraste peut limiter l'intensité du contraste et contribuer à une apparence hétérogène. Exemple d'une image cryo-fixée et cryo-substituée : figure 3.2b.

Une seconde notion est importante concernant les données : **la quantité d'annotations produites** par image. C'est la proportion de l'image qui est utilisable pour l'apprentissage d'un modèle et pour son évaluation. Comme la production d'annotation est coûteuse en temps, seule une partie des images est annotée.

3.1.1 Données internes

Une partie des données que nous utilisons est issue de la collaboration avec des laboratoires de recherche en biologie. Ces laboratoires produisent les images dans le but de mieux comprendre le fonctionnement de la cellule. Dans cette sous-section, nous allons lister les données auxquelles nous avons eu accès grâce à ces partenariats.

Ces données sont considérées comme internes car indisponibles au public pour l'instant. Une partie des données issues de la collaboration avec l'IGBMC sont en cours de publication. Il est à noter que les données auxquelles nous accédons ne sont pas les données brutes issues de l'appareil de mesure, mais des données ayant déjà subi plusieurs traitements. De plus grandes visualisations de ces images sont disponibles dans la section A.3.

IGBMC, équipe Schultz

Les premiers jeux de données sont issus de l'équipe Schultz de l'IGBMC. L'équipe étudie la biologie structurale et est spécialisée en imagerie de microscopie électronique cellulaire et moléculaire, en particulier avec les microscopes FIB-SEM. L'instrument utilisé dans l'équipe est un Zeiss Auriga 60, un microscope FIB-SEM. La nomenclature utilisée pour les images est la suivante : *échantillon-acquisition*, le numéro d'acquisition est omis lorsqu'il n'y en a qu'une.

L'échantillon *LW4* [Spiegelhalter et al., 2010, Alpy et al., 2013] correspond à une expérience dans laquelle les cellules HeLa (lignée cellulaire cancéreuse utilisée en biologie cellulaire et en recherche médicale) ont été infectées par un virus afin de produire la protéine StarD3 [Wilhelm et al., 2017] qui fait des ponts entre le réticulum endoplasmique et les endosomes. Le phénotype observé est un enrichissement du compartiment endosomal et une amplification importante en endosomes tardifs. *LW2* correspond à l'échantillon contrôle, une cellule HeLa infectée par un virus qui n'exprime aucune protéine. Les échantillons *LW4* et *LW2* sont cryo-fixés et cryo-substitués, l'enrobage utilisé est un lowicryl (HM20) et la coloration un mélange osmium / acétate d'uranyl.

Voici le détail complet de la préparation utilisée pour les échantillons *LW4* et *LW2*. Nous n'irons pas dans les détails pour les autres échantillons, mais cet exemple permet de mieux comprendre la difficulté de préparation de tels échantillons.

Les cellules HeLa sont cultivées sur un disque de saphir recouvert de carbone dans un mélange DMEM (Dulbecco's Modified Eagle Medium, un milieu de culture utilisé pour les cellules de mammifères) contenant 10% de sérum foetal de veau, avant d'être fixées à haute pression à la température de l'azote liquide. L'appareil de congélation à haute pression utilisé est un Leica EM PACT2. L'échantillon est déshydraté à -90°C dans de l'acétone contenant du tetroxide d'osmium de l'acétate d'uranyl et du glutaraldéhyde. La température est montée à -45°C pendant 18 heures avec une augmentation de $2,5^{\circ}\text{C}/\text{h}$. Les cellules sont ensuite déshydratées avec de l'acétone et imprégnées de résine (Lowicryl HM20) dans un bain pendant 2 heures. Quand la concentration en résine atteint 25%, la température est augmentée à -25°C et la concentration en résine à 100%. Enfin, la

résine est polymérisée à -25°C durant 48 heures à l'aide d'ultraviolet puis réchauffée à température ambiante en 9 heures.

L'échantillon *I3* correspond à une cellule HeLa non modifiée, fixée chimiquement à température ambiante. Ce type de fixation préserve moins la structure de la cellule ainsi que ses composantes.

Les images *I3* et *LW4-1* sont les premières sur lesquelles nous avons pu travailler dès 2019, et sont utilisées tout au long de la thèse. Ces images bien que représentant toutes deux une cellule HeLa sont de natures très différentes.

Pour la première image, *I3* (figure 3.2a) fixée chimiquement, les objets d'intérêt sont principalement reconnaissables grâce à leurs membranes et à leur texture. Pour la seconde, *LW4-1* (figure 3.3) fixée par congélation à haute pression, les objets d'intérêt sont reconnaissables à leur niveau d'intensité.

En 2022, Victor Hanss, doctorant dans l'équipe Schultz a produit de nouvelles images à partir de l'échantillon *LW4* et *LW2* : *LW4-2*, *LW4-3* (figure 3.3a), *LW4-4* (figure 3.3b), *LW2-1* (figure 3.4a), *LW2-4* (figure 3.4b) et *LW2-5*.

Les résolutions et taille de chacune des images sont disponibles dans la table 3.2.

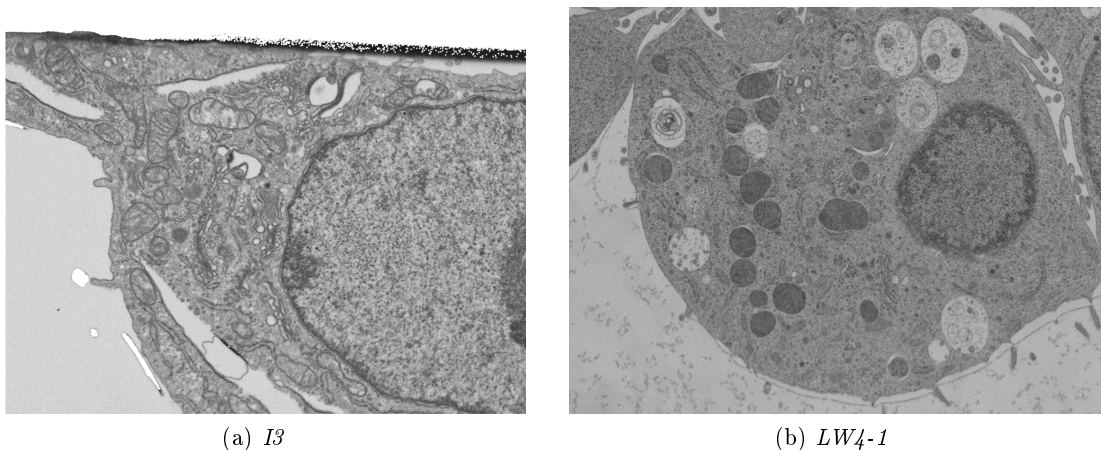


FIGURE 3.2 – Exemples d'images de coupes issues de l'image *I3* et *LW4-1*.

Toutes les annotations ont été produites par Véronique Mallouh, ingénieur de recherche au laboratoire IGBMC pour l'équipe Schultz, chargé du traitement et de l'annotations des données. Pour ces images, les annotations ont été faites par tranches, ce qui signifie que nous avons un nombre de tranches annotées. Pour l'image *I3*, les tranches sont consécutives, pour l'image *LW4-1* des blocs de 40 tranches sont annotés et espacés de 40 tranches non annotés.

Pour *I3*, les organelles suivantes sont annotées :

- mitochondries (tranches 1 à 500) ;
- réticulum endoplasmique (tranches 172 à 251) ;
- noyau (tranches 1 à 250) ;

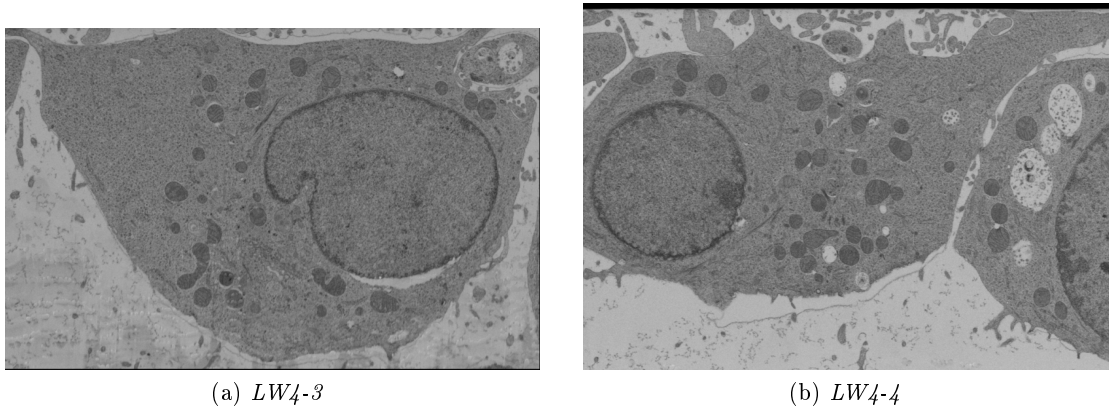


FIGURE 3.3 – Exemples d’images de coupes issues de l’échantillon *LW4*.

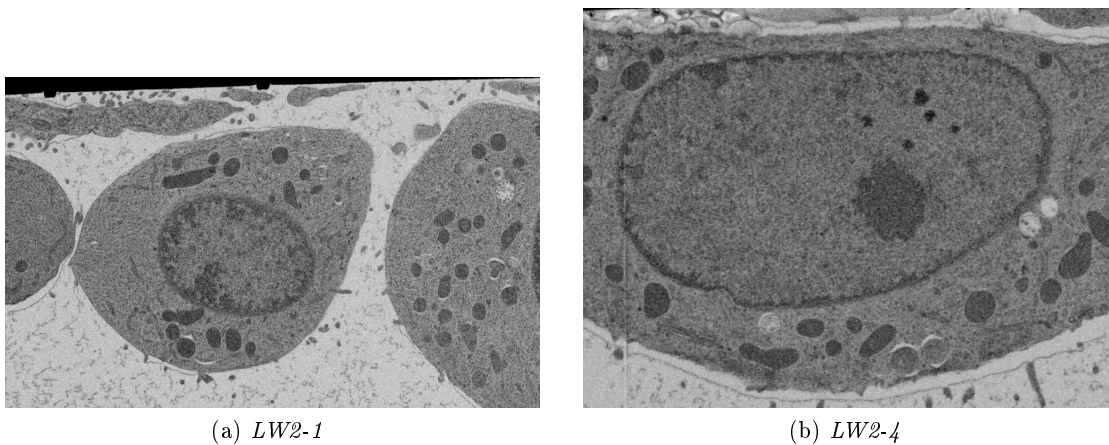


FIGURE 3.4 – Exemples d’images de coupes issues de l’échantillon *LW2*.

— membrane cellulaire (tranches 1 à 250).

Pour *LW4-1*, les organelles suivantes sont annotées :

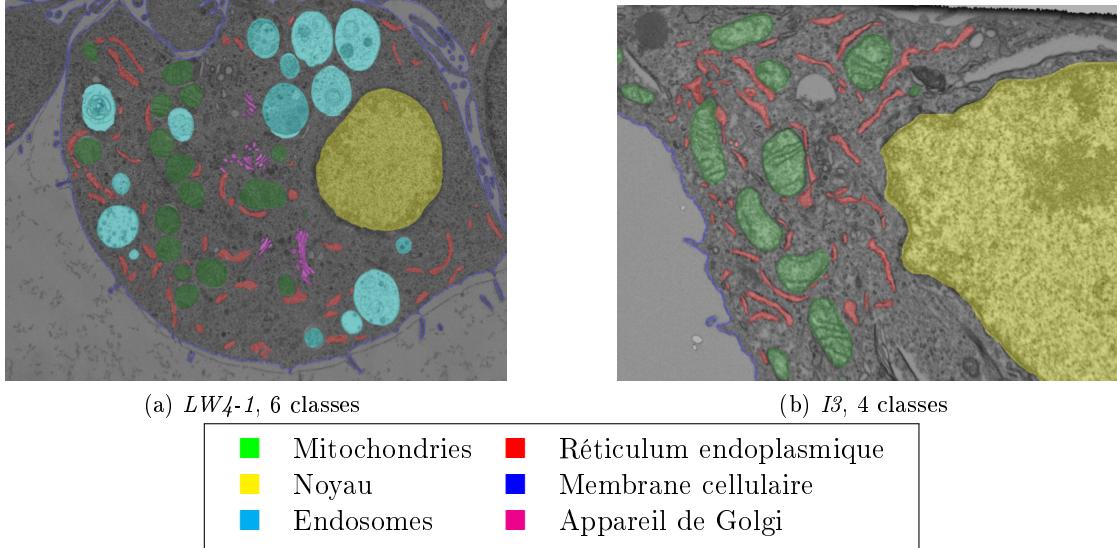
- mitochondries (8 blocs de 40 tranches) ;
- réticulum endoplasmique (2 blocs de 40 tranches) ;
- noyau (1 blocs de 40 tranches car il n’est plus présent ensuite) ;
- membrane cellulaire (8 blocs de 40 tranches) ;
- appareil de Golgi (1 blocs de 40 tranches) ;
- endosomes (8 blocs de 40 tranches).

La figure 3.5 montre un exemple d’annotation sur une tranche pour les images *I3* et *LW4-1*. La figure 3.6 montre une représentation 3D des annotations des mitochondries

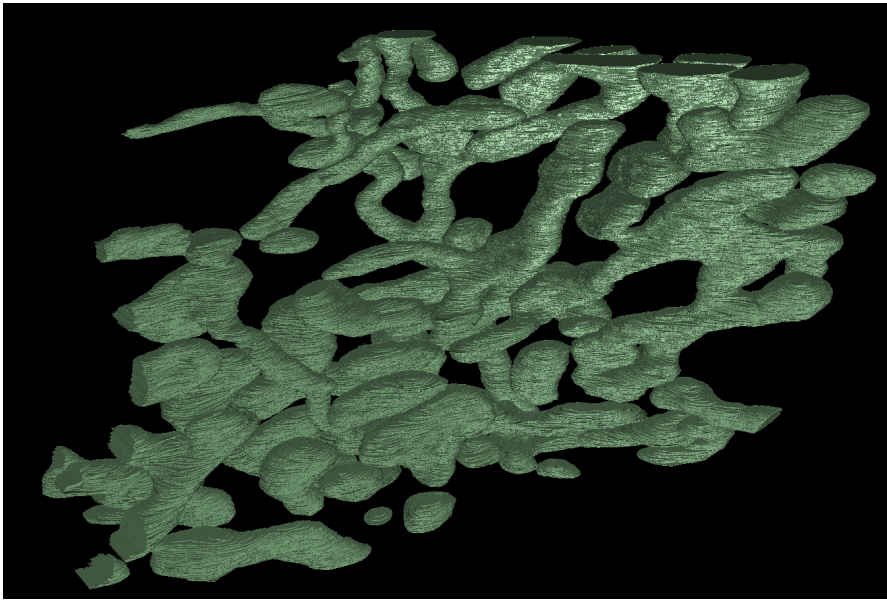
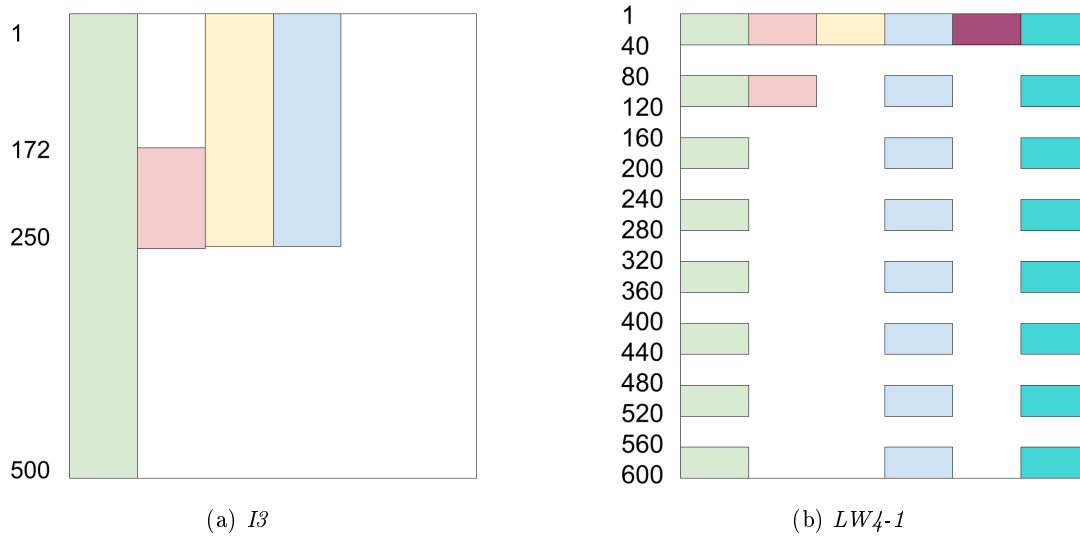
Nom	Échantillon	Résolution (nm)			Tailles (voxel)		
		Z	Y	X	Z	Y	X
I3	HeLa type naturel	20	5	5	500	1710	2580
LW4-1	HeLa STARD3	15	7,5	7,5	600	1394	1832
LW4-2	HeLa STARD3	8	8	8	422	1422	2048
LW4-3	HeLa STARD3	8	8	8	426	1236	1936
LW4-4	HeLa STARD3	8	8	8	500	1402	2046
LW2-1	HeLa témoin	8	8	8	1712	1624	2648
LW2-4	HeLa témoin	8	8	8	521	1120	3140
LW2-5	HeLa témoin	8	8	8	741	1398	3180

TABLE 3.2 – Résolutions et tailles des images IGBMC.

pour l'image *I3*. Cette représentation correspond au résultat attendu dans le cadre de nos travaux.

FIGURE 3.5 – Exemple d'annotations pour les images *I3* et *LW4-1*.

Pour les images *LW4-2*, *LW4-3* et *LW4-4*, des tranches non consécutives et représentatives de l'image ont été sélectionnées manuellement. Cinq tranches par image ont été annotées pour les mitochondries, réticulum endoplasmique, noyau, membrane cellulaire et endosomes. Des annotations faibles (annotation où seule une partie des pixels est annotée) ont également été produites pour chacune des images, 25 tranches pour *LW4-2* et 10 tranches pour *LW4-3* et *LW4-4*. Ces annotations faibles correspondent à des annotations plus rapides à produire ; les objets n'ont pas besoin d'être complètement annotés, mais une nouvelle classe apparaît, une classe de fond qui, dans le cas de segmentation dense, est implicite par dualité avec les annotations.

FIGURE 3.6 – Représentation 3D des mitochondries pour l'image I_3 .FIGURE 3.7 – Répartitions des tranches annotées pour I_3 et LW_4-1 . De gauche à droite, mitochondries (vert), RE (rouge), noyau (jaune), membrane cellulaire (bleu), appareil de Golgi (magenta) et endosomes (cyan).

La figure 3.8 montre un exemple d'annotation sur deux tranches de l'image LW_4-2 , une annotation dense (figure 3.8a) et une annotation faible (figure 3.8b).

La classe membrane cellulaire peut être représentée de plusieurs manières : soit en segmentant l'intérieur de la cellule, soit en segmentant uniquement la membrane avec une épaisseur de quelques voxels. Ces deux méthodes de représentation sont quasi équivalentes, car il est possible de passer d'une représentation à l'autre. Dans le cadre de l'entraînement d'un modèle de segmentation, cette différence de représentation est importante et doit être prise en compte pour obtenir de bons résultats. Dans le cas des annotations faibles, l'absence d'annotations pour chaque voxel ne permet plus la création d'une classe implicite représentant les éléments non annotés. Une mitochondrie peut tout à fait ne pas être annotée, mais cela ne signifie pas que ce n'en est pas une. Pour résoudre ce problème, la classe « extérieur » représentant le fond est utilisée.

La liste des données annotées est disponible dans la table 3.3 et leur répartition représentée dans la figure 3.7.

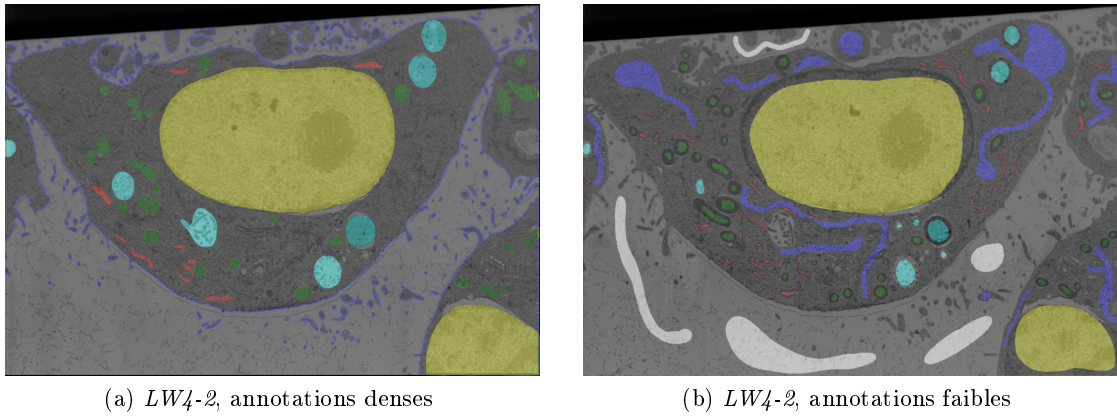


FIGURE 3.8 – Exemple d'annotations pour l'image *LW4-2*. Dans le cas des annotations faibles, la membrane cellulaire est représentée par deux classes, une classe « intérieur » (bleu) et une classe « extérieur » (blanc).

Nom	Annotations (nombre de tranches)					
	MITO	RE	CELL	NUC	ENDO	GOLGI
I3	500	79	250	250	0	0
LW4-1	320	80	320	40	320	40
LW4-2	5	5	5	5	5	0
LW4-3	5	5	5	5	5	0
LW4-4	5	5	5	5	5	0

TABLE 3.3 – Annotations disponibles pour les images IGBMC en nombre de tranches.

GoetzLab

Une seconde collaboration avec le GoetzLab nous a permis d'obtenir des images FIB-SEM de cellules cancéreuses in vivo. Le GoetzLab étudie la biomécanique des tumeurs et de la cascade métastatique, responsable de la prolifération du cancer dans le corps [Follain et al., 2018, Follain et al., 2020]. L'équipe étudie les mécanismes des métastases et se concentre sur les fonctions des organelles [Goetz et al., 2007, Hyenne et al., 2015, Ghoroghi et al., 2021].

Ces images sont de nature très différentes, car représentant des cellules cancéreuses in vivo obtenues dans des cerveaux de souris [Karreman et al., 2016]. Elles sont plus bruitées, mais également de taille beaucoup plus grande. Comme les noms donnés aux images lors de leurs acquisitions sont longs et peu pratique à utiliser, nous avons donné un identifiant aux images pour leur usage en interne. Les noms indiqués dans la table 3.4 sont des versions largement raccourcies des noms originaux. Nous avons principalement travaillé sur une des six images FIB-SEM fournies, *MB3-ROI1-2*, renommée *JG1* (figure 3.9), la seule pour laquelle des annotations ont été manuellement produites.

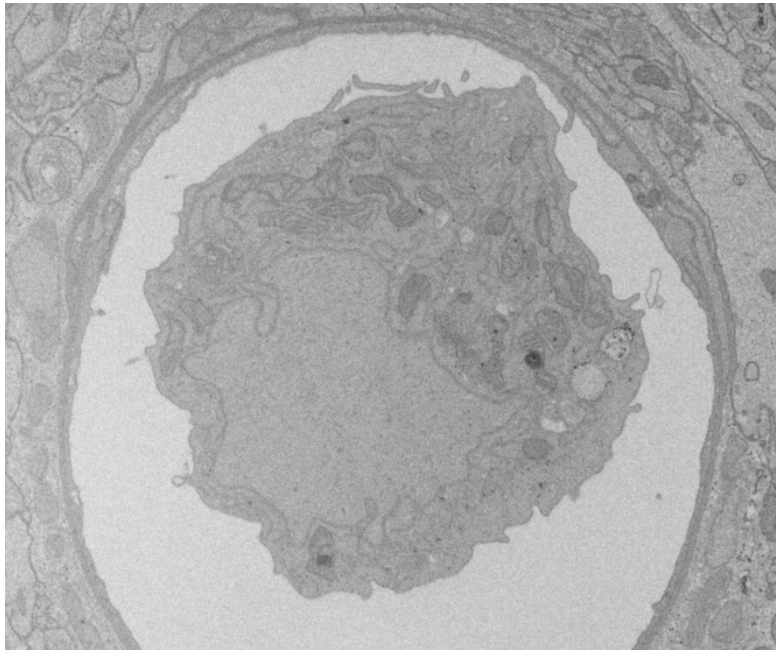


FIGURE 3.9 – Exemple d'une tranche de l'image JG1.

Parmi les sept images fournies, six ont été imagées à l'aide d'un FIB-SEM avec une résolution isotrope de 8nm et une (*JG7*) avec la technique ssTEM. En effet, le GoetzLab possède plus d'une trentaine d'images, mais seules les 6 images présentées ici ont été obtenues à l'aide d'un FIB-SEM.

Les images *JG1* et *JG6* sont issues d'une même acquisition, mais de deux cellules différentes. L'image *JG2* est issue de la même expérience, mais d'un second échantillon. Les images *JG3* et *JG4* représentent deux acquisitions effectuées sur un même échantillon.

Les images *JG1*, *JG2* et *JG6* représentent des cellules JIMT-1, une lignée cellulaire du cancer du sein. Les autres images correspondent à des images de cellules cancéreuses, mais nous ne sommes pas certains de la lignée cellulaire.

Tous les échantillons ont été fixés chimiquement et ont été inclus dans une résine Epon. Les résolutions et taille de chacune des images sont disponibles dans la table 3.4. La figure 3.10 représente un exemple de tranche pour chacune des images.

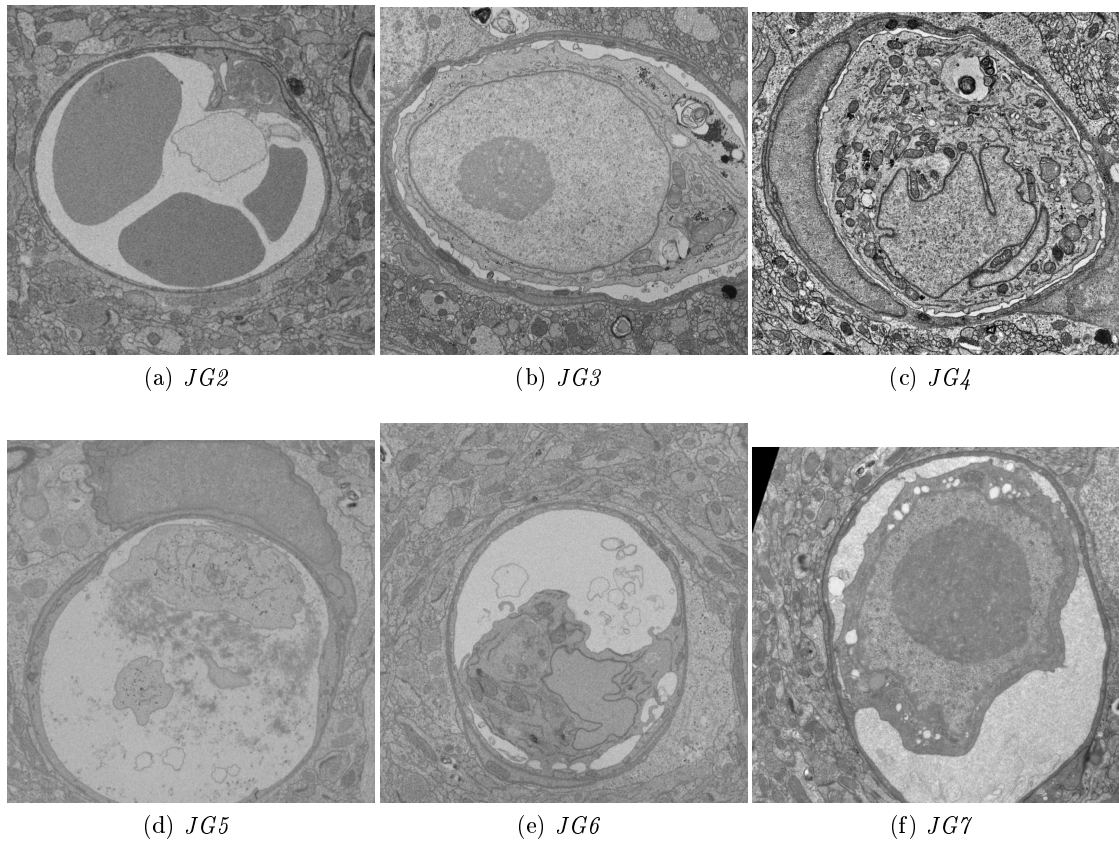


FIGURE 3.10 – Exemples de tranches des images GoetzLab.

Nom	ID	Échantillon	Résolution (nm)			Tailles (voxels)		
			X	Y	Z	X	Y	Z
MB3-ROI1-2	JG1	JIMT-1	8	8	8	2048	2824	3267
MB3-ROI2-A1-Inv-F	JG2	JIMT-1	8	8	8	2544	2512	2208
MB1-ROI3-1	JG3		8	8	8	3474	1797	1462
MB1-ROI3-2	JG4		8	8	8	3265	2752	2982
Aligned2000	JG5		8	8	8	1724	2604	2000
MB3-ROI1-2-TCe12	JG6	JIMT-1	8	8	8	2048	2824	1000
MB05-ROI2-1	JG7		5,2	5,2	65	3011	5048	50

TABLE 3.4 – Résolutions et tailles des images GoetzLab.

Des annotations sont uniquement disponibles pour l'image *JG1*. Elles ont été produites par Véronique Mallouh. Les annotations sont composées de cinq blocs de $384 \times 384 \times 384$ voxels. Pour ces images beaucoup plus bruitées que celles provenant de l'équipe Schultz, il n'est pas possible de reconnaître les organelles fines comme le réticulum endoplasmique. Les objets suivants ont été annotés :

- mitochondries ;
- noyau ;
- membrane cellulaire ;
- paroi vasculaire.

La figure 3.11 montre une représentation 3D d'un bloc et ses annotations.

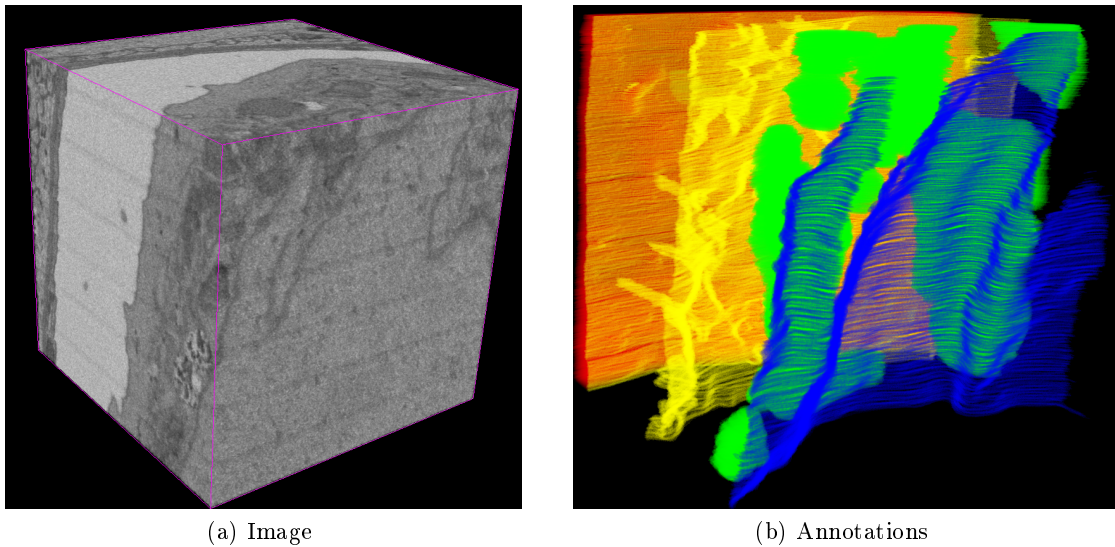


FIGURE 3.11 – Représentation 3D du cinquième bloc annoté de l’image *JG1*. Les couleurs ne correspondent pas aux couleurs utilisées précédemment pour des raisons de lisibilité. En bleu la membrane nucléaire, en vert les mitochondries, en jaune les membranes cellulaires et en rouge la paroi vasculaire.

3.1.2 Données publiques

Dans cette sous-section, nous allons présenter plusieurs jeux de données publics. Nous n’avons pas utilisé la plupart de ces jeux de données, soit car ils étaient trop différents de notre application, soit inexistantes lorsque les travaux en question furent menés. Cependant, leur utilisation pourrait être intéressante pour des applications de pré-entraînement, d’adaptation de domaine, ou pour se comparer plus facilement à d’autres méthodes.

Cet inventaire permet également de mettre en perspectives nos travaux, de comparer les quantités de données produites par d’autres laboratoires de recherche et de voir l’évolution dans la création de jeux de données.

Chacune des sous-sections présentées commencera par la liste des points communs entre le jeu de données et notre problématique.

La table 3.5 résume les informations concernant ces données. La colonne annotations permet de donner un ordre de grandeur du nombre de voxels (en milliards de voxels) qui sont annotés. Cette valeur a un but informatif, mais ne permet pas de comparer les jeux de données s’ils ne représentent pas les images aux mêmes résolutions ou ne concerne pas les mêmes types d’annotations.

OpenOrganelle

- FIB-SEM • quasi isotrope 4nm • organelles • multi-classes (35 classes)

OpenOrganelle est un projet du laboratoire de recherche Janelia appartenant à l’institut médical Howard Hughes situé à Chevy Chase aux États-Unis. L’objectif du projet est de proposer un portail de données d’images cellulaires et de segmentations issues de leurs travaux [Heinrich et al., 2020, Heinrich et al., 2021, Xu et al., 2020, Xu et al., 2021].

Les images OpenOrganelle sont des images de microscopie FIB-SEM représentant différentes cellules (HeLa, macrophages, lymphocytes T, cellule cancéreuse). Les données sont publiquement accessibles, mais dans les faits, c’est l’accès pour la visualisation de leur résultat qui est mis en avant (figure 3.12). La récupération des données d’entraînement est possible, mais nécessite le téléchargement de toutes les données (environ 2800 Go) et l’extraction des différents sous-blocs annotés.

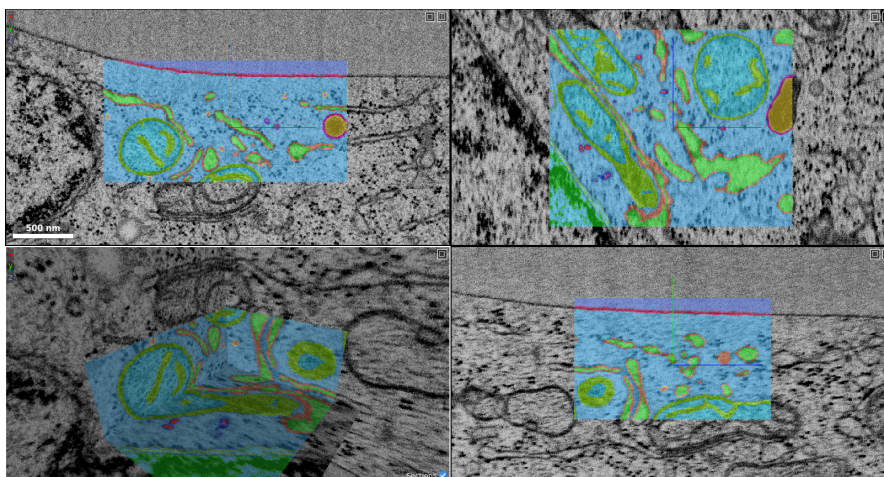


FIGURE 3.12 – Outil de visualisation en ligne des données OpenOrganelle, zoom sur un bloc annoté.

Les images sont de très grandes dimensions, quasi isotropes et représentent des cellules entières. L’acquisition de telles images est loin d’être trivial ; les images de cellules entières sont longues et difficiles à acquérir. Pour donner un exemple, l’image *jrc_hela-2* a été acquise en 25 jours.

Le jeu de données est composé de cinq images :

- *jrc_hela-2*, une cellule HeLa ;
- *jrc_hela-3*, une cellule HeLa ;
- *jrc_jurkat-1*, une lymphocyte T ;
- *jrc_macrophage-2*, un macrophage ;
- *jrc_sum159-1*, une cellule cancéreuse liée au cancer du sein.

Les images sont annotées par blocs (figure 3.13), en moyenne, des blocs de 230^3 . Le nombre de blocs dépend des images :

- *jrc_hela-2* 25 blocs ;
- *jrc_hela-3* 14 blocs ;

- *jrc_jurkat-1* 13 blocs ;
- *jrc_macrophage-2* 13 blocs ;
- *jrc_sum159-1* 12 blocs.

Le jeu de données propose au total 35 classes annotées. Dans les faits, les classes sont très spécifiques voire redondantes ; les mitochondries sont par exemple séparées en 3 classes.

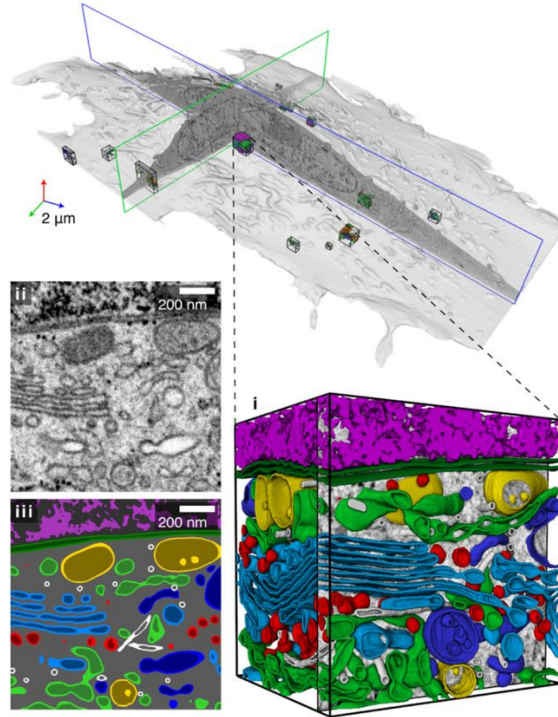


FIGURE 3.13 – Exemple d'un bloc d'entraînements du projet OpenOrganelle [Heinrich et al., 2020].

MitoEM

- organelle ● mitochondrie

Le jeu de données MitoEM a été publié en 2020 [Wei et al., 2020] et est ensuite devenu le sujet d'un challenge ISBI 2021. Les images MitoEM sont issues d'un microscope électronique à balayage multi-faisceaux (mSEM pour multi-beam scanning electron microscope) [Schalek et al., 2016, Shapson-Coe et al., 2021]. Les images représentent des échantillons de tissus humains ou des tissus de rat.

Les annotations correspondent à la segmentation d'instances des mitochondries. Le fait que les annotations soient des segmentations d'instances permet de les utiliser comme jeu de données de segmentation sémantique. La segmentation d'instance est une tâche plus complexe qui demande en plus de la segmentation sémantique, de séparer les instances des différentes mitochondries entre elles. L'intégralité des deux images est seg-

mentée (figure 3.14), ce qui en fait le plus grand jeu de données de segmentation de mitochondries à ce jour.

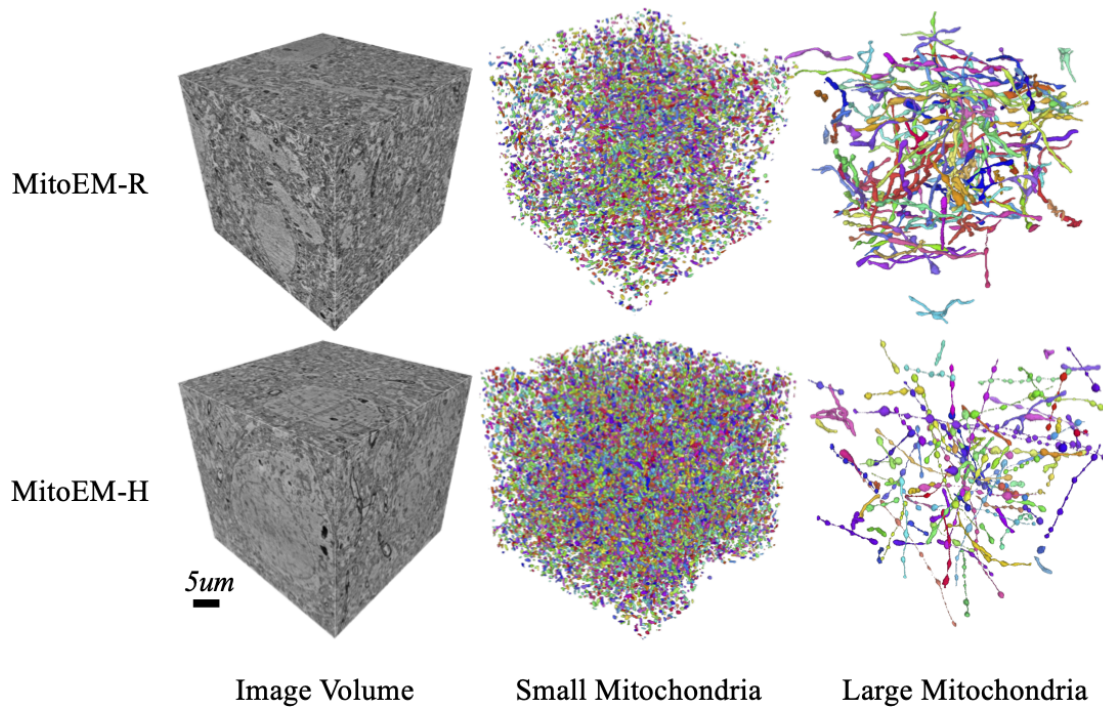


FIGURE 3.14 – Visualisation des jeux de données MitoEM-H et MitoEM-R [Wei et al., 2020].

ISBI 2012

- microscopie électronique cellulaire

Le jeu de données ISBI 2012 est issu du challenge du même nom. Les images ssTEM [Cardona et al., 2010, Arganda-Carreras et al., 2015] représentent le cordon nerveux ventral d'une larve de drosophile. Les données d'entraînement sont composées de 30 tranches dont les structures neuronales ont été segmentées (figure 3.15). Les annotations sont données de manière in-out, c'est-à-dire blanc pour les voxels des objets segmentés et noir pour le reste des voxels (qui correspondent principalement aux membranes).

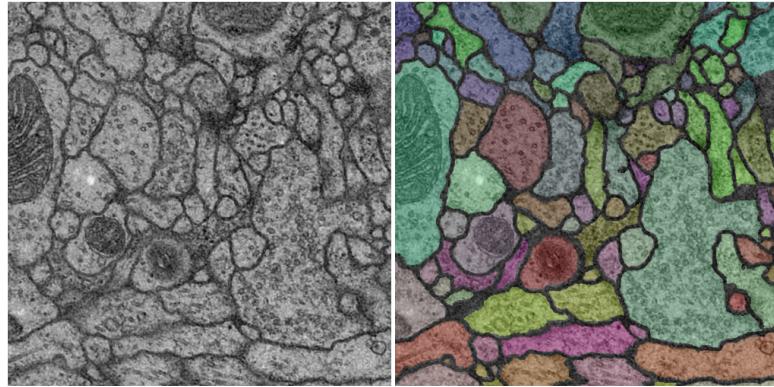


FIGURE 3.15 – Exemple d’image TEM ISBI 2012 et sa segmentation correspondante.

EPFL Lucchi

- FIB-SEM • isotrope 5nm • organelle • mitochondrie

Le jeu de données EPFL Lucchi, issu de l’article CVPR13 [Lucchi et al., 2012] est composé d’une image FIB-SEM représentant une partie de l’hippocampe CA1 d’un cerveau de souris. Les données annotées sont composées de deux sous-blocs de taille $165 \times 768 \times 1024$ dont les mitochondries sont segmentées (figure 3.16).

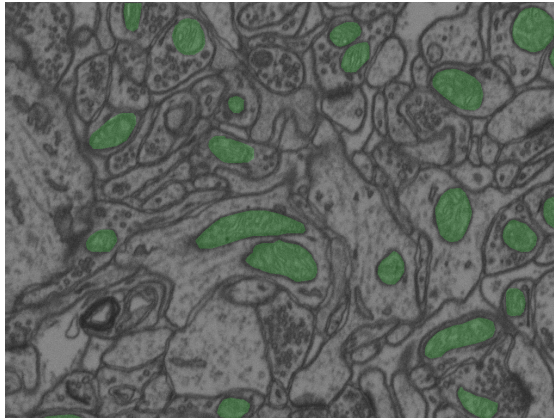


FIGURE 3.16 – Exemple d’une tranche d’entraînement du jeu de données EPFL Lucchi.

Kasthuri 2015 (Kasthuri++)

- microscopie électronique cellulaire • organelle • mitochondrie

Le jeu de données Kasthuri 2015 est issu des travaux menés par l’équipe ayant rédigé l’article [Kasthuri et al., 2015]. Les images représentent des tissus cérébraux de souris imagés avec la technique ATUM-SEM. Les annotations sont proposées pour les mitochondries et d’autres objets (intra-)cellulaires comme les synapses, vésicules, fibre nerveuse,

dendrites et cellules gliales. En 2020, une version améliorée et corrigée des annotations pour les mitochondries a été proposée dans l'article [Casser et al., 2020].

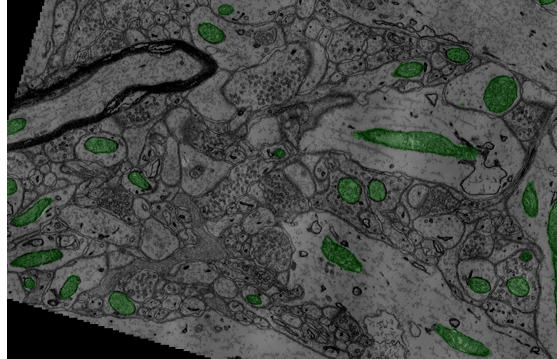


FIGURE 3.17 – Exemple d'une tranche d'entraînement du jeu de données Kasthuri 2015 (uniquement la classe de mitochondries).

CREMI

- microscopie électronique cellulaire

Le challenge CREMI (Circuit Reconstruction from Electron Microscopy Images) est issu du laboratoire de recherche Janelia, tout comme OpenOrganelle. Le jeu de données est composé de trois images ssTEM de cerveau de mouche. Tous les volumes sont segmentés, les classes utilisées sont pour les neurones, synapses et relations synaptiques (figure 3.18).



FIGURE 3.18 – Exemple d'une tranche du jeu de données CREMI.

3D Platelet EM

- microscopie électronique cellulaire • organelles • mitochondrie • membranes cellulaires

Le jeu de données 3D Platelet EM est issu de l'article [Guay et al., 2021b] et est composé de deux images 3D SBEM de thrombocytes (plaquettes sanguines) et de segmentations sémantiques pour les deux volumes. Les segmentations proposées concernent les cellules, mitochondries et différents objets spécifiques aux thrombocytes comme les granules plaquettaires (figure 3.19).

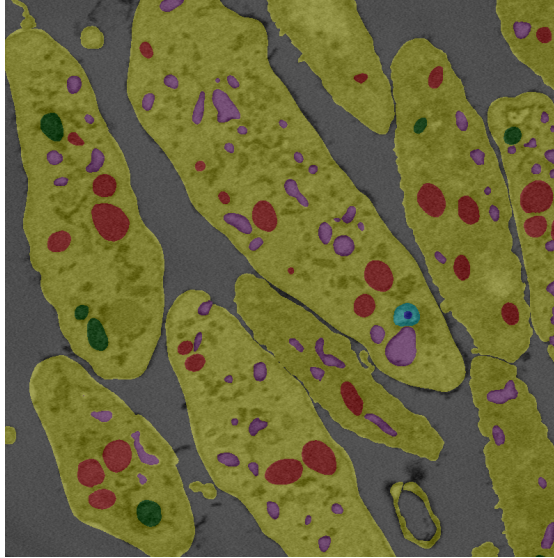


FIGURE 3.19 – Exemple d’une tranche d’entraînement du jeu de données 3D Platelet EM. En jaune la cellule, en vert les mitochondries, en rouge les canalicules, en bleu, cyan et violet, différents niveaux de granules plaquettaires.

CEM500k, CEM1.5M et CEM-MitoLab

- microscopie électronique cellulaire • organelle • mitochondrie

Le jeu de données CEM500k est très différent de tous les autres. Ce jeu de données est composé d’environ 500000 images cellulaires 2D, mais sans aucune annotation. Ces images sont issues d’environ 600 volumes 3D et 10000 images 2D provenant de plus de 100 projets d’imagerie différents. Plus récemment, le jeu de données a été agrandi pour contenir 1,5 millions d’images (CEM1.5M). Ce jeu de données est en particulier intéressant pour le pré-entraînement de modèles 2D [Conrad and Narayan, 2021, Conrad and Narayan, 2023].

Le jeu de données CEM-MitoLab est composé d’images provenant de CEM1.5M pour lesquelles des annotations ont été créées pour les mitochondries [Conrad and Narayan, 2023]. Au total, ce sont 21860 images 2D qui sont annotées, représentant 135285 instances de mitochondries.

Nom	Échantillon	Image	Résolution (nm)			Dimension (voxel)			Annotations (Giga Voxels)
			Z	Y	X	Z	Y	X	
OpenOrganelle 2021									
jrc_hela-2	HeLa (ATCC CCL-2)	FIB-SEM	5,2	4	4	6368	1600	12000	0,291
jrc_hela-3	HeLa (ATCC CCL-2)	FIB-SEM	3,2	4	4	12000	1000	12400	0,175
jrc_jurkat-1	Jurkats E6-1 (ATCC TIB-152)	FIB-SEM	3,4	4	4	8560	3000	10000	0,176
jrc_macrophage-2	THP-1 (ATCC TIB-202)	FIB-SEM	3,4	4	4	11087	2000	10000	0,163
jrc_sum159-1	SUM-159	FIB-SEM	4,6	4	4	7632	2800	16000	0,080
MitoEM 2020									
MitoEM-H	Tissus humain	mSEM	30	8	8	1000	4096	4096	8,389
MiroEM-R	Tissus de rat	mSEM	30	8	8	1000	4096	4096	8,389
Autres									
ISBI 2012	VNC Drosophile	ssTEM	50	4	4	60	512	512	0,008
EPFL lucchi 2012	CA1 hippocampus	FIB-SEM	5	5	5	1065	1536	2048	0,260
Kasthuri 2015	Tissu cérébral	ATUM-SEM	30	3	3	85	1613	1463	0,199
			30	3	3	75	1553	1334	
CREMI 2017	Cerveau de mouche	ssTEM	40	4	4	200	3072	3072	0,586
			40	4	4	200	3072	3072	
			40	4	4	200	3072	3072	
3D Platelet EM 2021	Thrombocyte	SBEM	50	10	10	50	800	800	0,077
			50	10	10	24	800	800	
			50	10	10	121	609	400	
CEM-MitoLab 2022						224	224	224	1,097

TABLE 3.5 – Informations pour les données accessibles publiquement. Les cellules en gras correspondent aux données similaires aux nôtres (méthodes d'imagerie et résolution et type d'annotations).

3.2 Évaluation

Dans cette seconde section, nous allons voir les différentes métriques permettant d'évaluer nos segmentations, et discuter de leurs utilisations.

Il n'y a pas à l'heure actuelle de consensus sur la métrique à utiliser pour comparer les résultats d'une segmentation. Les auteurs choisissent généralement les métriques en fonction de leur domaine d'application et des méthodes utilisées dans les références antérieures, afin de permettre une comparaison sur des jeux de données similaires.

Dans le domaine de la segmentation sémantique d'images de la vie courante, c'est plus généralement l'intersection sur union (IoU pour Intersection-over-Union) qui est utilisée. Dans le domaine biomédical, on utilise plus régulièrement le F1-score.

Par exemple, les auteurs de [Çiçek et al., 2016, Chen et al., 2018a] donnent leurs résultats en intersection sur union, les auteurs de [Haberl et al., 2018, Heinrich et al., 2021] en F1-score et les auteurs de [Xiao et al., 2018, Liu et al., 2020] proposent les deux métriques.

Pour illustrer notre propos, et évaluer l'utilisation de chacune des métriques dans les différents domaines, nous avons analysé les résultats de benchmark disponible sur le site `paperswithcode.com`, en comptant les occurrences de chacune des métriques (seule la métrique principale est comptée). Pour le domaine de segmentation sémantique biomédicale, nous avons regroupé les tâches suivantes : segmentation d'images médicales, segmentation de lésions, segmentation de tumeurs cérébrales, segmentation cellulaire, segmentation de vaisseaux rétiniens et segmentation d'images médicales 3D. Au total, nous avons sélectionné 175 benchmarks, 100 tâches de segmentation sémantique classique et 75 tâches biomédicales. Pour les 100 tâches de segmentation, l'intersection sur union est utilisée 80 fois, le F1-score 8 fois et d'autres métriques 12 fois. Pour les 75 tâches de segmentation biomédicale, l'intersection sur union est utilisée 13 fois, le F1-score 53 fois et d'autres métriques 9 fois. La figure 3.20 représente graphiquement ces proportions.

Il existe d'autres métriques, moins courantes qui peuvent également être informatives sur la qualité d'une segmentation. On peut regrouper les différentes métriques en trois sous-groupes. Les mesures de recouvrement, les mesures de distances aux contours et les comparaisons de composantes connexes.

3.2.1 Mesure de recouvrement

Une mesure de recouvrement permet de comparer les proportions de pixels de l'image qui sont correctement classifiés par rapport aux pixels erronés. Les mesures de recouvrement utilisent les notions de vrais positifs (TP pour true positive), vrais négatifs (TN pour true negative), faux positifs (FP pour false positive) et faux négatifs (FN pour false negative) qui sont représentées schématiquement dans la figure 3.21. On note respectivement P la prédiction et G la segmentation de référence.

À partir de ces notions, nous pouvons construire des métriques de recouvrement. Nous l'avons vu précédemment, les métriques les plus communes en segmentation sont l'intersection sur union et le F1-score.

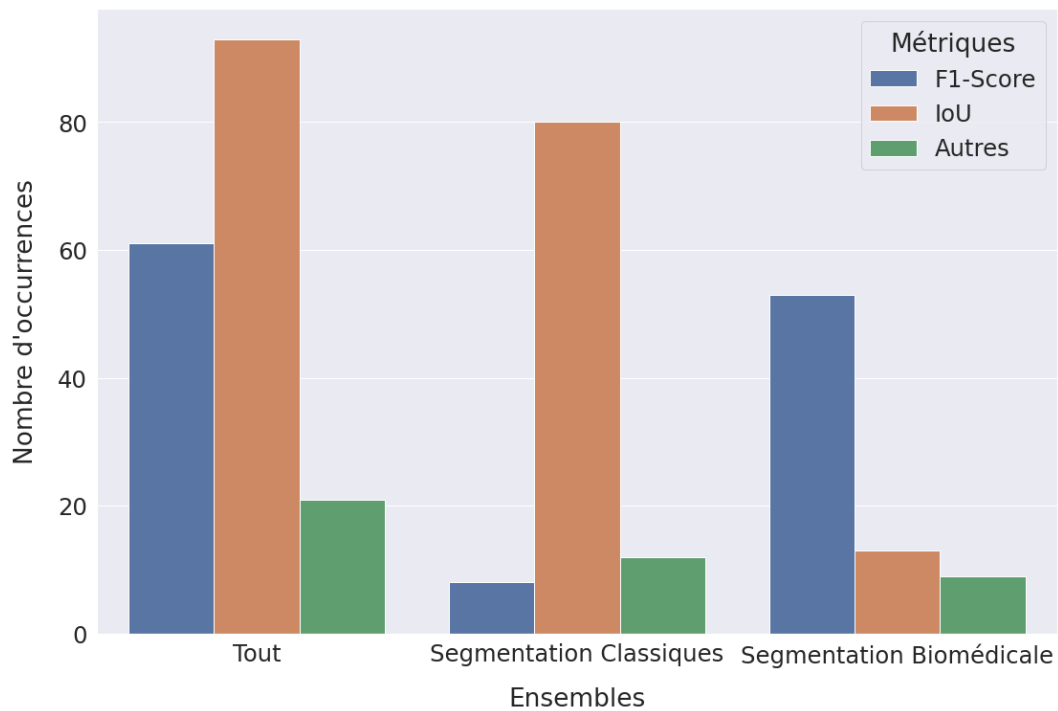


FIGURE 3.20 – Représentation de l'utilisation des métriques intersection sur union et F1-score dans le domaine de la segmentation.

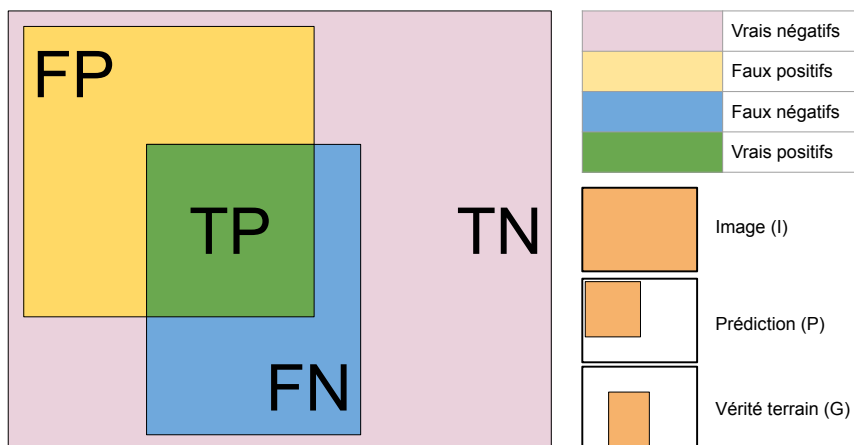


FIGURE 3.21 – Résumé schématique des notions d'erreurs possibles par pixel des mesures de recouvrement.

Dans le cas où les masques de segmentation sont discrets et binaires, le F1-score est équivalent à l'indice de Sørensen-Dice. Dans le domaine de la segmentation, le terme

F1-score est généralement utilisé lorsqu'une métrique est concernée alors que le terme Dice est souvent préféré lorsque l'on parle de fonction de coût permettant d'entraîner un réseau de neurones, mais ce n'est pas une règle, les deux termes étant parfois mélangés. De même, l'intersection sur union est parfois appelé indice de Jaccard. Nous définissons les équivalences entre notations ensemblistes et binaires dans les équations 3.1 à 3.4.

$$|P| = TP + FP \quad (3.1)$$

$$|G| = TP + FN \quad (3.2)$$

$$|P \cap G| = TP \quad (3.3)$$

$$|P \cup G| = TP + FP + FN \quad (3.4)$$

Les équations 3.5 à 3.11 décrivent les métriques les plus courantes ainsi que les équivalences IoU / Jaccard (noté *JSC*) et F1 / Dice (noté *DSC*). Il est important de ne pas confondre l'exactitude (accuracy en anglais) et la précision.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.5)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.7)$$

$$\begin{aligned} F1 &= \frac{2 \times Precision \times Recall}{Precision + Recall} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN} \end{aligned} \quad (3.8)$$

$$\begin{aligned} DSC &= \frac{2|P \cap G|}{|P| + |G|} \\ &= \frac{2 \times TP}{TP + FP + TP + FN} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN} \\ &= F1 \end{aligned} \quad (3.9)$$

$$JSC = \frac{TP}{TP + FP + FN} \quad (3.10)$$

$$\begin{aligned}
\text{IoU} &= \frac{|P \cap G|}{|P \cup G|} \\
&= \frac{TP}{TP + FP + FN} \\
&= \text{JSC}
\end{aligned} \tag{3.11}$$

Pour nos mesures de recouvrement, nous avons décidé d'utiliser le F1-score, la métrique la plus commune dans notre domaine d'étude.

3.2.2 Mesures de distance aux contours

Les mesures de distance permettent d'évaluer le bon placement des contours des objets segmentés. Comparées aux mesures de recouvrement, les mesures de distances aux contours sont plus sensibles aux modifications topologiques des composantes connexes (fusion, trou) tout en étant moins sensibles aux légers déplacements de contours.

La distance de Hausdorff permet de mesurer une distance entre deux ensembles. Si on définit une distance orientée qui correspond à la distance maximum entre les points d'un premier ensemble vers le point le plus proche dans le second ensemble, la distance de Hausdorff correspond à la distance maximale des distances orientées entre les deux ensembles. C'est la valeur maximale des distances minimales.

On considère nos deux masques de segmentation G et P comme deux ensembles non vides d'un espace métrique et $d(x, A)$ la distance d'un élément x vers un ensemble A :

$$d(x, A) = \min_{y \in A} \|x - y\|_2 \tag{3.12}$$

La distance de Hausdorff d_H est définie comme :

$$d_H(P, G) = \max \left\{ \sup_{x \in P} d(x, G), \sup_{y \in G} d(y, P) \right\} \tag{3.13}$$

Si on adapte la formulation aux contours des segmentations notés ∂G pour les contours de G et ∂P pour les contours de P , la distance de Hausdorff correspond à la distance de surface symétrique maximale [Yeghiazaryan and Voiculescu, 2018] :

$$\text{HD} = \max \left\{ \max_{x \in \partial P} d(x, \partial G), \max_{y \in \partial G} d(y, \partial P) \right\} \tag{3.14}$$

La distance de surface moyenne est définie de la même manière, mais en utilisant la distance moyenne plutôt que la distance maximale. La distance de surface symétrique moyenne (ASSD : average symmetric surface distance) est une mesure permettant de caractériser la différence moyenne de distance entre les contours des objets segmentés. C'est la valeur moyenne des distances minimales.

$$\text{ASSD} = \frac{\sum_{x \in \partial P} d(x, \partial G) + \sum_{y \in \partial G} d(y, \partial P)}{|\partial P| + |\partial G|} \quad (3.15)$$

On peut trouver différentes formulations ou noms pour l'ASSD comme la distance de Hausdorff moyenne [Aydin et al., 2021] ou dans le cas plus général d'ensembles de points, la distance de chanfrein [Nguyen et al., 2021].

La distance de Hausdorff donne une information sur le pire élément, alors que l'ASSD sur la moyenne de tous les éléments. Pour nos mesures de distance aux contours, nous utiliserons l'ASSD.

3.2.3 Mesure de composantes connexes

Les mesures de composantes connexes comptent les composantes connexes correctement segmentées. Pour effectuer une mesure de composantes connexes, on traite chaque composante individuellement. On définit ensuite les résultats positifs et négatifs en fonction d'une mesure de recouvrement ou d'une mesure de distance. Dans le cas de la segmentation sémantique où les objets ne sont pas instanciés, on peut uniquement utiliser des mesures de recouvrement qui ne dépendent pas des faux positifs et vrais négatifs. On peut par exemple utiliser le rappel (équation (3.7)). Ce problème n'est pas présent en segmentation d'instances. Contrairement aux mesures de recouvrement et de distance aux contours, les mesures de composantes connexes sont sensibles aux objets quelles que soient leurs dimensions.

Pour le challenge MitoEM (ISBI 2021) de segmentation d'instances [Wei et al., 2020], la métrique d'évaluation correspond à l'exactitude sur les composantes : une composante est valide si son intersection sur union dépasse 75%.

Dans notre article de 2021 portant sur la segmentation automatique des organelles [Meyer et al., 2021], nous proposons un comptage des mitochondries et réticulum endoplasmique correctement détectés, sous-détectés et non détectés. Les statuts de détections étaient définis en fonction de valeurs seuils du rappel ($> 0,8$ détectés et $< 0,10$ non détectés).

3.2.4 Comparaison des mesures

Dans cette dernière sous-section, nous allons présenter plusieurs exemples afin de montrer comment les différentes métriques que nous avons choisies se comportent en fonction de différentes erreurs de segmentation potentielles. Nous verrons que même si les mesures sont basées sur des concepts différents, en pratique, les tendances sont similaires.

Pour commencer, le premier exemple de travail met en avant les limitations de chacune des métriques. Nous avons créé des segmentations de référence et segmentations synthétiques. Les segmentations proposées sont de nature différente, mais pourtant, ont des scores similaires. Dans le premier cas (figure 3.22), les F1-score des deux segmentations sont égaux. Dans le second cas (figure 3.23), ce sont les ASSD qui sont quasiment égaux.

Même si dans la pratique, il est peu probable que ce genre de cas survienne, l'utilisation des deux métriques permet d'être certain que la segmentation ne présentera pas d'aberrations.

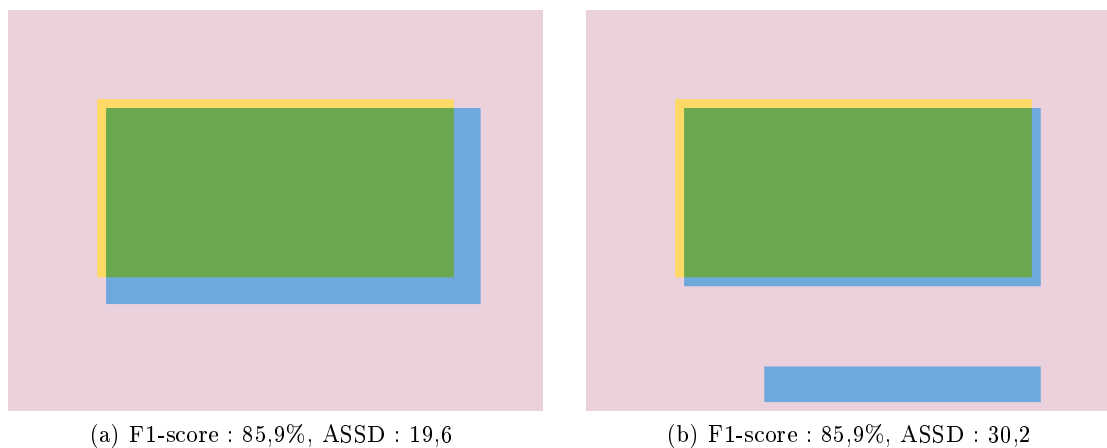


FIGURE 3.22 – Exemple de travail : F1-score similaire.

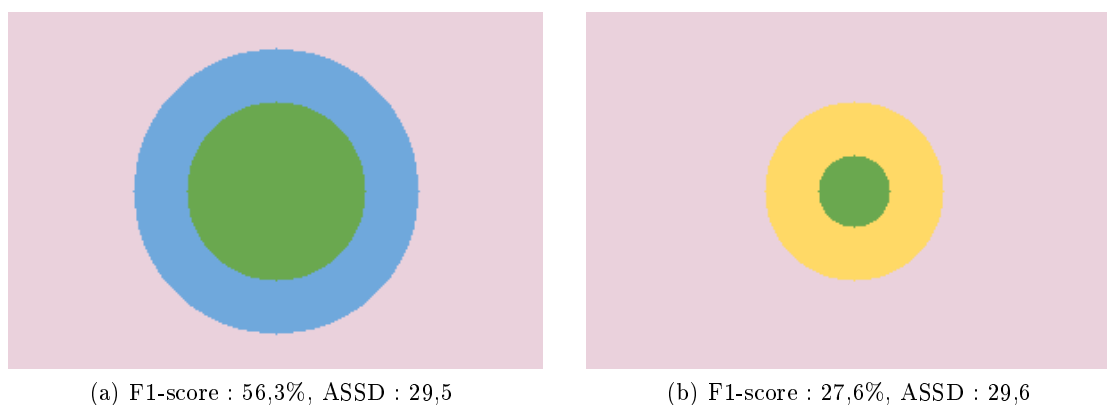


FIGURE 3.23 – Exemple travail : ASSD similaire.

Pour ce second exemple, nous avons extrait une coupe de la segmentation de référence de l'image *LW4-1* pour les classes des mitochondries et du réticulum endoplasmique. Nous leur avons appliqué des transformations visant à simuler des erreurs de segmentation, érosion ou dilatation avec un élément structurant en forme de disque et de rayon croissant, décalage de plusieurs pixels dans une direction et transformation élastique [Simard et al., 2003] de plus en plus forte.

Les résultats sous forme de graphiques sont présents dans les figure 3.24 à 3.27. On représente l'opposée de la valeur de l'ASSD afin de mieux visualiser la tendance avec le F1-score.

On remarque que si les tendances sont similaires, l'ASSD pénalise systématiquement plus fortement la classe des mitochondries (plus gros objets). On peut également re-

marquer que dans le cas des déformations élastiques, l'ASSD est fortement impacté dès qu'une faible déformation est appliquée.

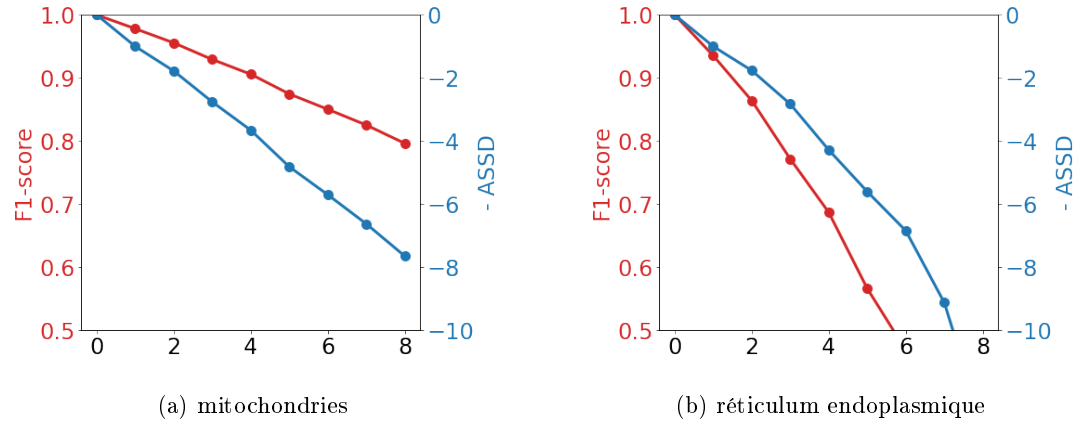


FIGURE 3.24 – Transformations de la segmentation de référence par érosion avec un élément structurant de plus en plus grand.

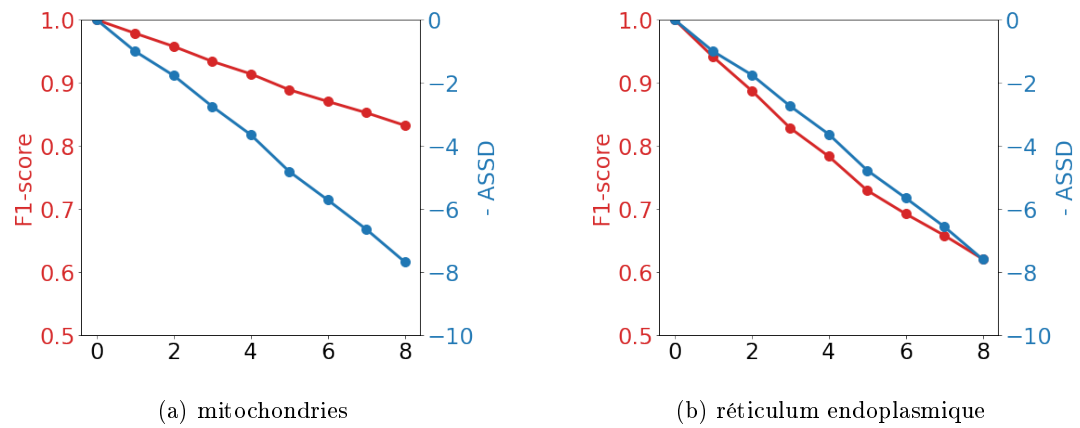


FIGURE 3.25 – Transformations de la segmentation de référence par dilatation avec un élément structurant de plus en plus grand.

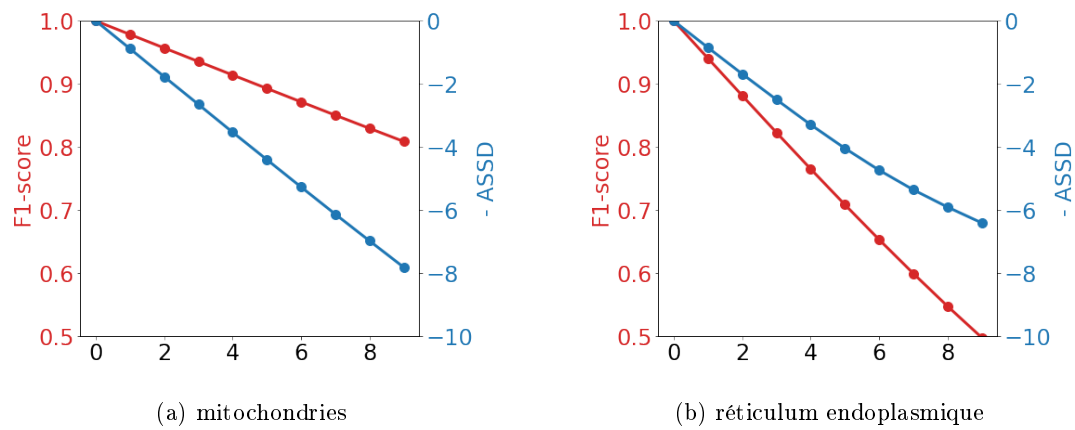


FIGURE 3.26 – Transformations de la segmentation de référence par décalage avec un décalage de plus en plus grand.

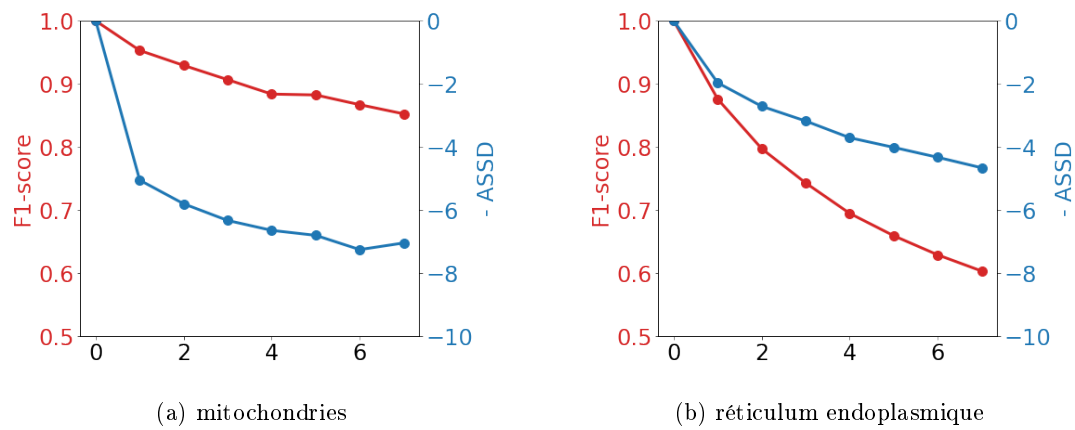


FIGURE 3.27 – Transformations de la segmentation de référence par transformation élastique de plus en plus forte.

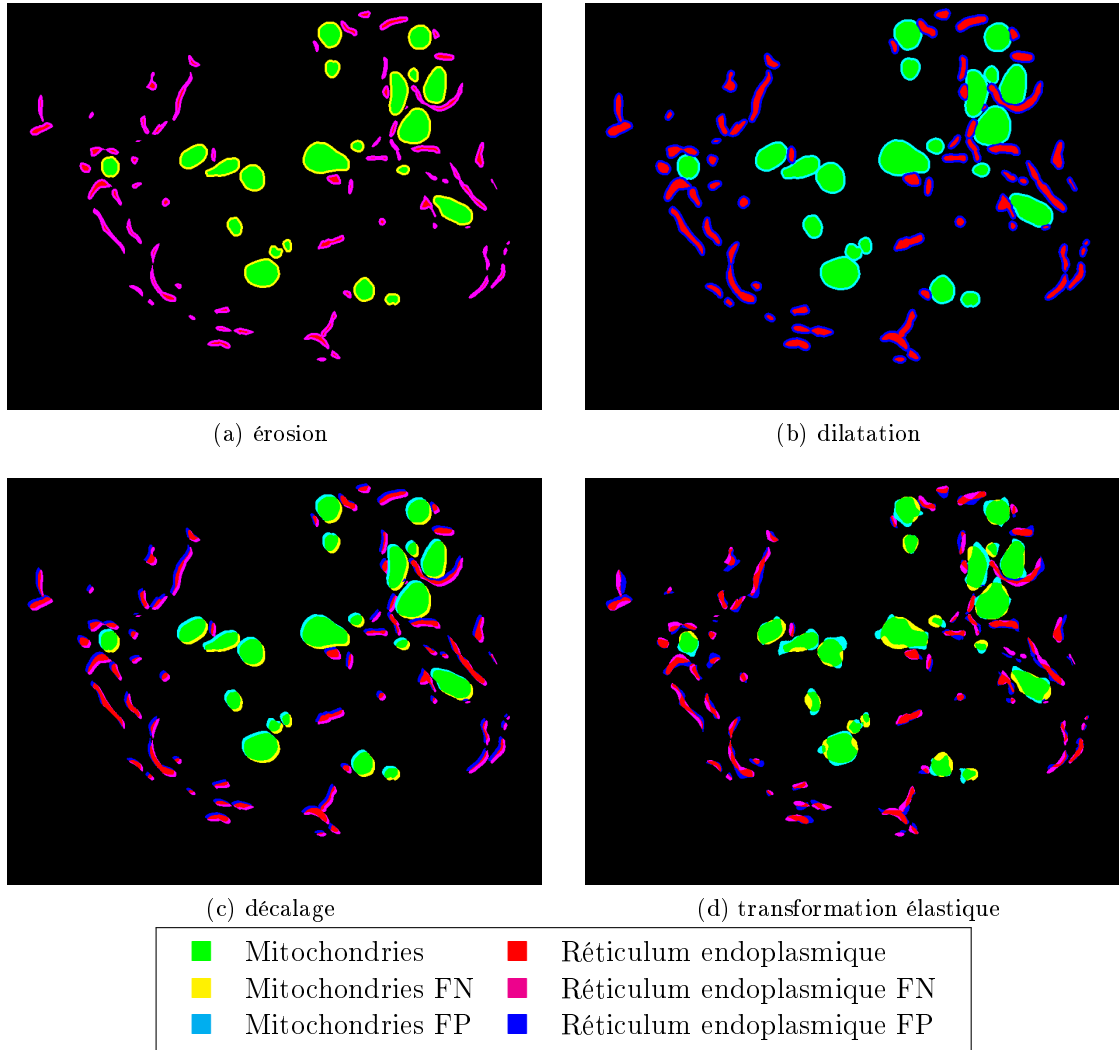


FIGURE 3.28 – Exemple des transformations appliquées.

Dans la plupart des cas, la variation est similaire entre les deux métriques. On remarque pour le cas de l'érosion, que le réticulum endoplasmique est fortement impacté. C'est une classe composée de petits objets qui disparaissent très rapidement si l'on érode trop la segmentation. La transformation élastique est la seule transformation n'entraînant pas une dégradation quasi linéaire des résultats.

En pratique, l'analyse de plusieurs métriques est nécessaire lorsque les résultats sont difficiles à interpréter, que les scores sont très proches ou bruités. Par exemple, dans certains cas que nous verrons dans le chapitre 4, deux méthodes sembleront très proches en F1-Score alors que l'ASSD permet de mettre en évidence une différence significative de qualité.

Dans ce chapitre, nous avons présenté les différents jeux de données que nous utiliserons, ainsi que ceux proches de notre problématique. La multiplication récente du nombre de jeux de données rend compte d'un intérêt grandissant pour les méthodes utilisant l'apprentissage, mais également d'un besoin grandissant du côté des biologistes.

Nous avons également détaillé les métriques que nous utiliserons pour évaluer nos segmentations, et montré comment ces mesures se comportent en fonction des différentes erreurs de segmentation.

Apprentissage profond et segmentation d'images de microscopie électronique cellulaire

4.1	Protocole expérimental	70
4.2	Segmentation automatique avec annotations denses	73
4.2.1	Modèles multi-classes et binaires	73
4.2.2	Modèles 2D et 3D	76
4.2.3	Variations du modèle U-Net	80
4.2.4	Taille de patch 2D pour l'entraînement	84
4.2.5	Taille du jeu d'entraînement	86
4.2.6	Fonction de perte	87
4.2.7	Modèle pré-entraîné	90
4.2.8	Adaptation de domaine et CycleGAN	90
4.2.9	Premières conclusions	94
4.3	Segmentation interactive et annotations faibles	95
4.3.1	Segmentation interactive et simulation d'interaction	95
4.3.2	Interactions via interface homme machine	104
4.3.3	Annotations faibles	105
4.3.4	Résultats de segmentation avec annotations faibles	108
4.3.5	Conclusion	109
4.4	Morphologie mathématique et apprentissage profond	111
4.4.1	Opérateurs morphologiques	111
4.4.2	Pré-traitements des images	113
4.4.3	Réseaux de neurones et couches morphologiques	117
4.4.4	Structures hiérarchiques et opérateurs connexes	119
4.4.5	Pré-traitement des images par opérateurs connexes	123

4.4.6	Pré-traitement des images par extraction d'attributs	125
4.4.7	Conclusion	132

Dans ce chapitre sont présentées nos principales contributions. Nous détaillons les méthodes employées, dont une partie est issue de méthodes préexistantes, mais aussi des propositions de processus nouveaux. Nous présentons les expérimentations mises en œuvre afin d'évaluer les améliorations proposées dans le but d'obtenir une segmentation sémantique (voir chapitre 1) pour nos images.

Tout d'abord, nous détaillons le protocole expérimental et les données utilisées pour nos expériences. La seconde section concerne la segmentation d'image automatique basée sur les méthodes d'apprentissage profond adaptées à notre contexte. Nos images sont des volumes 3D ; nous comparons alors l'utilisation de modèles 2D et 3D, la taille des patches utilisés pour l'entraînement ainsi que les variations de composition de ces modèles. Nous souhaitons segmenter plusieurs organelles ; nous évaluons donc les approches multi-classes et binaires et l'effet des fonctions de perte sur les différents objets d'intérêt. De plus, nous travaillons avec des jeux de données composés de peu d'images ; nous étudions alors l'impact de la taille des jeux de données sur les segmentations ainsi que les approches utilisant des modèles pré-entraînés. Nous présentons également une expérimentation proposant d'utiliser une méthode d'adaptation de domaine avec un CycleGAN. Ensuite, la troisième section s'intéresse aux méthodes interactives et faiblement supervisées. Enfin, dans la dernière section, nous nous concentrons sur les méthodes utilisant à la fois apprentissage profond et morphologie mathématique.

La figure A.25 représente schématiquement l'exploration scientifique suivie au cours de la thèse.

4.1 Protocole expérimental

Dans cette section, nous présentons le protocole expérimental mis en place afin de comparer quantitativement les méthodes et variations de paramètres. Ce protocole prend en compte les aspects pseudo-aléatoires de l'entraînement et l'interdépendance des modifications apportées aux méthodes. Le résultat final d'un entraînement de modèle dépend de plusieurs paramètres aléatoires comme l'initialisation des poids du modèle, l'ordre dans lequel les données d'entraînement sont utilisées ou encore les augmentations de données aléatoires appliquées. Il est possible de supprimer les variations aléatoires d'un entraînement en fixant les paramètres d'initialisation des générateurs pseudo-aléatoires, mais cela ne permet pas de comparer correctement des méthodes, uniquement de prouver une réplication. Dans la plupart de nos expérimentations, en répétant suffisamment de fois les entraînements, on peut trouver un résultat où une méthode A est meilleure qu'une méthode B pour un individu donné, sans pour autant que la méthode soit statistiquement meilleure.

Dans un monde où les ressources seraient illimitées, nous pourrions répéter les entraînements des milliers de fois, mais cela n'est pas possible en réalité. En moyenne,

un entraînement complet dure entre 6 et 24 heures ; il est donc nécessaire de trouver un compromis entre ressources disponibles et taille de l'échantillon. Pour comparer nos méthodes, nous répétons les expériences, plusieurs fois ; ce nombre varie, mais est généralement compris entre 5 et 10.

Les résultats sont donnés sous forme de boîtes à moustache des scores. Comme les résultats de nos expériences sont calculés sur des effectifs relativement faibles, des valeurs extrêmes pourraient avoir un impact très fort sur le calcul de la moyenne et de l'écart-type. Les boîtes à moustache représentent les valeurs correspondant au minimum, maximum, médiane, première et troisième quartile.

Dans certain cas, des cercles présents en dehors des distributions correspondent à des valeurs extrêmes. Les moustaches ne s'étendent pas plus qu'à un certain point de l'écart inter-quartile. Par défaut, la bibliothèque logicielle Python Matplotlib utilise un paramètre égal à 1,5. Les moustaches ne s'étendent alors pas plus qu'à $1,5 * IQR$ avec $IQR = Q3 - Q1$. Si un élément est à une distance supérieure, il sera représenté sous forme d'un point blanc. Nous avons choisi de garder ce paramètre par défaut afin de rendre nos travaux les plus aisés à lire pour les lecteurs ayant l'habitude de ce format.

Les expérimentations ont été effectuées tout au long de la thèse, ce qui explique certaines variations de paramètres, des jeux de données n'existant pas encore au moment où l'expérimentation a été effectuée.

Pour toutes nos expériences, nous utilisons l'optimiseur Adam [Kingma and Ba, 2015], une extension de l'algorithme du gradient stochastique permettant des convergences plus rapides. C'est actuellement l'algorithme d'optimisation le plus utilisé dans la communauté et nous l'utilisons avec les paramètres par défaut ($\alpha = 0,001$, $\beta_1 = 0,9$, $\beta_2 = 0,999$, $\epsilon = 1e-07$). Nous appliquons également systématiquement des augmentations aux données, rotations et symétries. Nous avons choisi de ne pas utiliser l'augmentation basée sur les déformations en raison de la nature de nos données pour lesquelles la forme est très importante pour caractériser les objets.

Jeux de données

Chaque jeu de données est séparé en trois parties, qui correspondent à trois phases d'utilisation d'un modèle.

- Train : la partie utilisée pour entraîner le modèle.
- Valid : la partie de validation, pouvant être utilisée pour choisir des paramètres ou sélectionner un état d'entraînement.
- Test : la partie utilisée uniquement pour tester les modèles et produire des résultats quantitatifs.

La partie de test doit impérativement être disjointe des autres parties afin d'évaluer correctement les performances du modèle sur de nouvelles données, des données non utilisées pour l'entraînement. Il est également préférable que les deux autres parties, train et valid, soient disjointes, dans le but d'éviter le sur-apprentissage. Nous utilisons par exemple le jeu de validation pour évaluer le modèle entraîné à chaque époque et ainsi sélectionner le modèle intermédiaire avec les meilleures performances.

Nous séparons spatialement au maximum le jeu d’entraînement du jeu de test. Les coupes sélectionnées pour faire partie de ces ensembles sont volontairement écartées par l’ensemble de validation. Le tableau 4.1 résume les noms et informations sur les jeux de données utilisés dans la section 4.2. Une version plus détaillée est disponible en annexe (A.2). Dans les sections suivantes, nous référencerons directement les noms des images, sous-entendant le jeu de données complet (chapitre 3).

Jeu de données	Image	Coupes			Classes				
		Train	Valid	Test	Mi	Re	Ce	Nu	En
I3-ISBI	I3	40	20	19	x	x	x	x	
LW4-ISBI	LW4-1	40	20	20	x	x	x		x
IGBMC-S-multi	I3	40	20	20	x	x			
	LW4-1	40	20	20					
IGBMC-620-MITO	I3	490	5	5	x				
	LW4-1	111	4	5					
I3-ISO	I3	800	100	100	x				
LW4-ISO	LW4-1	80	40	40	x	x	x		x
JG1-Mito 5×384	JG1	1536	192	192	x				
MitoEM-H	MitoEM-H	400	100		x				

TABLE 4.1 – Résumé des noms donnés aux jeux de données utilisés. Pour les jeux de données comportant le suffixe « ISO », les images ont été rendues isotropes en générant de nouvelles coupes par interpolation.

4.2 Segmentation automatique avec annotations denses

Dans cette section, nous étudions des variations dans nos méthodes de segmentation en nous inspirant de l'état de l'art. Nous nous focalisons sur les modèles dérivés de l'architecture U-Net, un modèle éprouvé et performant. L'objectif est de déterminer la meilleure approche à adopter pour nos images bruitées, faiblement contrastées et contenant de nombreux objets complexes. Nous étudions également l'impact de la quantité de données annotées sur les résultats de nos modèles.

Notre étude est articulée autour de sept idées.

1. Modèles multi-classes ou binaires.
2. Modèles 2D ou 3D.
3. Variations de couche pour l'architecture U-Net.
4. Taille des patches.
5. Taille du jeu de données.
6. Fonction de perte.
7. Modèle pré-entraîné.

Dans le chapitre 2, nous avons vu que différentes variations de méthodes sont proposées pour la segmentation d'images, en particulier pour les images de microscopie électronique. La plupart des améliorations sont interdépendantes ; il serait trop coûteux en nombre d'expériences et donc en heures de calcul d'évaluer toutes les variations en même temps. Nous avons évalué la plupart des variations de manière indépendante, mais également répété dans d'autres contextes les expériences dont les conclusions étaient peu significatives. Pour rendre l'enchaînement des sous-sections plus agréable à la lecture, l'ordre des sous-sections n'est pas celui des expériences qui ont été menées tout au long de la thèse, mais permet d'en faire une synthèse de manière plus compréhensible. Dans chaque cas, nous présenterons les résultats d'une expérience, mais la plupart des expériences ont été répétées sur plusieurs jeux de données avec différentes variations.

Les caractéristiques des expériences de cette première section sont résumées dans la table 4.2. Les colonnes de la table correspondent aux informations suivantes :

- Rep : nombre de modèles entraînés et évalués pour chaque cas ;
- Archi : le nom de l'architecture utilisée ;
- Loss : le nom de la fonction de perte utilisée ;
- Patch : la taille des patches utilisés pour l'entraînement ;
- Batch : la taille des lots utilisés, en anglais, la batch size ;
- Data : le jeu de données utilisé.

4.2.1 Modèles multi-classes et binaires

Dans cette première sous-section, nous étudions les différences entre modèles multi-classes et binaires pour la segmentation des images. Notre objectif est de produire une

Partie	Rep	Archi	Loss	Patch Batch	Data
4.2.1	6	U-Net	L_{Dist}	256×256 32	IGBMC-S-multi
4.2.2	5	U-Net	L_{Dice} L_{Dist}	256×256 6 $16 \times 192 \times 192$ 1 $80 \times 80 \times 80$ 1	LW4-ISO
4.2.3	6	*	L_{Dist}	256×256 16 $32 \times 256 \times 256$ 1 $32 \times 128 \times 128$ 4	MitoEM-H
4.2.4	10	CNN	L_{CE}	* 16	IGBMC-620-MITO
4.2.5	10	U-Net	L_{Dist}	256×256 8	I3-ISO * JG1-Mito 5×384 *
4.2.6	10*	U-Net	*	512×512 4 $32 \times 256 \times 256$ 1	MitoEM-H
4.2.7	15	U-Net Eff-U-Net	L_{Dice}	256×256 4	I3-ISBI LW4-ISBI

TABLE 4.2 – Résumé des caractéristiques des expériences de la section segmentation automatique. Le symbole * signifie que l’expérience comporte des variations ou détails supplémentaires dans la section en question.

segmentation multi-classe de nos images. Le nombre de classes dépend des images et des annotations que nous utilisons sur les images en question.

Pour obtenir la segmentation, deux approches sont envisageables. D’un côté, nous pouvons utiliser un modèle multi-classe, qui est entraîné à prédire sur toutes les classes en même temps. Le modèle prédira un vecteur de longueur $|C|$ pour chaque pixel. D’un autre côté, nous pouvons entraîner un modèle binaire différent pour chaque classe et utiliser plusieurs modèles binaires indépendants afin de prédire chaque classe indépendamment.

L’approche binaire est intéressante pour deux raisons. Premièrement, un modèle binaire est techniquement plus facile à entraîner, il n’est pas nécessaire de s’assurer que les différentes classes soient correctement segmentées en même temps avec une seule fonction de coût. Ensuite, il est possible de manière triviale d’adapter la quantité de données pour chacune des classes, il n’est pas nécessaire qu’une coupe soit annotée pour toutes les classes pour qu’elle soit utilisable. Il y a cependant besoin de mettre en place une stratégie de fusion des résultats et il est indispensable d’entraîner à nouveau tous les modèles lorsque l’on souhaite segmenter une nouvelle image.

Expérience

Dans cette expérience, nous comparons les segmentations produites par des modèles multi-classes à des segmentations produites par des modèles binaires.

Nous utilisons quatre modèles (deux modèles binaires et deux modèles multi-classes) et les entraînons à segmenter les images *I3* et *LW4-1* simultanément. Les deux modèles binaires correspondent aux deux classes, « Binaire-M » pour la classe des mitochondries

et « Binaire-R » pour la classe du réticulum endoplasmique (RE). Les deux cas multi-classes (Multi- α et Multi- β) ne sont pas implémentés de la même façon, mais sont très proches. Dans le cas α , on calcule les coûts en une seule fois alors que dans le cas β , le modèle possède des sorties indépendantes et on calcule les coûts pour chacune des classes séparément avant d'en faire la moyenne. Le cas β était originalement prévu dans le but de pondérer les classes, mais cette fonctionnalité n'a pas été nécessaire.

La figure 4.1 regroupe les résultats de cette expérience. Les mesures de distance sont similaires et cohérentes avec les mesures de F1-score. Le F1-Score est compris entre 94% et 97% pour la classe des mitochondries et entre 71% et 75% pour le RE. Pour l'image $I\beta$, les scores médians sont proches et d'environ 95% pour les mitochondries et de 73% pour le RE. Pour l'image $LW4-1$, les scores médians sont également proches, mais les distributions des résultats sont plus restreintes.

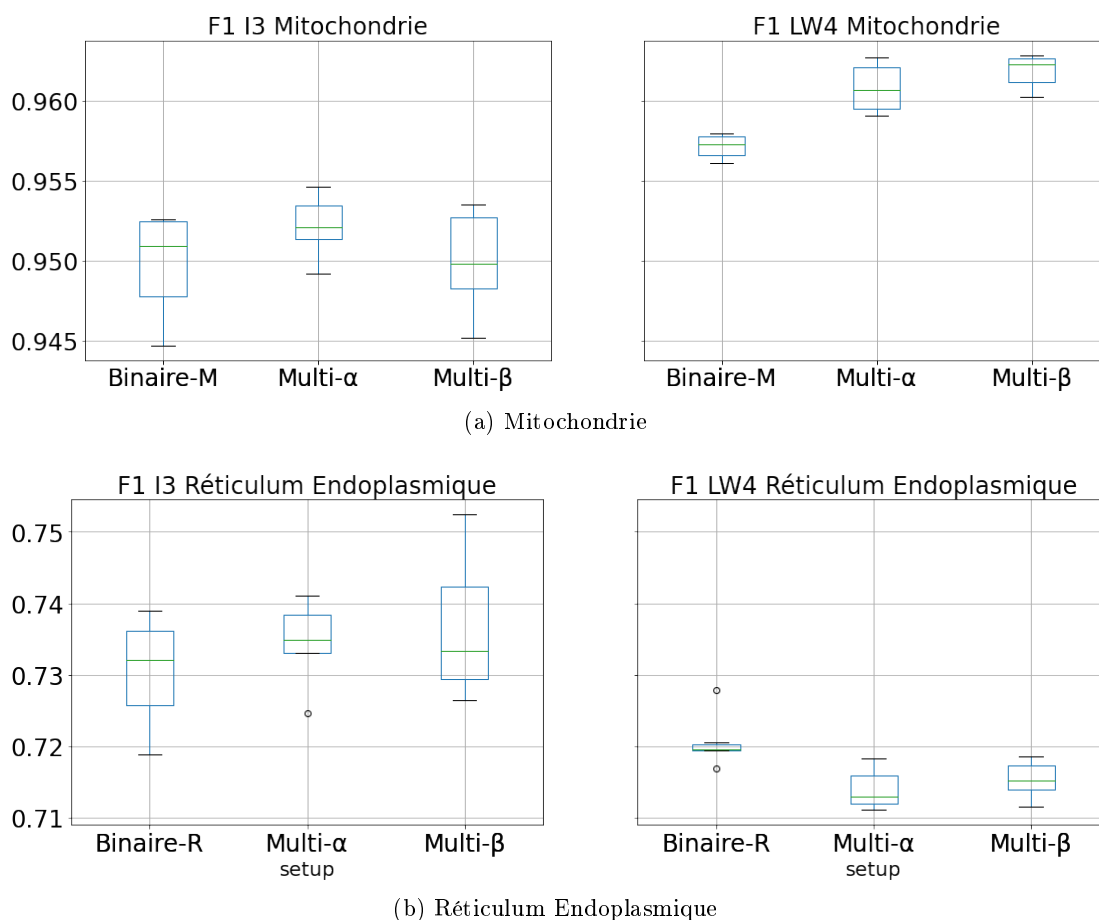


FIGURE 4.1 – Résultats de l'expérience binaire et multi-classes.

Les modèles multi-classes donnent généralement de meilleurs résultats avec des jeux de données égaux, mais la création de jeux d'entraînement est plus longue et difficile. Les modèles binaires peuvent ainsi donner de meilleurs résultats sur des classes spé-

cifiques où l'on aurait plus de données annotées. Nous avons réalisé un essai où l'on reproduit l'expérience en donnant à un modèle binaire 500 coupes avec les annotations des mitochondries pour l'image *I3*. Le modèle binaire est alors meilleur que les modèles multi-classes présentés.

Néanmoins, il est important de remarquer que les différences sont très faibles, et que dans la plupart des cas, l'amélioration n'est pas significative.

Le choix entre modèles binaires et multi-classes dépend principalement du cas d'utilisation et des données disponibles. S'il n'existe aucune donnée, et que l'objectif à long terme est de segmenter automatiquement plusieurs images, les modèles multi-classes sont à privilégier. Si l'objectif est de segmenter rapidement quelques classes afin d'analyser une image complexe, l'utilisation de plusieurs modèles binaires semble plus indiqué.

Dans la suite, nous utiliserons à la fois des modèles multi-classes et des modèles binaires, ces derniers permettant de nous concentrer sur une classe en particulier et d'éviter d'introduire des biais (améliorations différentes sur les classes distinctes) dans nos expériences.

4.2.2 Modèles 2D et 3D

Dans cette sous-section, nous comparons l'utilisation de modèles 2D et 3D. Nos images sont des volumes 3D, en l'occurrence des images anisotropes dont l'anisotropie est connue. Cette information tridimensionnelle peut être intéressante pour correctement segmenter certaines organelles ; c'est ce que fait en pratique l'expert lorsqu'il annote les images.

Les modèles 3D sont les plus présents dans l'état de l'art, et c'est la capture d'informations 3D qui est mise en avant par les auteurs de [Çiçek et al., 2016] ; auteurs qui introduisent pour la première fois les modèles U-Net 3D. Les modèles 3D peuvent capter et effectuer des traitements sur l'information tridimensionnelle d'une image, alors que ce n'est pas le cas pour les modèles 2D.

Dans un contexte de ressources finies et limitées, un compromis entre les ressources disponibles, les tailles d'entrées et de modèle sont nécessaires. Un modèle plus « grand » est un modèle composé de plus de paramètres. Dans notre cas, nous avons défini les limites matérielles dans la section 1.2. Pour rappel, le GPU proposé est une NVIDIA GeForce RTX 2080 Ti avec 11 Go de VRAM.

Expérience

Dans cette expérience, nous comparons modèles 2D et 3D. Il n'est pas trivial de comparer ces modèles tout en les mettant au même niveau. Dans notre cas, nous avons choisi de comparer 5 cas, 1 modèle 2D et 2 modèles 3D qui sont entraînés avec 2 tailles de patch différentes. Les modèles ont été dimensionnés de manière à avoir un nombre de paramètres similaire et qui correspond à un compromis entre notre limite matérielle et une taille de patch 3D convenable. La nomenclature utilisée pour les noms de modèle est la suivante : unet[2d|3d]-[profondeur]-[nombre de filtres niveau 1]. Les modèles sont les suivants :

- unet2d-5-32 : 34,6 millions de paramètres ;

- unet3d-4-38 : 34,6 millions de paramètres ;
- unet3d-5-19 : 34,0 millions de paramètres.

Pour les modèles 3D, on entraîne deux versions avec des tailles de patches différentes et un seul patch par lot.

La première version utilise des patches de taille $16 \times 192 \times 192$, dans le but de conserver un champ récepteur 2D suffisamment grand. La seconde utilise des patches cubiques de taille $80 \times 80 \times 80$. Dans le cas 2D, on entraîne le modèle avec des patches de taille 256×256 et une taille de lot de 6. Au final, les modèles ont un nombre similaire de paramètres et une taille d'entrée similaire, compensée par la taille de lot dans le cas 2D. Ce choix permet de comparer les différentes méthodes dans le cas d'une limitation matérielle similaire.

Les figures 4.2 à 4.5 montrent les résultats de cette expérience et la figure 4.6 regroupe l'intégralité des scores médians. Pour la classe des mitochondries, les meilleurs modèles atteignent un F1-Score médian d'environ 95% avec une variance faible et les pires modèles entraînent une baisse de score d'environ 10%. La fonction de perte a un impact assez faible. Pour le RE, le score médian des meilleurs modèles est plus proche de 74%, mais la fonction de perte L_{Dice} permet une amélioration de +3% à +15%.

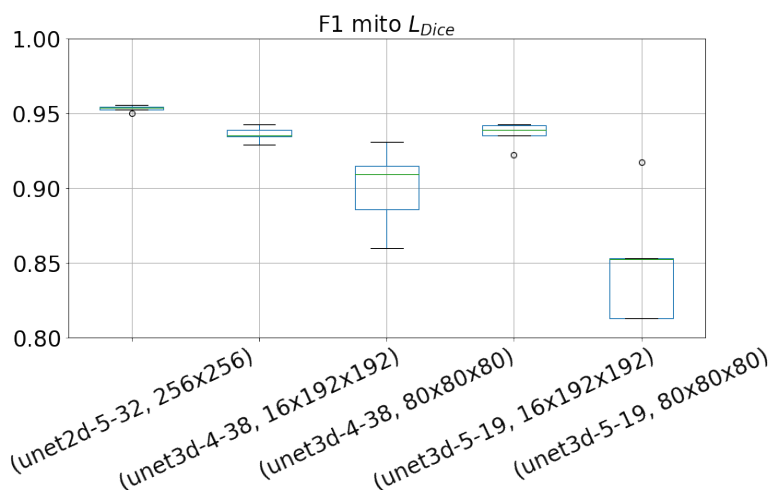


FIGURE 4.2 – Résultats de l'expérience 2D et 3D, classe mitochondrie, fonction de perte L_{Dice} .

Comme nos résultats ne correspondent pas à ceux anticipés après notre analyse de l'état de l'art, c'est l'expérience que nous avons probablement la plus répétée, dans différents cadres, sur différents jeux de données, avec des résultats toujours cohérents avec notre conclusion.

Si l'on met les deux méthodes à égalité concernant le nombre de paramètres et la taille des entrées, les modèles 2D surpassent les modèles 3D pour la segmentation de nos images. Si l'on s'intéresse à la figure de synthèse (figure 4.6), on peut remarquer que les modèles utilisant la fonction de perte L_{Dice} sont systématiquement plus performants que leurs variations utilisant L_{Dist} pour la classe réticulum endoplasmique. De manière

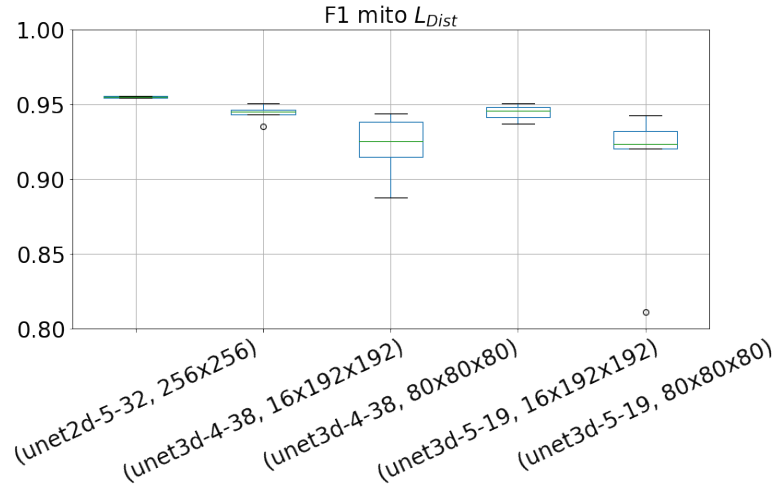


FIGURE 4.3 – Résultats de l’expérience 2D et 3D, classe mitochondrie, fonction de perte L_{Dist} .

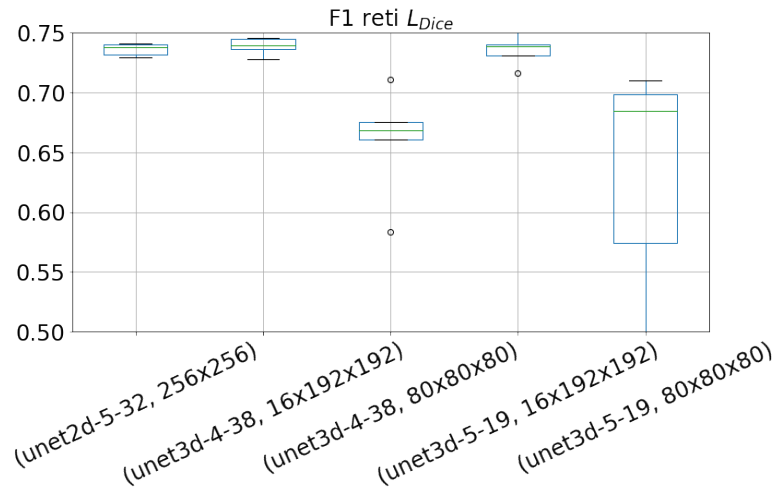


FIGURE 4.4 – Résultats de l’expérience 2D et 3D, classe réticulum endoplasmique, fonction de perte L_{Dice} .

inverse, dans les cas 3D, les modèles utilisant la fonction de perte L_{Dist} sont plus efficaces pour la classe mitochondrie. En somme c’est le modèle 2D entraîné avec la fonction L_{Dice} qui est le plus performant. Néanmoins, cette conclusion n’est valide que si l’on applique notre prérequis de limite matérielle. Dans l’absolu, un modèle 3D pourrait être plus performant qu’un modèle 2D et c’est d’ailleurs ce qui est souvent présenté dans l’état de l’art : des modèles 3D plus performants, mais composés de plus de paramètres ou avec une taille d’entrée plus grande. Dans notre expérience, lors de la prédiction, les modèles

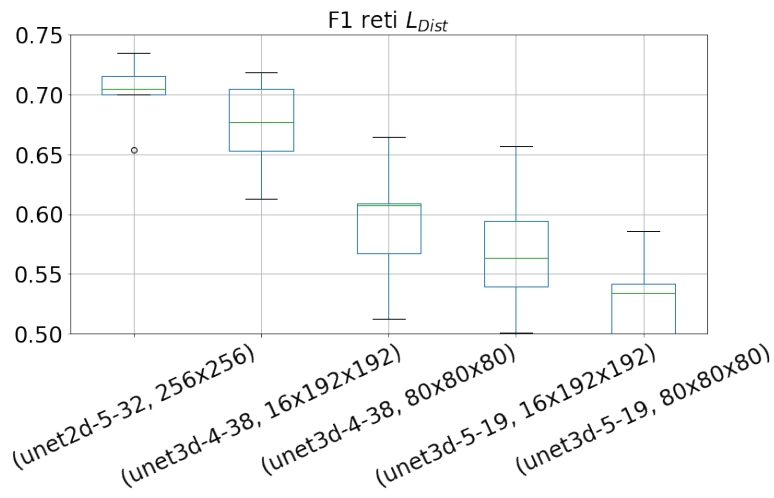


FIGURE 4.5 – Résultats de l'expérience 2D et 3D, classe réticulum endoplasmique, fonction de perte L_{Dist} .

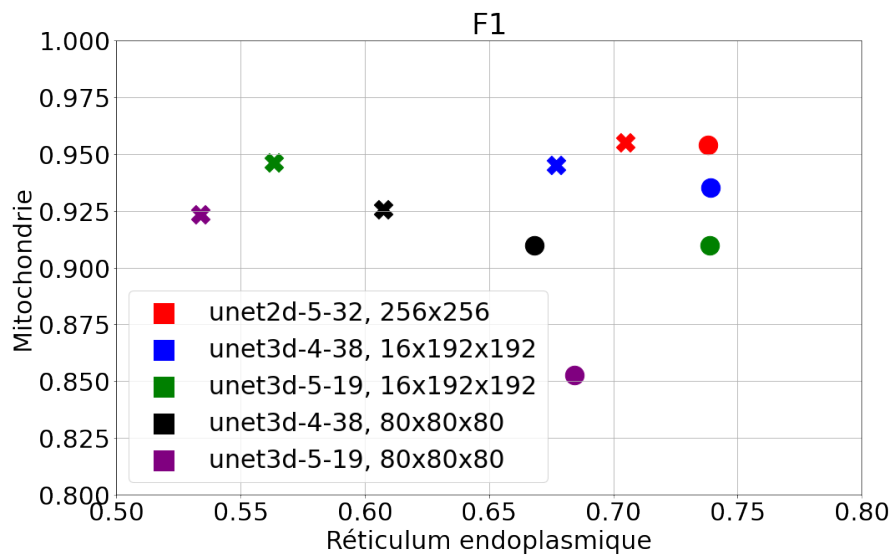


FIGURE 4.6 – Résultats de l'expérience 2D et 3D, bilan des scores médians. Les ronds correspondent à la fonction de perte L_{Dice} et les « × » correspondent à la fonction de perte L_{Dist} .

2D ne peuvent capter d'informations tridimensionnelles, mais ils peuvent en contrepartie capturer une zone 2D plus grande.

Il serait possible que la haute définition de nos images et la taille des objets à segmenter empêche le modèle 3D d'avoir suffisamment de contexte (une mitochondrie mesure environ 150×150 pixels dans l'image LW_4-1), mais les résultats comparant différentes

variations de modèle U-Net (sous-section 4.2.3, figures 4.10 et 4.11) et différentes tailles de patch (sous-section 4.2.4) indiquent le contraire et corroborent nos résultats. Il est également envisageable que la quantité de données nécessaire à l’entraînement d’un modèle 3D soit supérieure à celle d’un modèle 2D, et que la quantité utilisée pour l’entraînement dans notre expérience ne soit pas suffisante. Enfin, il est intéressant de noter que pour la classe du réticulum endoplasmique, donc avec des objets plus petits, le modèle unet3d-4-38 égale, voire surpasse le modèle 2D dans certains cas.

4.2.3 Variations du modèle U-Net

Dans le chapitre 2, nous avons vu que différentes versions d’architectures sont utilisées pour la segmentation d’images. Dans cette sous-section, nous étudions différentes variations autour de l’architecture U-Net. Cinq architectures 2D et 3 architectures 3D sont utilisées. L’expérimentation à la fois en 2D et en 3D permet de mettre en évidence une amélioration si elle est commune (variations similaires en 2D et 3D suite à un ajout). Nous expérimentons la variation dans la composition de blocs, et dans les connexions entre blocs.

Dans le modèle U-Net classique [Ronneberger et al., 2015], les connexions entre blocs sont faites à l’aide de concaténation. Dans leur article, les auteurs de [Chaurasia and Culurciello, 2017] proposent d’utiliser des additions pour ce genre de connexions. Nous comparons donc différents types de connexions : concaténations (U-Net), additions (Link-Net) et multiplications (Mult-Net). La figure 4.7 représente schématiquement l’architecture U-Net. Ce sont les connexions, représentées en bleu foncé qui changent.

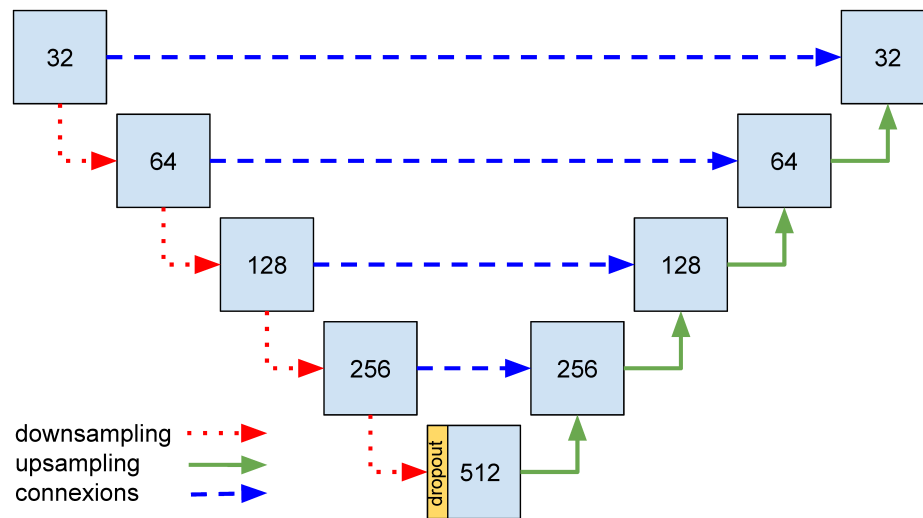


FIGURE 4.7 – Représentation schématique de l’architecture U-Net. Les nombres présents dans les blocs bleus représentent le nombre de filtres par couche de convolution.

Avec le modèle U-Net classique, les blocs sont composés de deux opérations successives, répétées deux fois : une convolution 3×3 et une couche d’activation utilisant la

fonction ReLu [Fukushima, 1975, Nair and Hinton, 2010]. Nous avons déjà validé une amélioration, utilisant des couches de normalisation de lots [Ioffe and Szegedy, 2015], qui sont tout à fait courantes dans l'état de l'art, ainsi qu'un dropout à l'entrée du bloc le plus profond [Srivastava et al., 2014]. En évaluant différentes architectures, nous avons montré dans [Meyer et al., 2021] qu'un modèle utilisant un encodeur EfficientNet [Tan and Le, 2019] permet d'améliorer la qualité des segmentations dans certains cas. Cependant, cette amélioration se fait au détriment de la simplicité de la méthode proposée, et donc de la simplicité d'utilisation et d'implémentation. Cette volonté de simplicité n'est pas centrale ni requise dans nos travaux ; pourtant il est important de la prendre en compte lorsque l'on propose quelque chose de nouveau. Pour qu'une amélioration soit utile et utilisée, elle ne doit pas être démesurément difficile à mettre en place techniquement par rapport à son apport.

Dans cet objectif, nous souhaitons évaluer l'utilisation de connexions résiduelles sur les blocs de traitements [He et al., 2016a]. La figure 4.8 représente les variations entre couche simple et résiduelle. La partie droite du bloc résiduel représente une opération de convolution sur un noyau 1×1 .

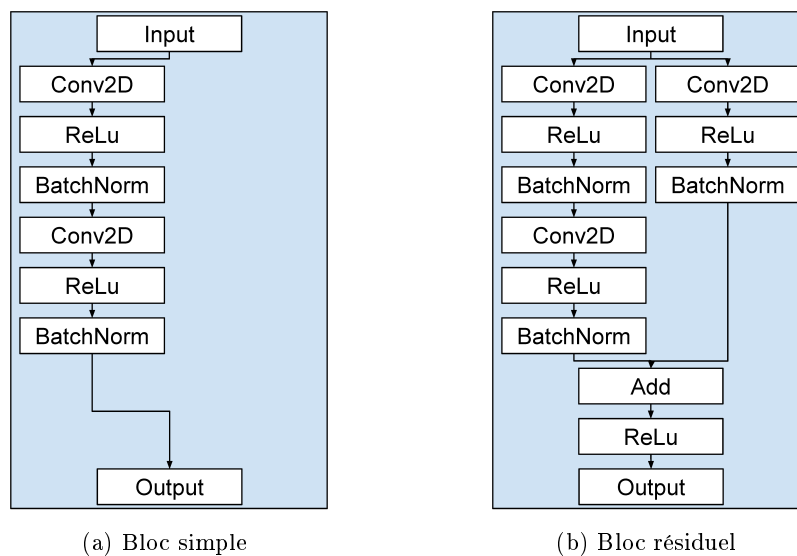


FIGURE 4.8 – Représentation schématique des blocs composant l'architecture U-Net.

Cette modification techniquement très simple à mettre en place permet selon les auteurs de [Li et al., 2018a] d'éviter la transition du problème d'optimisation convexe vers un comportement chaotique, ce qui explique un meilleur apprentissage. La figure 4.9 représente graphiquement les surfaces de la fonction de perte de deux modèles, un premier sans connexions résiduelles et un second avec.

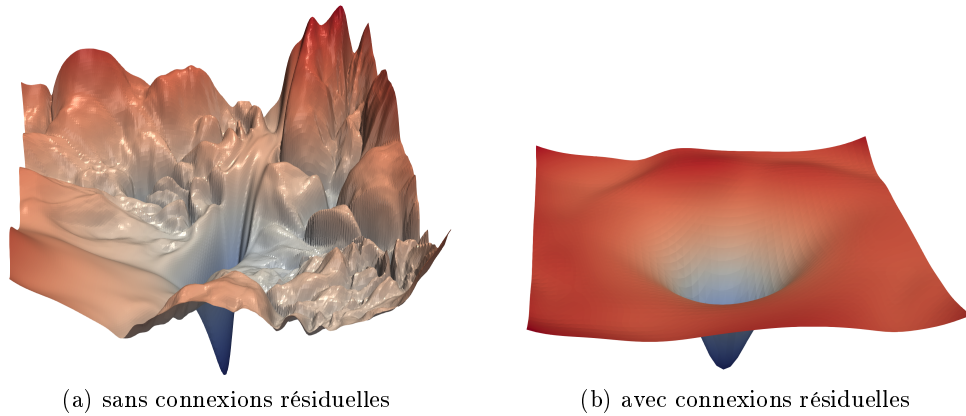


FIGURE 4.9 – Représentation graphique des surfaces de loss d'un ResNet-56 avec/sans connexions résiduelles [Li et al., 2018a].

Expérience

Dans cette expérience, nous comparons les différentes modifications d'architectures présentées précédemment. Les figures 4.10 et 4.11 représentent les résultats de cette expérience. Pour les meilleures méthodes (2D U-Net-Res 32 et 3D U-Net-Res 32), le F1-Score médian est d'environ 91.5% mais l'ASSD est 15% meilleure pour le cas 2D.

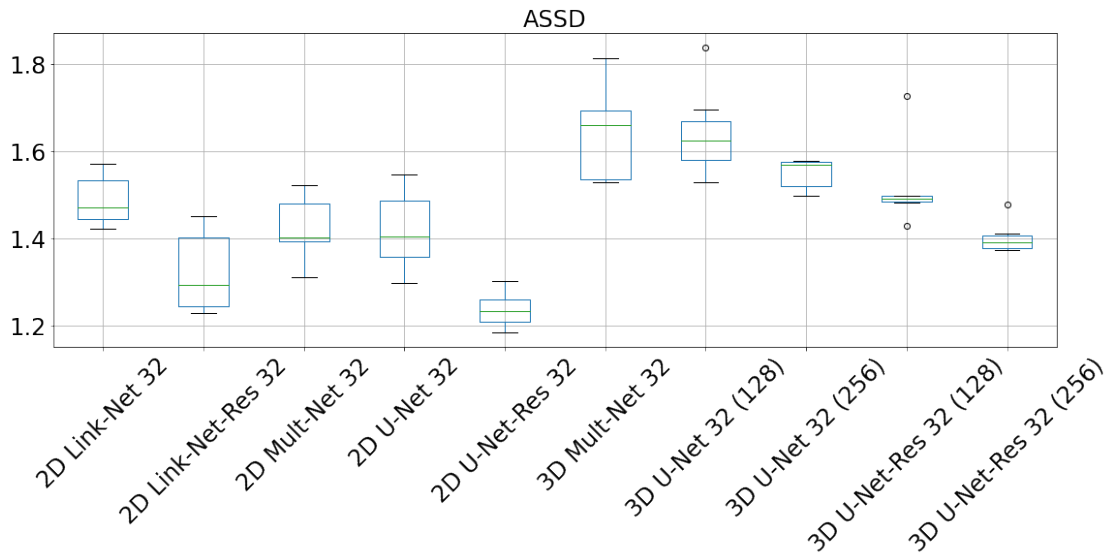


FIGURE 4.10 – Résultats de l'expérience de variations du modèle U-Net, ASSD. Rappel : une valeur plus petite indique une meilleure segmentation.

Les résultats de l'expérience nous permettent plusieurs observations. Premièrement, l'utilisation de connexions résiduelles permet une amélioration systématique et signifi-

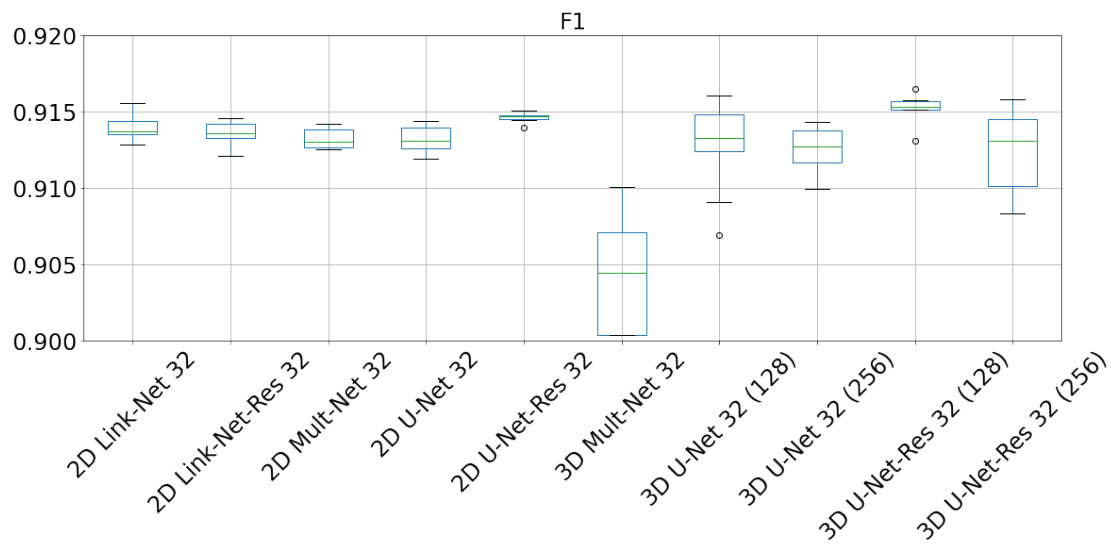


FIGURE 4.11 – Résultats de l'expérience de variations du modèle U-Net, F1 Score.

cative, tout en ayant un nombre de paramètres relativement proches. Avec des U-Net utilisant un plus grand nombre de filtres, la différence en nombre de paramètres devient de plus en plus faible (U-Net 32 +9%, U-Net 64 +4,5%). La table 4.3 représente le nombre de paramètres et la durée d'entraînement minimum (pas d'arrêt anticipé ou early stopping dans cette expérience) pour les variantes les plus intéressantes.

Deuxièmement, malgré l'augmentation faible du nombre de paramètres, l'augmentation en temps d'entraînement est, quant à elle, bien plus impactée (cet impact se réduit également avec de plus gros modèles). Cependant, ces chiffres restent difficiles à interpréter, car les modèles ont été entraînés sur un centre de calcul, et différentes configurations matérielles ont pu être utilisées, raison pour laquelle la valeur de temps minimale est utilisée dans la table 4.3.

Troisièmement, les modifications des connexions entre blocs sont soit inutiles (Link-Net), soit délétères (3D Mult-Net).

Enfin, nous pouvons confirmer la conclusion précédemment énoncée dans la sous-section 4.2.2 concernant les modèles 2D et 3D ; les modèles 2D sont plus efficaces dans notre contexte.

Architecture	Nombre de paramètres (millions)	Durée d'entraînement (seconde)
2D U-Net 32	7,8	23 285 (6,5 heures)
2D U-Net-Res 32	8,5	33 439 (9,3 heures)
3D U-Net 32 (128)	22,6	120 445 (33,5 heures)
3D U-Net-Res 32 (128)	22,9	128 553 (35,7 heures)
2D U-Net 64	31,0	
2D U-Net-Res 64	32,4	

TABLE 4.3 – Nombre de paramètres et temps d'entraînement minimaux enregistrés pour différentes architectures. Les nombres de paramètres pour les U-Net 64 sont donnés comme simple indication, ces modèles étant utilisés dans d'autres expériences.

4.2.4 Taille de patch 2D pour l'entraînement

Dans cette sous-section, nous étudions l'impact de la taille des patches utilisés pour l'entraînement sur les performances des modèles.

Comme nous l'avons vu dans la sous-section 4.2.2, l'avantage principal des modèles 2D est de nous permettre l'utilisation de patch capturant une plus grande surface. Lors de l'entraînement, plus la taille de patch est grande, plus la zone de prédiction non impactée par les paddings du modèle est grande. Plus la taille de patch est petite, plus le nombre de patch par lot peut être grande. Avec une trop petite taille de lot, la descente de gradient risque d'être bruitée [Goodfellow et al., 2016]. Cependant, il a également été observé en pratique que l'utilisation de tailles de lot plus grandes conduit à une dégradation significative de la qualité de généralisation des modèles [Keskar et al., 2016].

L'objectif dans notre cas est de connaître la taille de patch minimale, c'est-à-dire le plus petit patch pour lequel l'entraînement n'est pas négativement perturbé. Il y a deux raisons qui nous poussent à souhaiter trouver cette valeur minimale. Premièrement, cela nous garantira que notre choix de taille de patch n'est pas trop petite. Deuxièmement, cela pourra nous indiquer une taille minimale de surface qui serait nécessaire pour que les modèles 3D puissent capturer le minimum d'information 2D nécessaire.

Lors de la phase de prédiction, il n'y a aucun intérêt à utiliser des patches plutôt que la tranche entière si cela est techniquement possible. Si l'étape d'apprentissage consomme beaucoup de mémoire, ce n'est pas le cas pour l'étape d'inférence. Pour toutes les images que nous avons utilisées, il a la plupart du temps été possible de prédire les coupes en entier dans le cas 2D. S'il n'est pas possible de prédire la coupe en une seule fois, on utilisera alors uniquement les pixels non impactés par les paddings en recadrant la sortie du modèle.

Expérience

Pour évaluer l'impact de la taille des patches, nous avons utilisé une méthode différente des méthodes précédentes. Plutôt que d'entraîner un modèle à segmenter, nous avons entraîné un CNN à classifier le pixel central. Lorsqu'on utilise un CNN, la totalité de

l'image donnée en entrée peut être utilisée dans le processus de classification d'un seul élément. Cette méthode nous permet d'interpréter plus simplement les résultats. Les tailles de patch proposées sont les suivantes :

- 23×23
- 41×41
- 67×67
- 97×97
- 127×127
- 199×199
- 257×257

La figure 4.12 représente l'exactitude maximale des 10 modèles entraînés. Nous avons défini l'exactitude dans l'équation (3.5), section 3.2. L'exactitude maximum correspond à la valeur d'exactitude du meilleur des 10 modèles. On peut observer une croissance logarithmique de l'exactitude sur l'intervalle $[23, 127]$, puis on atteint un plateau avec une exactitude de 97.5%.

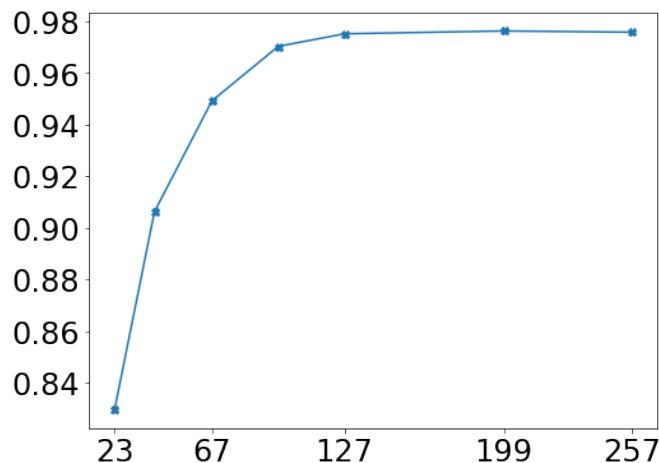


FIGURE 4.12 – Résultats de l'expérience sur les tailles de patch, exactitude maximale.

Cette expérience nous permet de montrer qu'une taille de patch de 127×127 est suffisante pour obtenir de bons résultats, les tailles de patch plus grandes n'améliorant pas l'exactitude du modèle. Transposé à notre problème de segmentation, cela signifie qu'il n'est pas nécessaire d'avoir plus de 64 pixels de contexte dans chaque direction pour segmenter nos images.

Si l'on met en relation ces résultats avec ceux de la sous-section 4.2.2, cela peut expliquer les moins bons résultats des modèles 3D avec des patches de taille $80 \times 80 \times 80$ mais pas ceux pour les patches $16 \times 192 \times 192$.

4.2.5 Taille du jeu d'entraînement

Dans cette sous-section, nous nous intéressons à l'impact de la taille des jeux d'entraînement utilisés sur les performances des modèles entraînés.

La production de données d'entraînement est une phase fastidieuse et coûteuse en temps. Une interrogation régulière des biologistes avec lesquels nous avons travaillé est de connaître la quantité d'annotations nécessaires pour pouvoir entraîner un modèle de segmentation. La réponse à cette question est loin d'être triviale et dépend de plusieurs paramètres : la qualité de l'image à segmenter, les objets que l'on souhaite segmenter et ce que l'on entend par « une bonne segmentation ». Dans cette sous-section, nous étudions l'impact de la quantité de données utilisées pour l'entraînement sur la qualité de nos segmentations.

Pour cette expérience, nous comparons les performances des modèles en faisant varier le nombre de coupes utilisées pour l'entraînement. Nous répétons l'opération deux fois sur deux images différentes. Dans le premier cas, nous utilisons une version isotrope par interpolation de l'image *I3* et un nombre de coupes allant de 1 à 800. Dans le second cas, nous utilisons l'image *JG1* avec un nombre de coupes croissant par bloc. Lorsque l'on dit utiliser 1 coupe, cela signifie que nous utilisons une coupe provenant de chacun des 4 blocs d'entraînement.

Les figures 4.13 et 4.14 montre les résultats de cette expérience. Les résultats médians sur le jeu *I3-ISO* sont compris entre 81% et 93% avec un maximum atteint dans le cas où le nombre de coupes utilisées pour l'entraînement est égal à 100. Pour le jeu *JG1-Mito*, les scores médians sont compris entre 33% et 68% avec un maximum dans le cas où le nombre de coupes utilisées est maximal.

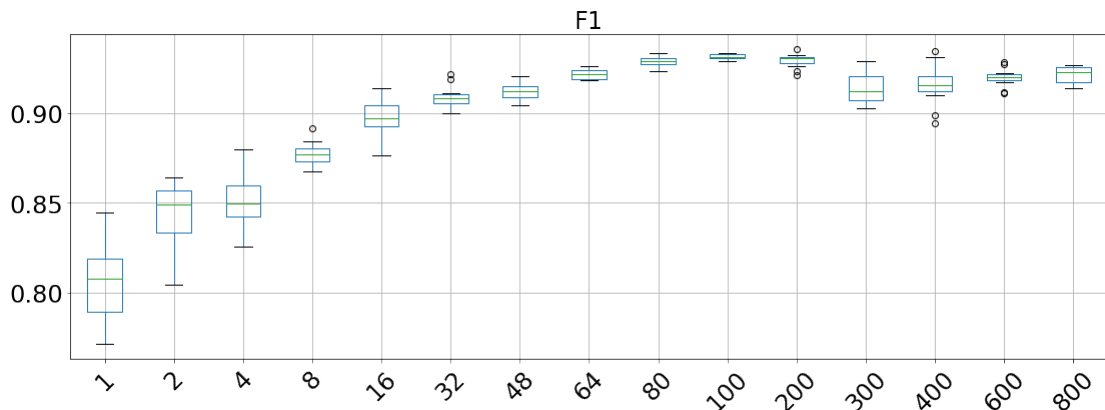


FIGURE 4.13 – Résultat de l'expérience sur la taille du jeu d'entraînement *I3-ISO*, F1 Score en fonction du nombre de coupes par bloc utilisées pour l'entraînement.

Dans le premier cas étudié, on note une dégradation des performances lorsque l'on dépasse les 100 coupes. Ce résultat peut être expliqué par un biais lié aux choix de coupes, la densité de mitochondries dans l'image *I3* est décroissante par rapport à l'axe *Z*. Lorsque l'on utilise plus de 100 coupes, on a déjà atteint un optimum, mais la dégradation de

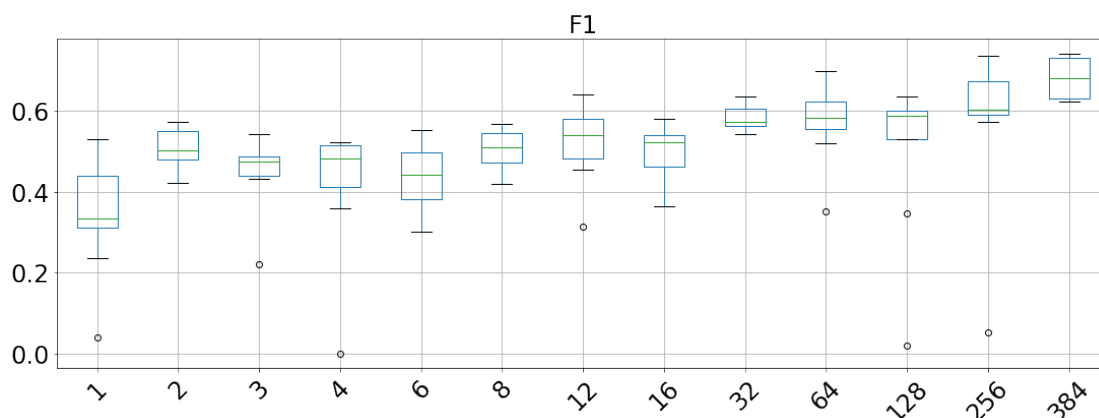


FIGURE 4.14 – Résultats de l’expérience sur la taille du jeu d’entraînement *JG1-Mito* 5×384 , F1 Score en fonction du nombre de coupes par bloc utilisées pour l’entraînement.

la qualité des données ne permet plus d’améliorations. De plus, les coupes de test sont placées au début de l’image, donc la corrélation avec les premières coupes d’entraînement est plus grande.

Dans le second cas utilisant l’image *JG1*, les résultats, bien que bruités nous montrent une amélioration significative et croissante. Des améliorations supplémentaires seraient probables si l’on ajoutait des données à notre jeu d’entraînement.

Cette expérience nous permet de remarquer que la quantité et la qualité des données utilisées sont toutes les deux importantes, mais sans réellement permettre de définir un seuil général pour notre problématique, qui dépend de la qualité de segmentation recherchée. En pratique pour la suite, nous utilisons toutes les données que nous avons.

4.2.6 Fonction de perte

Dans cette sous-section, nous étudions l’impact du choix de la fonction de perte ainsi que de ces hyper-paramètres sur l’apprentissage. Nous avons essayé différentes fonctions de perte avant de sélectionner les plus prometteuses à comparer dans nos expériences. Les fonctions retenues sont L_{Dice} , L_{Dist} et L_{IoU} définies précédemment dans la sous-section 2.1.1, page 24.

Expérience 1

Dans cette expérience, nous entraînons des modèles en utilisant les différentes fonctions de perte. Pour le calcul de la fonction de perte L_{Dist} , deux paramètres d’échelles appelé s_{in} et s_{out} sont utilisés afin de diviser la valeur du résultat de la transformation de distance et ainsi rendre la pente autour du bord de l’objet plus ou moins forte, à l’intérieur et à l’extérieur. Les deux paramètres peuvent être utilisés indépendamment pour l’intérieur et l’extérieur de l’objet. Ces paramètres sont représentés dans le nom de la fonction de perte au format $L_{Dist}^{s_{out}}_{s_{in}}$. Dans le cas 2D L_{Dice} , un problème technique

entraînant l'arrêt prématuré des entraînements (tâches préemptées par un utilisateur prioritaire) a largement réduit le nombre de répétitions, nous reviendrons sur ce problème dans la conclusion. Pour le cas 2D, le modèle est un U-Net avec 64 filtres dès la première couche et dans le cas 3D un U-Net avec 32 filtres en première couche.

La figure 4.15 représente le score pour des modèles 2D et la figure 4.16 pour des modèles 3D.

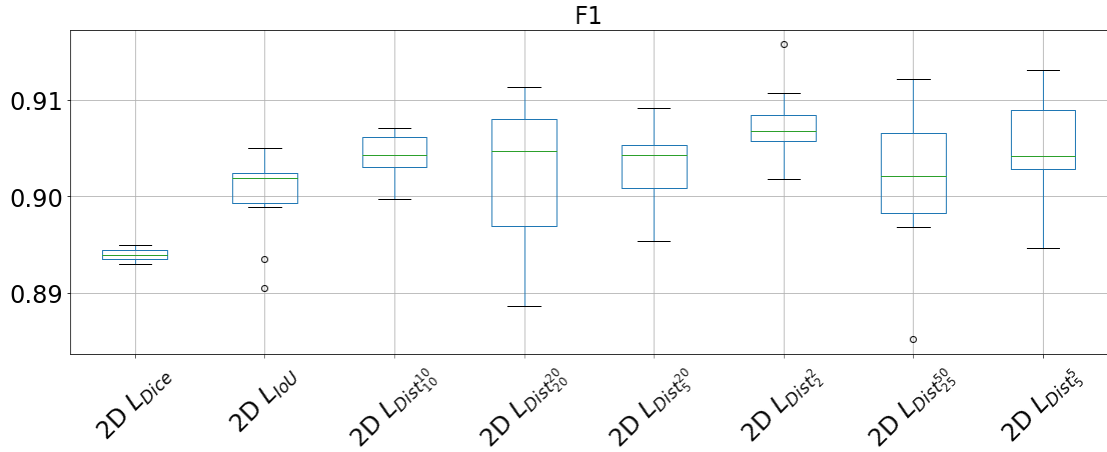


FIGURE 4.15 – Résultats de l'expérience de choix de loss en 2D, F1 Score.

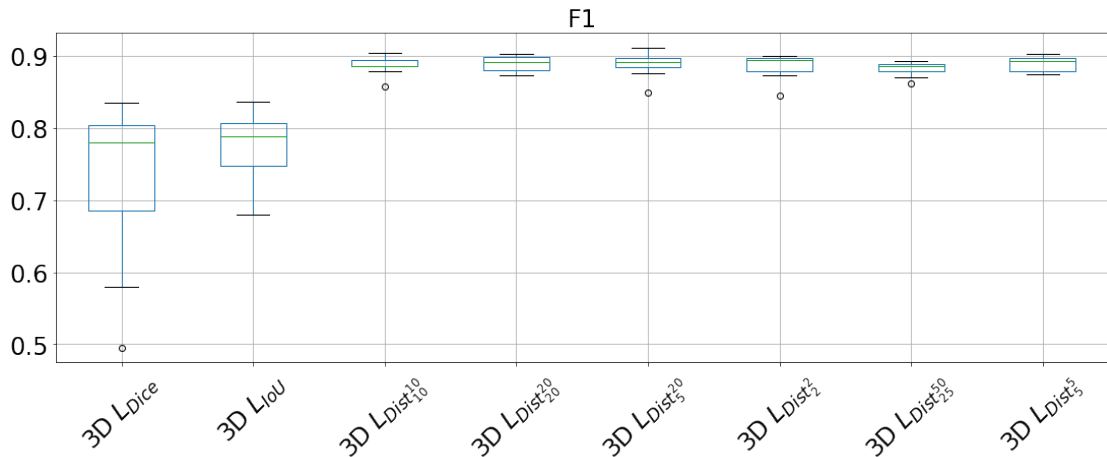


FIGURE 4.16 – Résultats de l'expérience de choix de loss en 3D, F1 Score.

Cette expérience nous permet de conclure que la fonction de perte L_{Dist} permet d'entraîner de manière plus efficace des modèles de segmentation pour la classe des mitochondries. L'impact du paramètre d'échelle est peu important ; les résultats montrent une différence faible. Il serait hasardeux de tirer une conclusion dans le cas 2D L_{Dice} , même si l'intersection sur l'union semble donner de meilleurs résultats, la figure ne représente

pour ce cas que deux entraînements. Dans le cas 3D, la différence de score médian entre L_{Dice} et L_{IoU} est négligeable.

Expérience 2

Nous avons vu que pour les mitochondries, la fonction L_{Dist} est le meilleur choix. Dans cette sous-section, nous reprenons les résultats de l'expérience de la sous-section 4.2.2 afin d'observer les résultats sur d'autres classes.

La figure 4.17 représente individuellement chaque résultat d'entraînement pour les 4 classes du jeu de données.

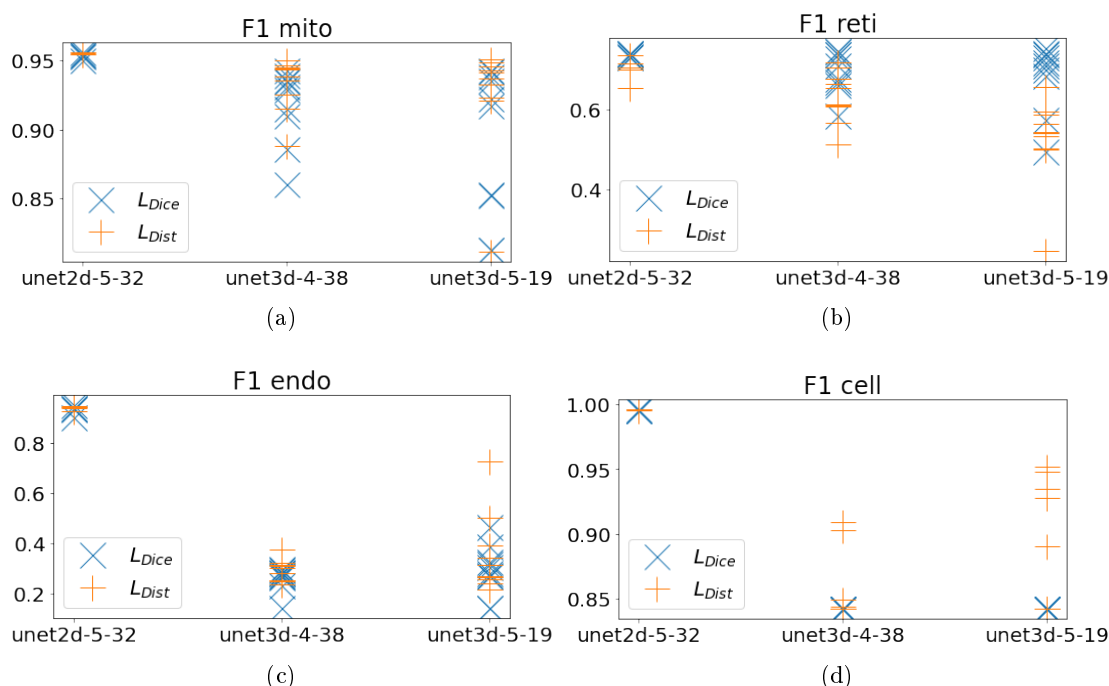


FIGURE 4.17 – Comparaisons des fonctions de pertes pour modèles 2D et 3D sur 4 classes.

Si pour le cas des mitochondries, la conclusion précédente n'est pas modifiée, les résultats sur une autre classe est quant à elle inversée. Dans le cas du réticulum endoplasmique, la fonction L_{Dice} donne de bien meilleurs résultats, en particulier pour les modèles 3D, mais également en 2D. C'est un résultat important pour nous, car le réticulum endoplasmique fait partie des classes pour lesquelles la segmentation est difficile, et où l'amélioration est la plus importante. S'il n'est pas inutile d'améliorer notre score de quelques points pour une classe déjà bien segmentée, les améliorations sur des classes plus complexes restent une priorité afin d'atteindre un seuil de « la bonne segmentation visuelle ».

Nos expériences montrent des résultats contradictoires ne permettant pas une conclusion sans équivoque. L'information la plus importante que l'on peut réellement tirer de

cette sous-section c'est que même en essayant au maximum de rendre nos expériences indépendantes, certains paramètres comme le jeu de données ou la classe visée peuvent avoir un impact important sur nos résultats.

Nous avons décidé d'utiliser de manière régulière la fonction de perte L_{Dist} qui donne de meilleurs résultats sur la plupart des classes. Visuellement, les prédictions des modèles produisant une distance aux contours sont moins bruitées.

4.2.7 Modèle pré-entraîné

Dans cette sous-section, nous évaluons l'utilisation de modèles pré-entraînés en les comparant à des modèles initialisés aléatoirement.

Par défaut, lorsque l'on utilise TensorFlow, les biais sont initialisés à 0 et les poids des noyaux de convolutions (W_{ij}) sont initialisés à l'aide d'un initialiseur de Glorot (aussi appelé initialiseur de Xavier) [Glorot and Bengio, 2010].

$$W_{ij} \sim U \left[-\frac{\sqrt{6}}{\sqrt{fan_{in} + fan_{out}}}, \frac{\sqrt{6}}{\sqrt{fan_{in} + fan_{out}}} \right]$$

où U est une distribution uniforme, fan_{in} est la taille de la couche précédente et fan_{out} est la taille de la couche courante.

Une alternative à cette initialisation peut être l'utilisation de modèle pré-entraîné sur une tâche de traitement d'image. C'est ce que propose par exemple Yakubovskiy avec la bibliothèque logicielle « Segmentation Models » [Yakubovskiy, 2019], utiliser des modèles dont l'encodeur est composé d'un CNN pré-entraîné sur ImageNet [Deng et al., 2009]. Dans cette sous-section, nous utilisons cette bibliothèque logicielle et ces modèles dont l'encodeur est pré-entraîné.

Expérience

Dans cette expérience, nous comparons 3 modèles, un modèle U-Net et deux modèles U-Net utilisant un encodeur EfficientNet, l'un deux ayant été entraîné sur la tâche de classification du jeu de données ImageNet. Ces résultats sont issus d'expériences valorisées dans la communication [Meyer et al., 2021].

La figure 4.18 représente les résultats de segmentation des différents modèles.

Dans les cas que nous avons étudiés, nous n'avons pas pu montrer de différences significatives entre l'utilisation d'un modèle pré-entraîné ou non pré-entraîné.

4.2.8 Adaptation de domaine et CycleGAN

Si les résultats de prédiction sont satisfaisants lorsque l'on entraîne un modèle sur une ou plusieurs images, la généralisation entre différents domaines ne l'est pas du tout. Par exemple, dans le cas où l'on entraîne un modèle uniquement sur quelques coupes de l'image I_3 , les prédictions sur le reste de l'image sont satisfaisantes, même si le processus d'imagerie crée ces images de manière indépendante. Cependant, si l'on utilise ce modèle sur une image dépendant de paramètres de préparation différents comme LW_4-1 , les

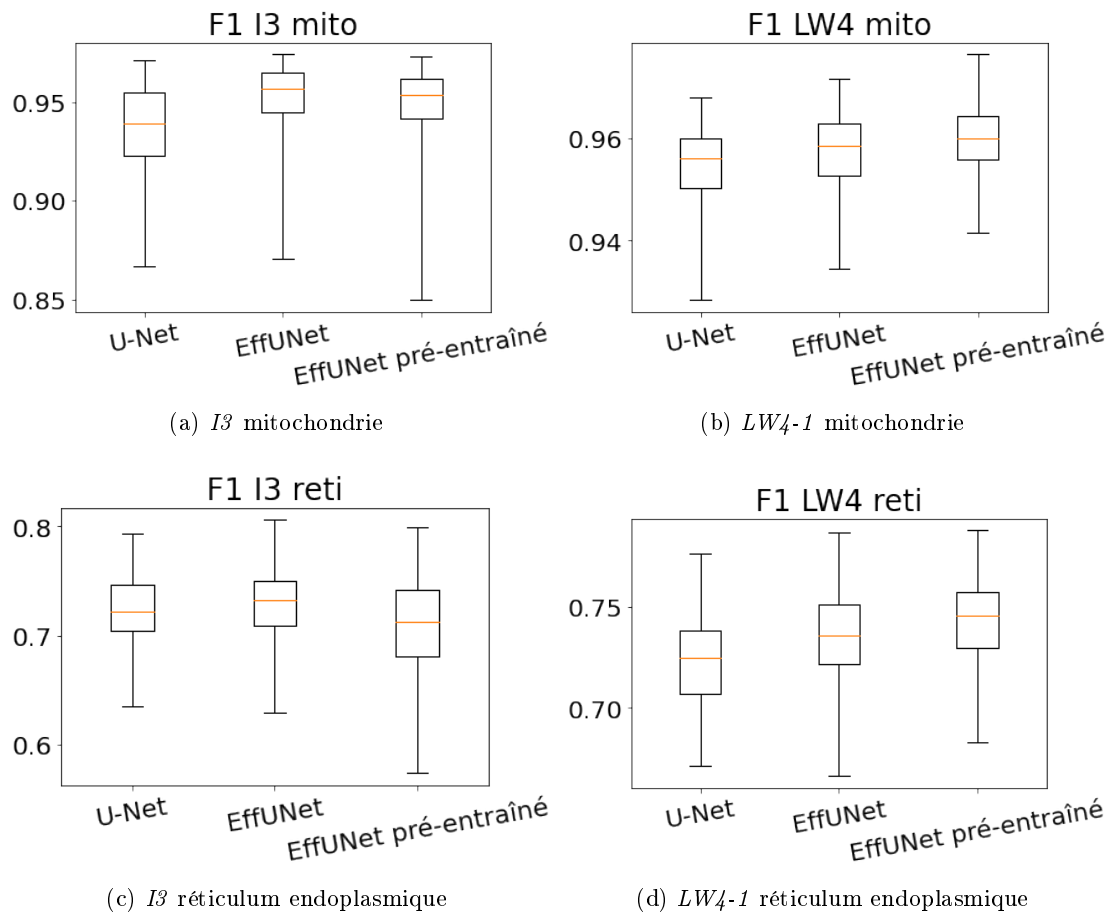


FIGURE 4.18 – Résultats de l'expérience sur les modèles pré-entraînés.

résultats sont quasi nuls. Dans les pires cas, aucun élément n'est segmenté, dans les meilleurs cas, la segmentation générée n'est pas exploitable, seuls quelques éléments étant détectés. La dégradation des résultats est d'autant plus importante lorsque les domaines s'éloignent, c'est-à-dire que les paramètres différents sont plus nombreux.

Il est possible d'utiliser deux images en même temps pendant l'entraînement, mais les résultats sur une troisième image différente sont tout de même insuffisants ; le modèle n'est pas capable de généraliser correctement vers les autres domaines. En réalité, lorsque l'on utilise plusieurs images lors de l'apprentissage, les prédictions du modèle sont très proches des résultats fournis par deux modèles entraînés indépendamment.

Pour tenter de résoudre ce problème, nous avons expérimenté l'utilisation d'un CycleGAN [Zhu et al., 2017] pour transformer les images d'un domaine vers l'autre. Cette approche a déjà été appliquée avec succès dans le domaine biomédical par les auteurs de [Armanious et al., 2019]. Un CycleGAN est composé de quatre modèles (architecture simplifiée, figure 4.19). D'un côté, on a un générateur qui prend en entrée une image du

domaine A et qui apprend à la transformer dans le domaine B et un discriminant qui apprend à reconnaître les images du domaine A. De l'autre côté, on a un générateur qui prend en entrée une image du domaine B et qui apprend à la transformer dans le domaine A, et un discriminant qui apprend à reconnaître les images du domaine B. Une image peut ainsi passer du domaine A au domaine B, puis à nouveau dans le domaine A.

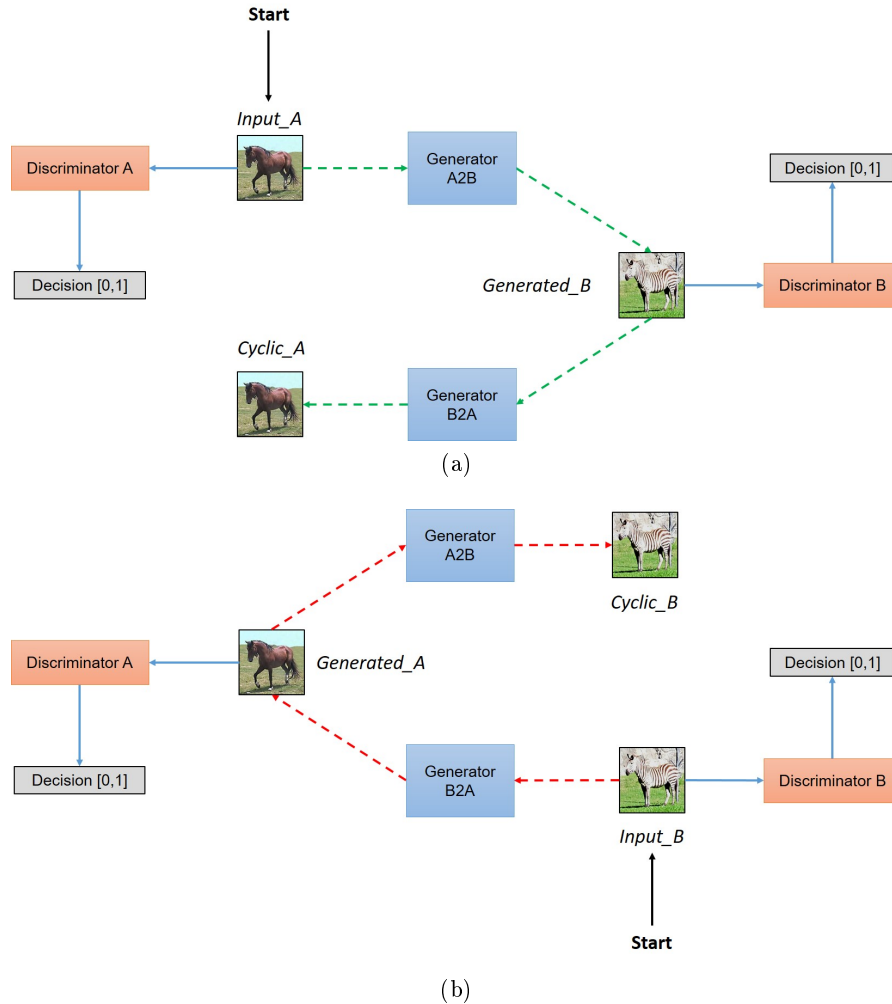


FIGURE 4.19 – Architecture de la méthode CycleGAN [Zhu et al., 2017].

Le CycleGAN permet donc d'apprendre le passage d'un domaine à l'autre de manière non supervisée. C'est un avantage non négligeable quand on connaît la difficulté de création de nouvelles annotations. La figure 4.20 représente des résultats d'utilisation de CycleGAN pour l'adaptation de domaine, ainsi que les résultats d'une segmentation sur ces images.

On peut remarquer que dans le cas où l'on transforme l'image *LW4-1* dans le domaine de l'image *I3*, les résultats de segmentation sur l'image transformée par un modèle

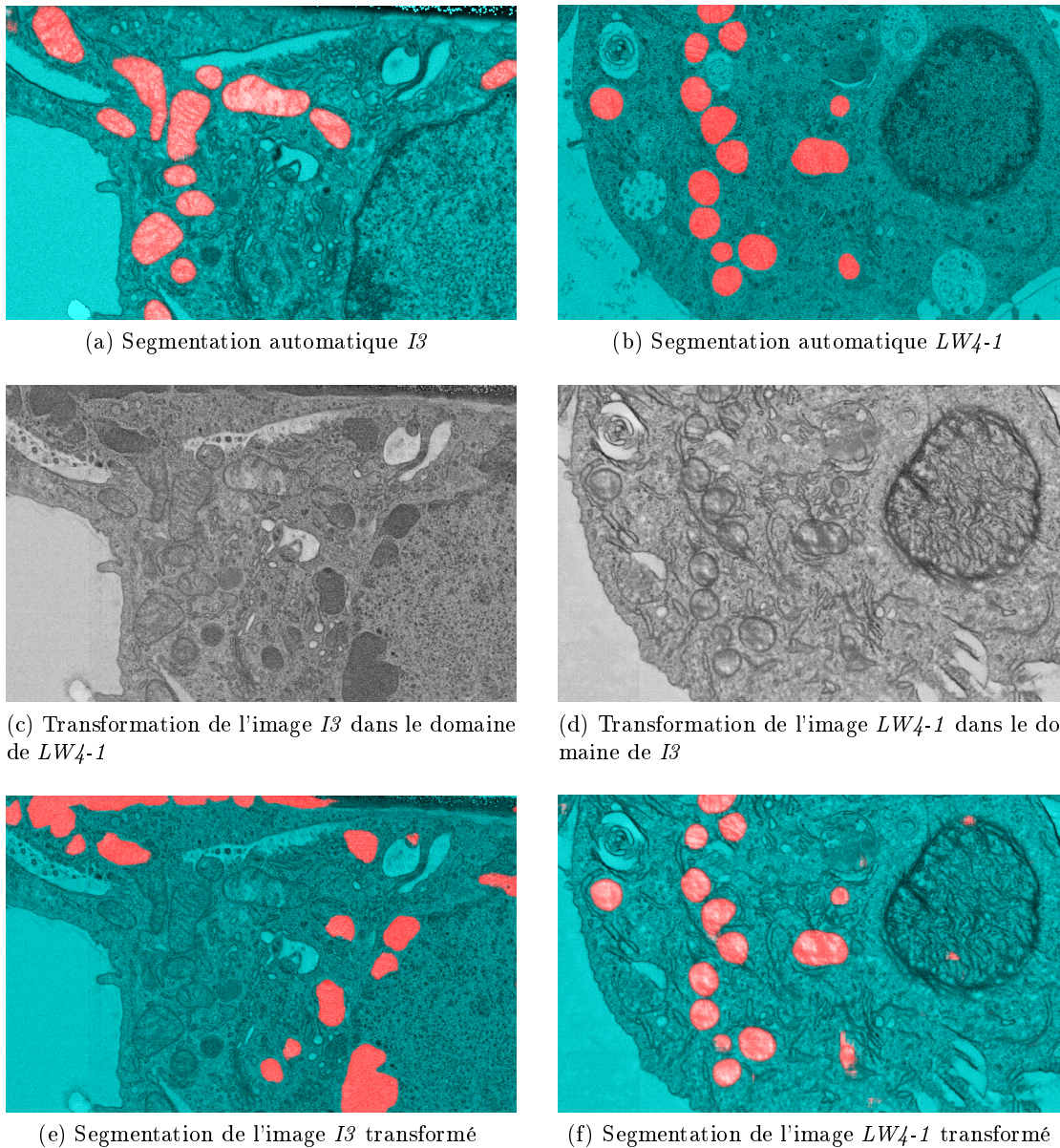


FIGURE 4.20 – Expérimentation CycleGAN.

entraîné sur l'image $I3$ uniquement sont qualitativement bons. Dans le sens inverse, les résultats de la transformation ne permettent pas la détection des objets. Visuellement, on remarque que les mitochondries ne sont pas, ou très peu transformées. On peut supposer que pour transformer une mitochondrie d'un domaine vers un autre, et non en noyau, il faudrait que le modèle intègre l'existence de ces objets. Ce n'est pas le cas dans la méthode présentée, qui est complètement non supervisée. Une solution pourrait être de

contraindre les modèles à l'aide d'une faible quantité d'annotations ou de connaissances a priori.

D'après l'état actuel des connaissances, le problème de mauvaise adaptation dans le sens $I\mathcal{B} \rightarrow LW4-1$ mais de bonne adaptation dans le sens $LW4-1 \rightarrow I\mathcal{B}$ n'est pas uniquement présent avec l'utilisation de CycleGAN, mais également avec d'autres méthodes d'adaptation de domaine [Stenger et al., 2023].

4.2.9 Premières conclusions

Dans cette première section, nous avons comparé de nombreuses variations dans les méthodes d'entraînement de nos modèles. L'objectif initial était de mettre en avant les améliorations possibles. Les expériences ont également permis de remarquer quelles modifications avaient un impact réel, négligeable ou instable.

Grâce à ces expériences, nous avons pu conclure sur les choix à faire afin d'entraîner au mieux les modèles dans notre contexte.

- Les modèles multi-classes sont à privilégier pour notre cas d'application où nous cherchons à développer un modèle le plus général pour nos images.
- Les modèles 2D sont plus pertinents que les modèles 3D.
- L'utilisation de connexions résiduelles sur les blocs de traitement du modèle U-Net permet une amélioration significative pour un coût très faible.
- Les patchs utilisés pour l'entraînement doivent avoir une taille minimale de 128×128 avec une préférence pour les tailles de patchs plus grandes.
- Un jeu d'entraînement composé de suffisamment de coupes annotées est nécessaire et la quantité d'annotation requise dépend également de la qualité des données présentes dans les images.
- Le choix de la fonction de perte doit prendre en compte les classes à segmenter, mais l'impact entre L_{Dice} et L_{Dist} est à relativiser. Nous préférons tout de même la fonction L_{Dist} donnant de meilleurs résultats dans la plupart des cas.
- L'utilisation de modèle pré-entraîné n'est pas nécessaire si l'on n'applique pas de méthode plus avancée d'adaptation de domaine, de transfer-learning ou de meta-learning.

Nous avons également étudié une méthode d'adaptation de domaine non supervisée utilisant un CycleGAN afin de tenter d'améliorer la robustesse de nos modèles aux changements de domaine d'image. La méthode n'est pas efficace pour répondre à notre problème de robustesse entre des images de différents domaines.

Suite à ces résultats, nous avons identifié deux problématiques intéressantes pour lesquelles plusieurs pistes de recherche pouvaient être ouvertes. La première concerne l'absence de généralisation entre les modèles entraînés sur des images de différents domaines. La seconde concerne la difficulté à segmenter des objets plus petits et moins contrastés comme le réticulum endoplasmique. Nous présentons ces pistes dans la suite de ce chapitre.

4.3 Segmentation interactive et annotations faibles

Comme nous venons de le voir dans la section précédente, la méthode de segmentation ne peut être appliquée sans disposer de segmentations de référence pour l'image courante à traiter. Le problème réside dans le fait que la création de ces segmentations de référence est une tâche longue et fastidieuse. De plus, lorsqu'un expert segmente une image, la décision est parfois difficile à prendre pour certains objets. Une solution est de proposer une méthode prenant moins de temps et étant moins contraignante pour les utilisateurs. Dans cette section, nous présentons différentes méthodes utilisant segmentation interactive et annotations faibles dans le but de réduire le temps et la difficulté d'annotation.

Commençons par définir les deux termes, segmentation interactive et annotations faibles, leurs points communs et différences.

La **segmentation interactive** regroupe toutes les méthodes utilisant des interactions dynamiques avec utilisateurs afin de segmenter une image. L'utilisateur manipule le modèle de manière interactive en donnant des indications durant le processus de segmentation. Les méthodes utilisant des **annotations faibles** regroupent toutes les techniques utilisant des annotations dont la qualité est réduite et dont la réduction de qualité est prise en compte. Plutôt que d'entraîner un modèle avec une segmentation de référence complète, prenant en compte tous les objets que l'on souhaite segmenter, on entraîne un modèle avec des données incomplètes. En pratique, cela revient à ne pas colorier complètement les éléments de l'image, mais seulement une partie.

Dans les deux cas, l'utilisateur va segmenter des images à l'aide d'informations faibles : des points positifs et négatifs, des marqueurs manuels ou encore des boîtes englobantes. Avec la méthode interactive, l'utilisateur intervient durant la phase de prédiction. Le modèle interactif est entraîné avec des segmentations de références et des interactions simulées. Avec les annotations faibles, le modèle est entraîné avec des données produites en amont de moins bonne qualité. Cela signifie par exemple que seulement une partie des objets seront segmentés.

Nous avons commencé à travailler sur la segmentation interactive et c'est à la suite de ces travaux que le rapprochement avec les annotations faibles a été fait et que les expériences de comparaisons ont été menées.

4.3.1 Segmentation interactive et simulation d'interaction

Dans cette première section, nous nous intéressons tout d'abord aux modèles interactifs. Nos travaux sont largement inspirés des travaux de Xu et al. [Xu et al., 2016].

Notre modèle interactif est semblable à un modèle de segmentation classique. La seule différence réside dans les données utilisées en entrée du modèle. Dans le cas non interactif, un modèle multi-classe prend en entrée une image de dimensions $x \times y \times 1$ et produit en sortie une carte de probabilité de dimensions $x \times y \times |C|$. Dans le cas interactif, on ajoute des canaux à l'image d'entrée représentant les cartes d'interactions. Il peut y avoir une carte unique, si l'on souhaite utiliser les contours comme marqueurs ou plusieurs cartes dans le cas où l'on souhaite une carte positive et une négative. Le

modèle est alors entraîné à fournir une segmentation de référence à l'aide des entrées combinant image et interactions.

L'objectif étant de rendre l'annotation manuelle plus rapide et aisée, les interactions sont alors simulées à partir des segmentations de références fournies.

Une première solution pourrait être de développer un modèle interactif général permettant la séparation d'objets individuellement comme dans l'article de Xu et al., mais ce n'est pas la méthode que nous avons retenue en premier lieu. Nous avons retenu une approche plus simple à mettre en place techniquement, mais également plus proche de notre objectif de segmentation sémantique. Nos annotations sont des segmentations de référence sans séparations des instances, ce qui signifie que deux objets qui se touchent forment une seule composante connexe.

Notre objectif est de segmenter une image de la manière suivante :

- l'utilisateur crée une segmentation de référence dense R_1 en quelques minutes ;
- on entraîne automatiquement un modèle interactif I avec cette segmentation de référence R_1 , en simulant les interactions ;
- l'utilisateur crée une segmentation de référence plus conséquente R_2 à l'aide du modèle interactif I qui lui permet de le faire plus rapidement ;
- on entraîne un modèle automatique A avec les segmentations de références R_1 et R_2 ;
- on produit automatiquement une segmentation pour toute l'image à l'aide du modèle A .

Cette méthode peut également être utilisée de manière itérative, jusqu'à l'obtention d'une segmentation satisfaisante. La figure 4.21 représente schématiquement ce processus itératif.

Afin de comparer les différentes méthodes, nous devons simuler des interactions utilisateurs. Ces interactions sont simulées à partir des annotations denses utilisées pour l'entraînement. Deux types d'interactions ont été sélectionnés.

Un premier type, appelé « DEXTR » (figure 4.22b) inspiré par l'article [Maninis et al., 2018] et utilisant les points extrêmes des objets à la manière de boîtes englobantes. Ces interactions ont un désavantage, le nombre constant d'interactions pour des objets de dimensions différentes, mais la simulation d'interaction est triviale.

Un second type, appelé « Fill » (figure 4.22c) basé sur la sélection de points par l'utilisateur, consistant à remplir l'intérieur des objets avec des interactions positives et le voisinage des objets par des interactions négatives. Le nombre d'interactions dépend d'un paramètre de densité, on ajoute des points aléatoirement dans la figure ayant une distance minimum entre eux, jusqu'à ce qu'il ne soit plus possible d'en ajouter. Nous avons expérimenté différents paramètres de densités qui seront détaillés dans la suite.

Dans les deux cas, les interactions sont transformées en carte d'interactions à l'aide de transformations de distance dont la valeur est restreinte par un paramètre de limite, fixé à 50.

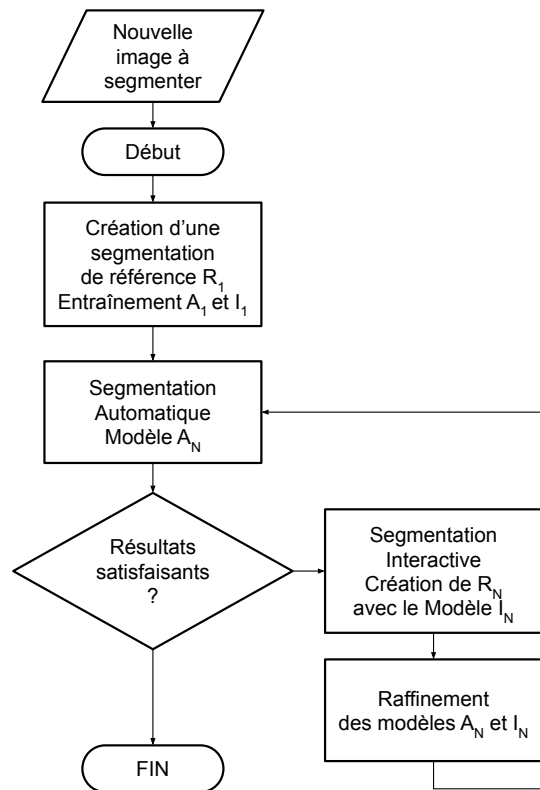


FIGURE 4.21 – Représentation schématique du processus itératif utilisant la segmentation interactive.

La figure 4.22 représente les différents types d'interactions proposés. On remarque dans la sous-figure 4.22b représentant les interactions extrêmes que la mitochondrie centrale allongée ne possède que peu d'annotations.

Notre expérience consiste à comparer 3 cas, le cas « Auto », où aucune interaction n'est fournie, le cas « DEXTR », où une carte d'interaction représentant les points extrêmes est fournie et le cas « Fill », où deux cartes d'interaction sont fournies. Les résultats suivants sont issus d'essais ayant été répétés 10 fois, le jeu de données utilisé est une variation issue de l'image I_3 et seules les mitochondries sont concernées. Le paramètre dans cette expérience correspond au nombre de coupes denses ayant été utilisées pour entraîner les modèles interactifs. Les figures 4.23 et 4.24 représentent les résultats de cette expérience sur le jeu de test en fonction du nombre de coupes utilisées pour l'entraînement.

Sur la figure 4.23, on remarque que même avec une seule coupe utilisée pour entraîner le modèle interactif (Fill 0001), on arrive à atteindre un score similaire à un modèle automatique utilisant 8 coupes.

Dans notre contexte, c'est l'approche utilisant peu de coupes denses qui nous intéresse, mais la figure 4.24 nous permet de remarquer que même en augmentant le nombre

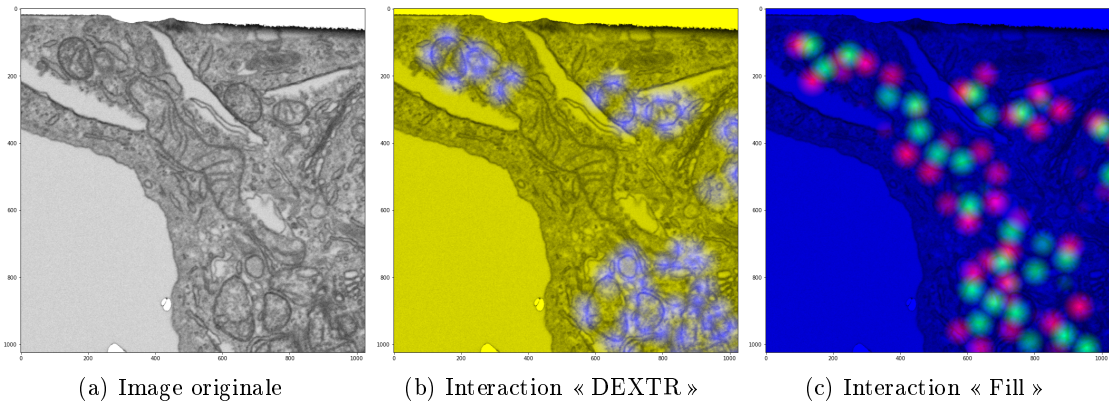


FIGURE 4.22 – Différents types d’interactions pour la classe des mitochondries. Dans la sous-figure 4.22b, les points bleus correspondent aux 4 points extrêmes des composantes connexes. Dans la sous-figure 4.22c, les points verts correspondent aux interactions positives et les points rouges aux interactions négatives.

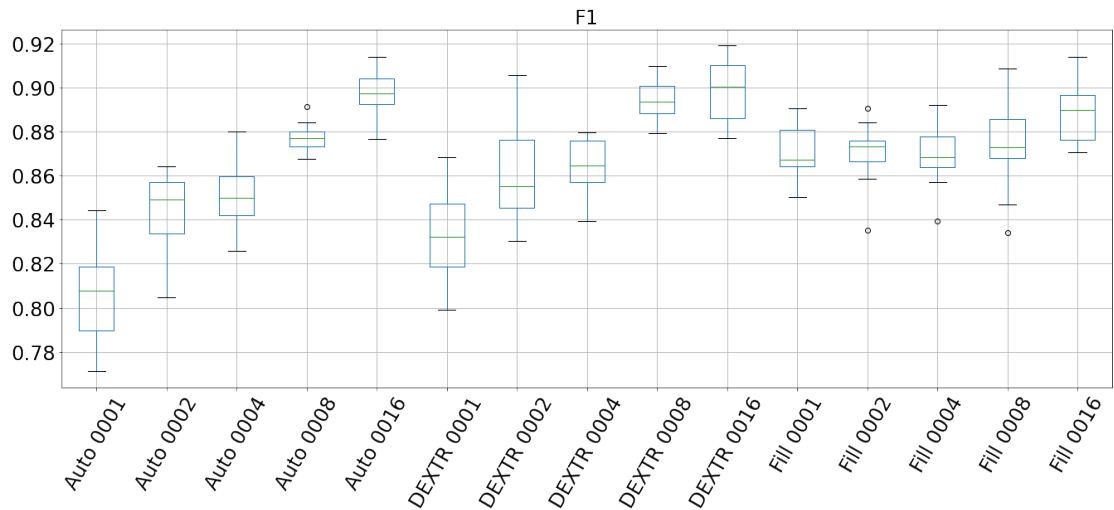


FIGURE 4.23 – Résultat de l’expérience de segmentation interactive sur l’image I_3 avec un nombre de coupes d’entraînement compris entre 1 et 16.

de coupes, les interactions extrêmes sont rarement plus efficaces. Ce phénomène est probablement lié au fait que nos objets sont de tailles très variables, et que les annotations en question ne s’y adaptent pas correctement.

Lorsque l’on utilise une seule coupe pour l’entraînement, la méthode d’annotation Fill- β permet une amélioration d’environ 10% par rapport aux annotations DEXTR et de 45% pour la méthode automatique. Dans le cas utilisant les 384 coupes, l’amélioration des méthodes d’annotations Fill- β et DEXTR sont d’environ 20% par rapport à la

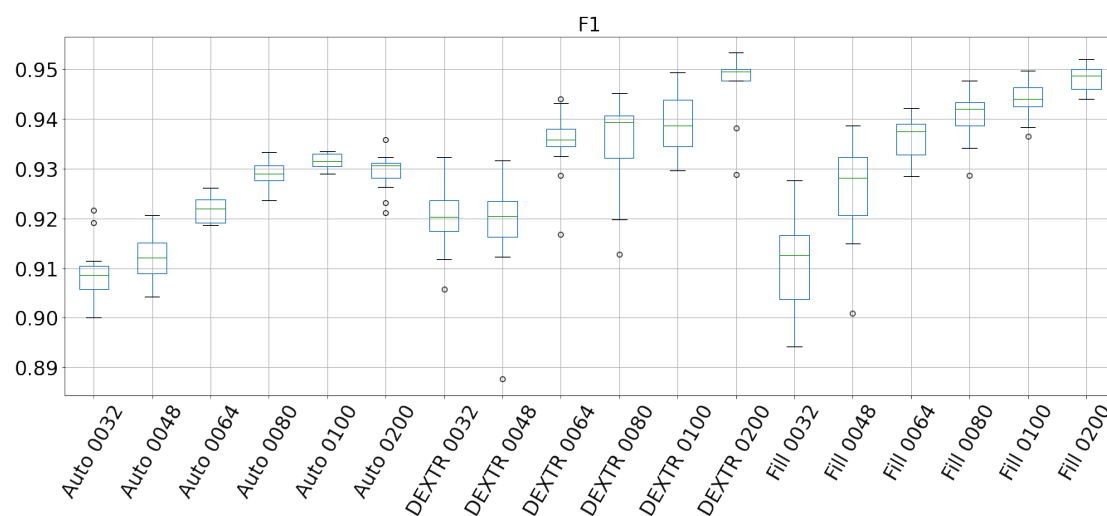


FIGURE 4.24 – Résultat de l’expérience de segmentation interactive sur l’image $I3$ avec un nombre de coupes d’entraînement compris entre 32 et 200.

méthode automatique. On peut donc conclure que l’ajout d’informations issues d’interactions simulées permet en effet d’améliorer nettement les performances de nos modèles et que les interactions simulées de remplissage sont plus efficaces dans notre contexte.

Dans la pratique, le jeu de données que nous avons utilisé ne pose pas réellement de problème. On arrive très facilement à atteindre un F1-Score de 90% avec 16 coupes denses, ce qui est plutôt satisfaisant. Néanmoins, nous avons également travaillé sur une image qui pose beaucoup plus de problèmes, l’image $JG1$. Cette image est beaucoup plus bruitée et les objets d’intérêt moins contrastés. De plus, on ne possède que 5 blocs de $384 \times 384 \times 384$ pixels pour entraîner et évaluer notre méthode. On peut remarquer ce manque de contraste dans les exemples de coupes présentés dans la figure 4.25.

Nous avons donc répété l’expérience sur ces nouvelles données, mais cette fois avec deux paramètres de densité pour la méthode de simulation de remplissage. Une première version « Fill- α » correspond aux paramètres déjà utilisés précédemment, et une seconde version « Fill- β » plus dense. Le paramètre de distance minimum pour l’ajout d’une interaction passe de 75 à 30, pour l’intérieur de l’objet, et de 80 à 50, pour l’extérieur. De manière plus simple à évaluer, le nombre d’interactions sur le jeu de validation passe d’environ 40 à 140, et sur le jeu de test de 53 à 158, ce qui représente moins d’une interaction par coupe dans le cas le plus dense.

De manière similaire aux résultats de l’expérience présentés dans la sous-section 4.2.5, le paramètre de l’expérience représente le nombre de coupes utilisées par bloc. Les figures 4.26, 4.27, 4.28 et 4.29 représentent les résultats de cette expérience.

Tout d’abord, on remarque que même avec la totalité des coupes d’entraînement, le modèle automatique peine à atteindre un score de 70%. Même avec une seule coupe d’entraînement, le modèle interactif « Fill- β » surpasse déjà largement le modèle auto-

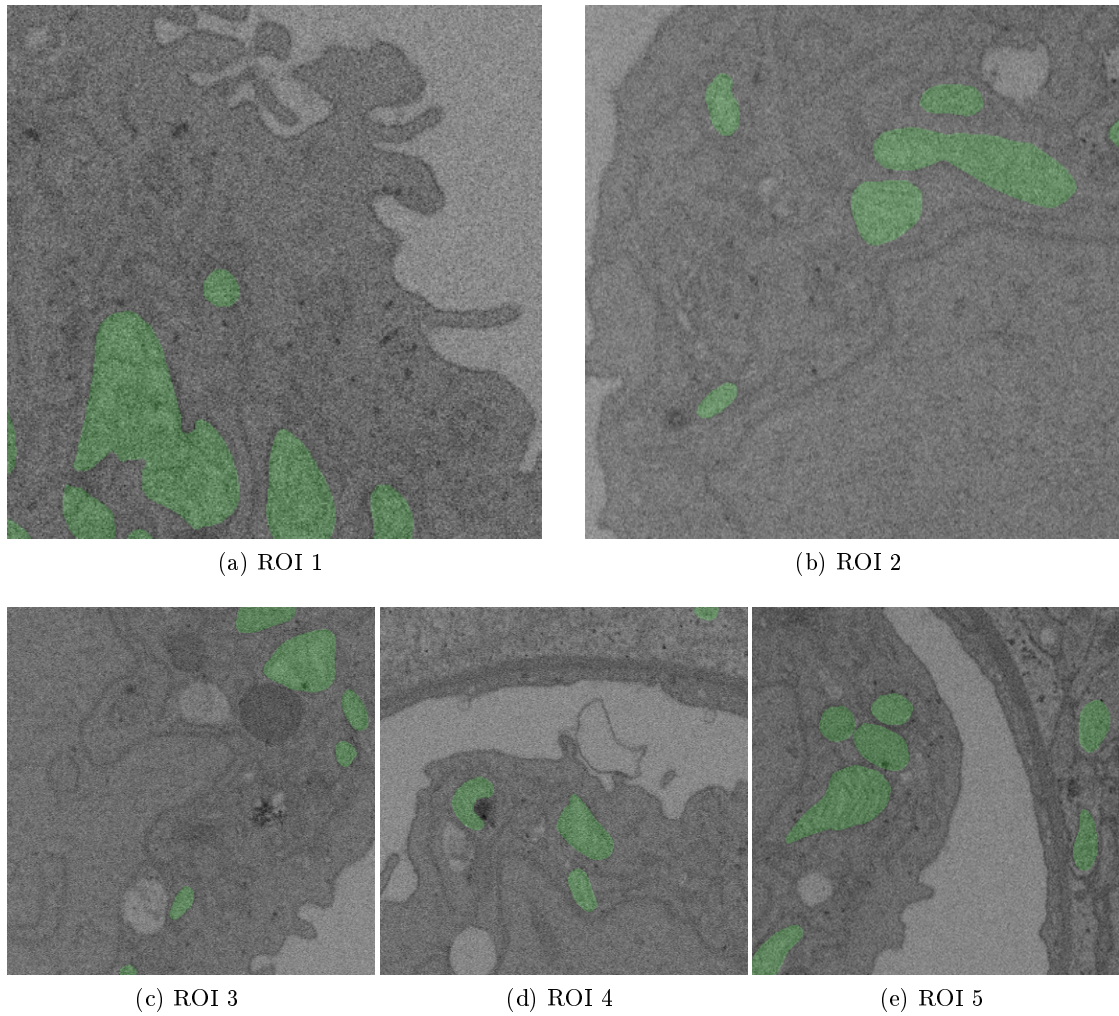


FIGURE 4.25 – Exemples de coupes des blocs d’entraînement et de test pour l’image *JG1*.

matique. Dans ces images, la répartition et la taille des mitochondries sont plus stables, leurs tailles ne varient pas énormément et ce sont toujours des boules plus ou moins déformées, de taille réduite. La méthode d’interaction extrême fonctionne très bien dans ce cas, mais sans surpasser la méthode de remplissage plus dense.

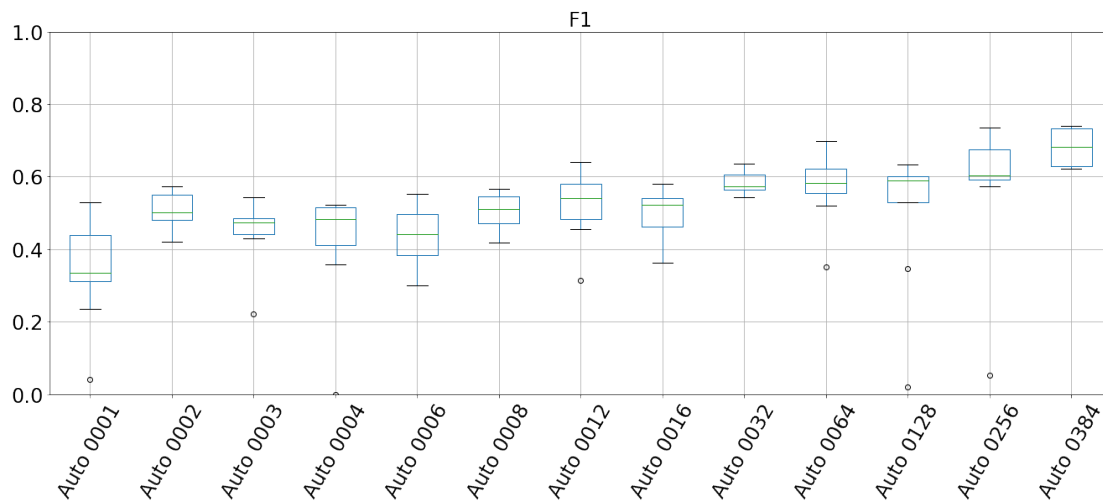


FIGURE 4.26 – Résultats de l'expérience de segmentation interactive sur l'image *JG1*, modèle automatique.

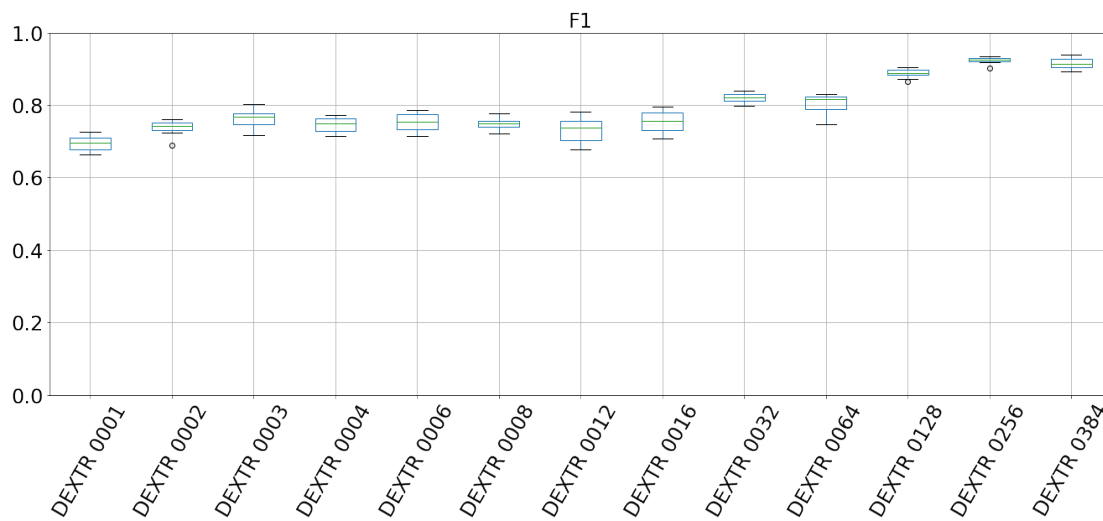


FIGURE 4.27 – Résultats de l'expérience de segmentation interactive sur l'image *JG1*, modèle interactif, interactions « DEXTR ».

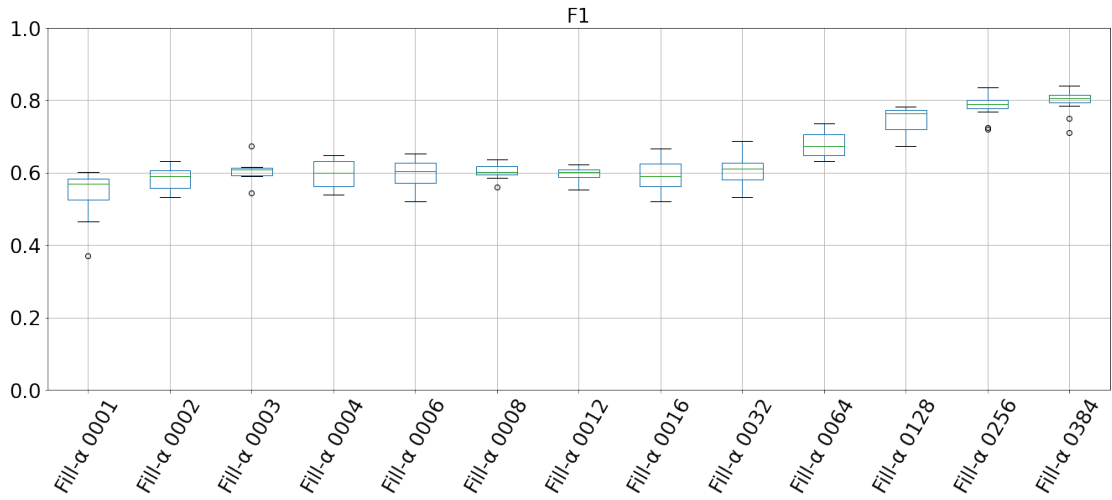


FIGURE 4.28 – Résultats de l'expérience de segmentation interactive sur l'image *JG1*, modèle interactif, interactions « Fill ».

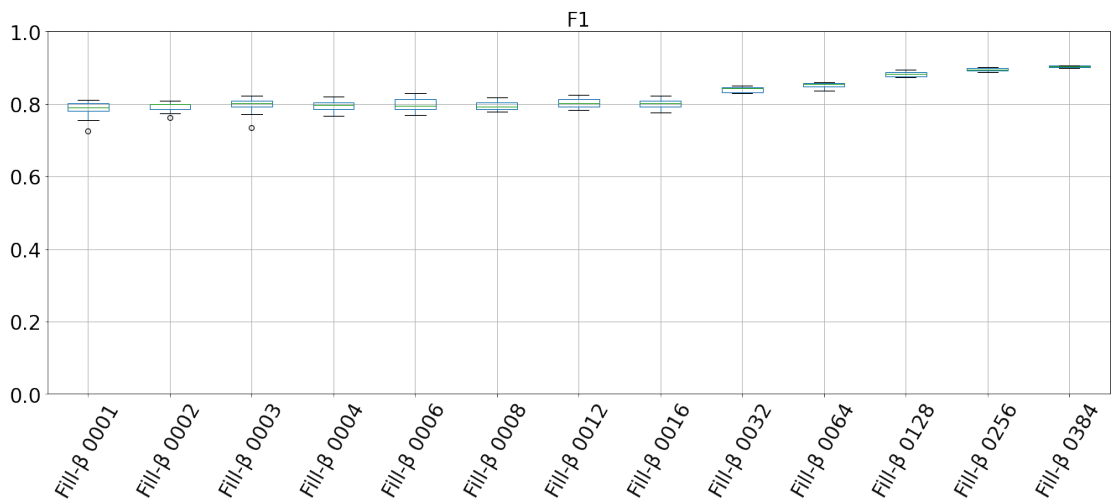


FIGURE 4.29 – Résultats de l'expérience de segmentation interactive sur l'image *JG1*, modèle interactif, interactions « Fill » plus dense.

Suite à ces résultats, nous avons pu préparer notre expérience complète qui consiste à entraîner un modèle automatique à partir de segmentation interactive. L'expérience consiste en plusieurs étapes :

1. entraînement d'un modèle interactif sur une coupe dense ;
2. génération de segmentations denses avec le modèle interactif et des interactions simulées ;
3. entraînement d'un modèle automatique avec les coupes segmentées interactivement.

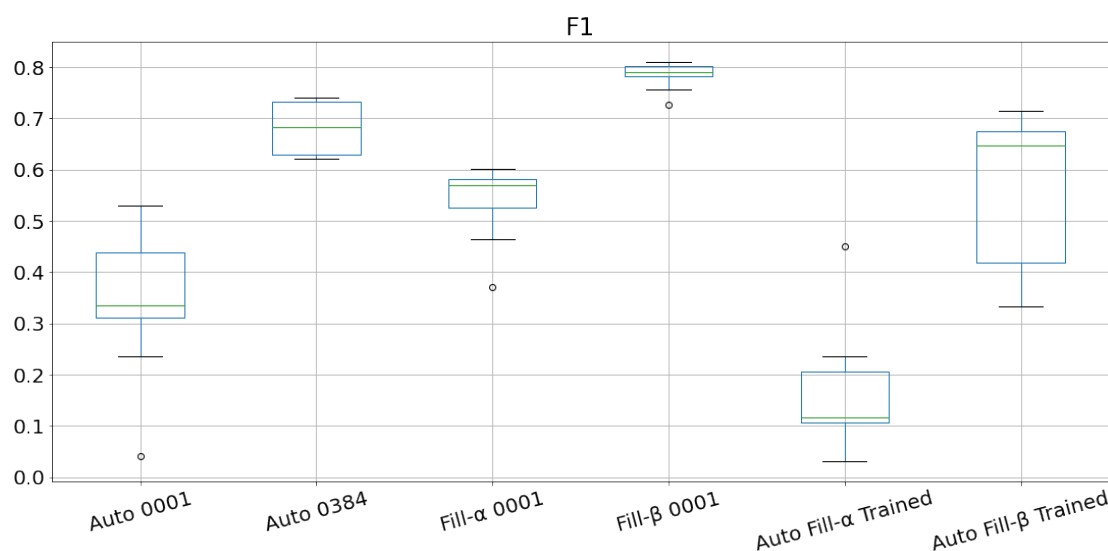


FIGURE 4.30 – Comparaison entre deux modèles automatiques, deux modèles interactifs et deux modèles automatiques entraînés sur les résultats des modèles interactifs.

Nous avons comparé les résultats de ces apprentissages pour les deux différentes méthodes d'interaction de remplissage. La figure 4.30 représente les résultats de cette expérience. Les cas « Auto Fill- $[\alpha|\beta]$ Trained » correspondent aux modèles automatiques entraînés à partir de segmentation interactive.

Premièrement, on remarque que le modèle entraîné à partir des interactions moins dense (Auto Fill- α Trained) est moins performant qu'un modèle automatique entraîné sur une seule coupe (Auto 0001). Le modèle ayant généré les coupes d'entraînement (Fill- α 0001) de ce modèle est plus performant que le modèle automatique. Cela nous indique que la qualité des segmentations interactives doit être suffisante, dans notre cas, au moins supérieure à 60%, afin d'entraîner correctement un modèle.

D'un autre côté, pour le modèle entraîné à partir d'interactions plus denses (Auto Fill- β Trained), les résultats sont prometteurs. Le score du 3e quartile est similaire au score médian du modèle automatique entraîné sur toutes les coupes (Auto 0384). Avec des segmentations interactives atteignant un score d'environ 80%, on arrive à entraîner correctement un modèle.

Rappelons que dans notre expérience, toutes les interactions sont simulées, donc il n’y a pas réellement de gain de temps, mais on peut espérer réduire le temps d’annotation de manière significative (au moins deux fois plus rapide). Les gains sont réalisés en termes de temps humain, plutôt qu’en temps absolu, en raison de l’entraînement d’un modèle intermédiaire

4.3.2 Interactions via interface homme machine

Afin de mettre en pratique notre méthode de segmentation interactive, nous avons développé une démonstration logicielle appelée « cutter ». Ce logiciel permet de segmenter interactivement les images à l’aide d’un modèle préalablement entraîné. La figure 4.31 représente le résultat de cette segmentation. Les images en bas à droite représentent les cartes d’interactions positives et négatives. Le canal de couleur rouge représente le résultat de la sortie du modèle de segmentation.

Ce logiciel est encore à l’état de prototype, instable et non optimisé, en particulier pour ce qui est de la segmentation interactive. Nous détaillerons d’autres utilisations de ce logiciel dans la section A.1.

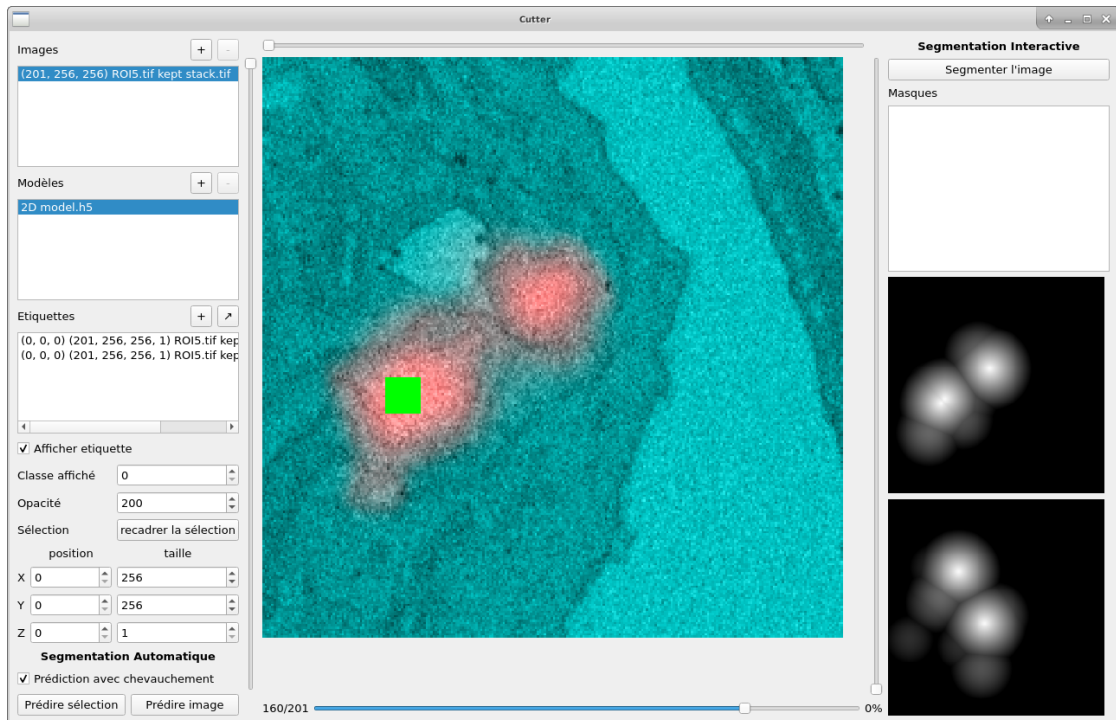


FIGURE 4.31 – Démonstration de l’outil de segmentation cutter. Cas d’utilisation pour la segmentation interactive sur l’image *JG1*.

4.3.3 Annotations faibles

Après avoir effectué ces expériences de segmentation interactive et travaillé sur la simulation d'interactions, nous avons conclu que bien qu'intéressante, l'utilisation d'un modèle intermédiaire est la principale contrainte pour faire adopter la méthode. Cette étape de création d'un premier modèle interactif ne permet pas de produire une segmentation en une seule étape pour l'utilisateur.

Une autre solution serait d'utiliser des annotations faibles, c'est-à-dire des annotations qui ne sont pas complètes. Ces travaux sont largement inspirés des travaux de Çiçek et al. qui dans leur article [Çiçek et al., 2016] ne proposent pas uniquement l'utilisation d'un modèle de U-Net 3D, mais également de « sparse annotation », des annotations faibles.

Les annotations faibles, tout comme les interactions précédemment évoquées sont beaucoup plus rapides à produire, mais pour l'utilisateur, il n'y a aucun changement en dehors de la phase d'annotation. Dans le cas des annotations denses (figure 4.32a), l'utilisateur annote les objets en les « coloriant ». La classe représentant « le fond » est créée par dualité avec les annotations. Dans le cas des annotations faibles (figure 4.32b), il est nécessaire d'annoter le fond avec une classe spécifique.

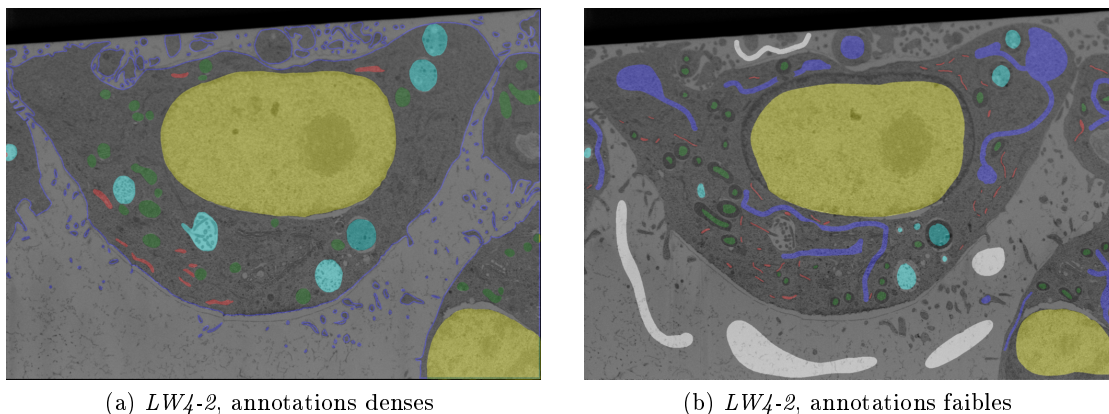


FIGURE 4.32 – Exemple d'annotations pour l'image *LW4-2*. Dans le cas des annotations faibles, la membrane cellulaire est représentée par deux classes, une classe « intérieur » (bleu) et une classe « extérieur » (blanc).

La première approche que nous avons suivie utilise un CNN classifiant le pixel central et minimisant l'entropie croisée, L_{CE} . L'utilisation de cette méthode avec des annotations faibles est triviale, il suffit de n'utiliser que les patches ayant comme pixel central un pixel annoté pour l'entraînement. Nous avons effectué plusieurs expériences préliminaires (pas de répétitions ou de résultats statistiques) sur l'image *I3* pour deux classes : mitochondrie et réticulum endoplasmique. Cette méthode fonctionne correctement, mais les contours sont généralement assez bruités (figure 4.33), et le temps d'inférence est extrêmement long car les pixels sont prédits un par un, patch par patch.

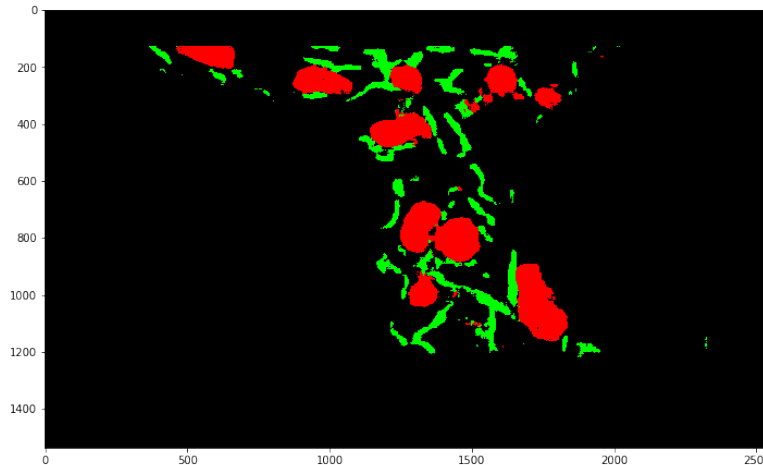


FIGURE 4.33 – Segmentation utilisant un CNN entraîné sur des annotations faibles. La figure 4.35a correspond à la segmentation de référence.

Pour résoudre ces problèmes, une solution consiste à entraîner un modèle de segmentation (type U-Net) à l'aide des annotations faibles. Durant la phase d'apprentissage, la différence entre les méthodes utilisant annotations classiques et faibles réside dans le calcul de la fonction de perte. La fonction de perte n'est alors calculée que sur les pixels annotés. Techniquement, on applique un poids (binaire) qui dépend de la présence d'une annotation, pour chaque élément du calcul de la loss.

Pour simuler les annotations faibles, nous avons utilisé une méthode qui consiste à squelettiser la segmentation avant de la dilater. La figure 4.34c représente ces annotations faibles simulées. On remarque que dans le cas de la classe du réticulum endoplasmique, notre simulation ne dégrade que très peu les annotations. De plus, pour les mitochondries, dans certains cas, le contour est tout de même bien défini grâce à la classe de fond.

Nos résultats expérimentaux permettent de comparer visuellement une segmentation de référence (figure 4.35a), une segmentation utilisant un modèle entraîné sur des annotations denses (figure 4.35b) et une segmentation utilisant un modèle entraîné sur des annotations faibles simulées (figure 4.35c).

Si l'on regarde attentivement les différentes segmentations, on peut remarquer deux caractéristiques principales. Premièrement, les deux modèles segmentent du réticulum endoplasmique en haut à gauche de l'image (flèches cyan). C'est en réalité un problème provenant de la segmentation de référence pour laquelle des erreurs humaines restent présentes. Deuxièmement, la segmentation proposée par le modèle utilisant les annotations faibles est plutôt satisfaisante. La majorité des objets sont bien segmentés, leurs bords sont déformés, mais les éléments sont bien détectés. Avec ces résultats, nous avons montré qu'il est possible d'entraîner un modèle avec des annotations faibles simulées. Les résultats de la segmentation sont satisfaisants, même si la méthode entraînée sur des segmentations denses est qualitativement meilleure.

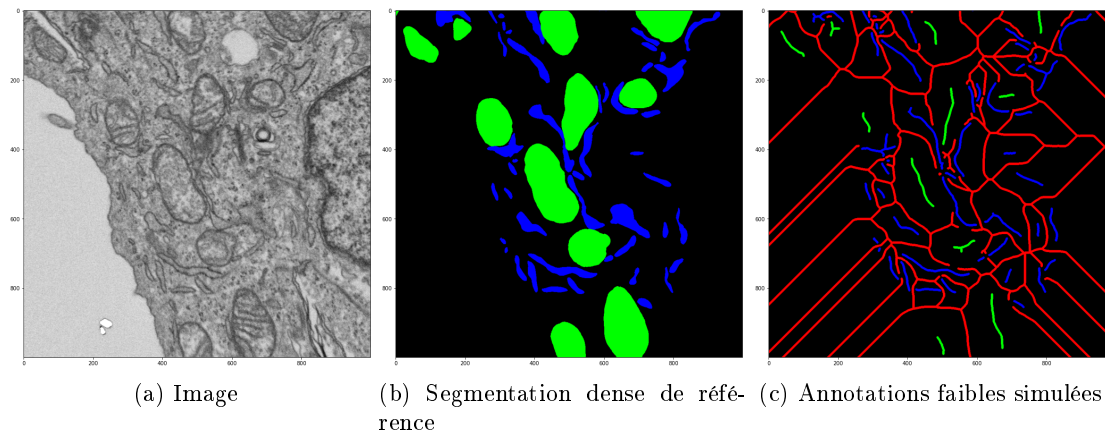


FIGURE 4.34 – Simulation des annotations faibles par squelettisation. La classe rouge représente le fond.

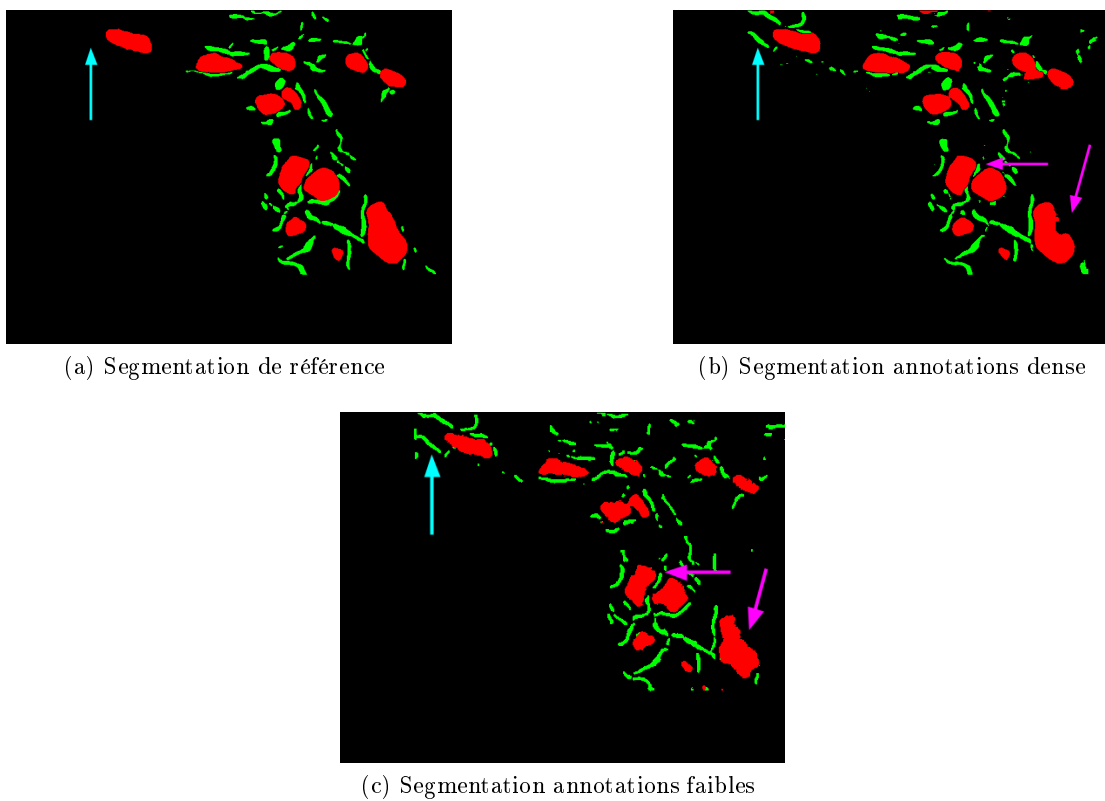


FIGURE 4.35 – Comparaisons entre modèles entraînés par annotations denses ou faibles. Les flèches cyan indiquent le réticulum absent de la segmentation de référence. Les flèches violettes indiquent deux mitochondries dont les bords sont déformés.

4.3.4 Résultats de segmentation avec annotations faibles

La dernière expérience que nous avons menée consiste à répéter l'expérience d'entraînement d'un modèle avec des annotations faibles, mais sur un jeu de données réelles. Pour cette expérience, nous utilisons l'image *LW4-2* et ses 25 coupes faiblement annotées (exemple d'une coupe figure 4.32).

Le modèle est entraîné sur 20 coupes, 5 coupes sont utilisées pour la validation et le test est fait sur 5 coupes qui ont été annotées de manière dense afin d'évaluer le modèle.

La figure 4.36 représente le score de chaque coupe de test individuellement. Les figures 4.37 et 4.38 représentent respectivement la segmentation de référence utilisée pour l'évaluation, et la segmentation proposée par le modèle.

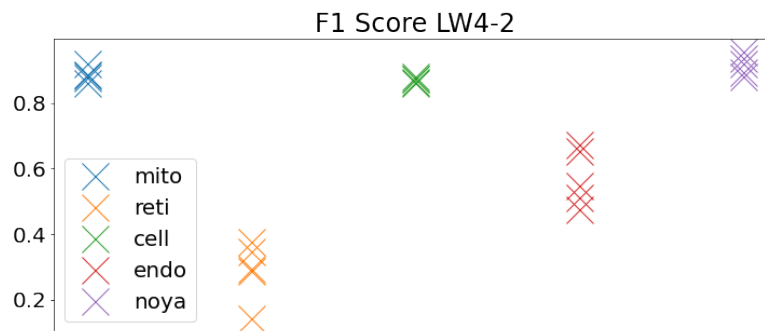


FIGURE 4.36 – F1-Score sur les 5 coupes de tests pour les 5 classes étudiées.

Les résultats (F1-Score) du modèle sur le jeu de test sont les suivants :

— Mitochondrie	88.55%
— Réticulum endoplasmique	28.88%
— Cellule	87.07%
— Endosome	57.27%
— Noyau	91.73%

Pour les classes mitochondrie, cellule et noyau, la segmentation est satisfaisante, tant qualitativement que quantitativement parlant. Pour les endosomes, le résultat est mitigé, visuellement les résultats sont plutôt bons, certains objets sont sur-détectés, mais ce sont surtout les artefacts présents aux bords de l'image qui posent problème (voir en bas à droite de la figure 4.38). Enfin, pour le réticulum endoplasmique, les résultats sont très variables, et la majorité des objets détectés ne sont en fait que des reliefs présents dans la cellule. Dans notre expérience simulant les interactions, nous avions de bons résultats, mais nous avons également précisé que notre simulation ne dégradait que très peu les annotations du réticulum.

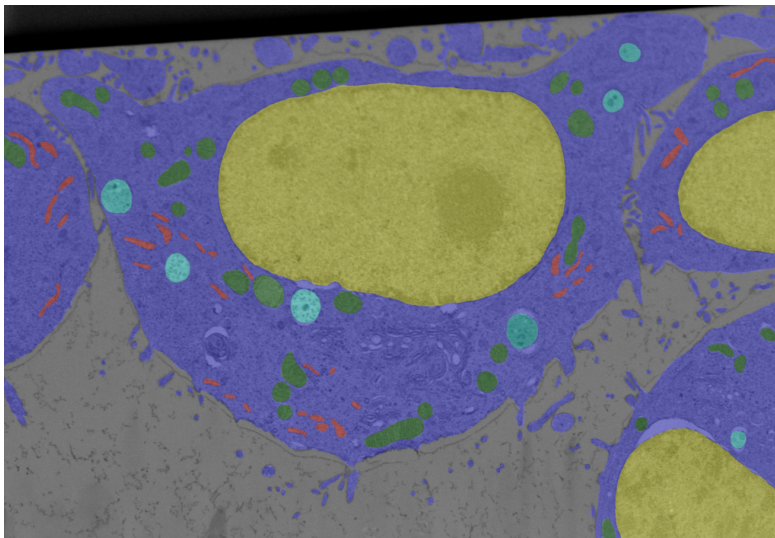


FIGURE 4.37 – Une coupe de segmentation de référence de l'image *LW4-2*.

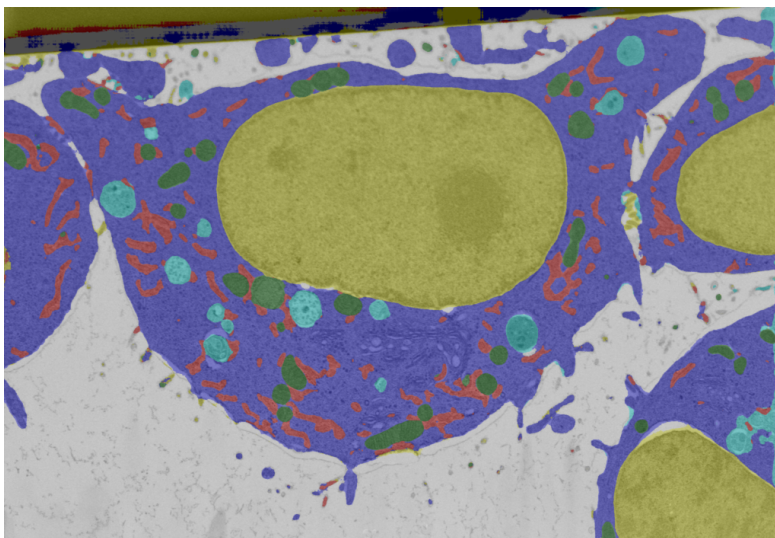


FIGURE 4.38 – Prédiction d'une coupe de l'image *LW4-2* par un modèle entraîné sur 25 coupes avec annotations faibles.

4.3.5 Conclusion

Dans cette section, nous avons évalué l'utilisation de modèles interactifs et entraînés avec des annotations faibles. Nous avons vu que dans les deux cas, ces techniques peuvent être utilisées afin d'entraîner un modèle, tout en réduisant le temps passé par l'utilisateur à préparer des segmentations de référence.

Au final, l'approche utilisant les annotations faibles semble plus pertinente, mais de nouvelles méthodes interactives pourraient changer cette situation. En particulier, nous nous sommes focalisés sur la création d'un modèle spécifique pour segmenter rapidement des coupes entières, mais une autre approche serait de développer un modèle général segmentant n'importe quel objet unique (une mitochondrie, une cellule, etc.).

On peut par exemple citer la méthode SAM (Segment Anything Model) récemment publiée [Kirillov et al., 2023], des exemples de segmentation automatiques utilisant cette méthode sont présents dans la figure 4.39. Les objets de l'image *LW4-1* (sauf reticulum endoplasmique) sont correctement reconnus alors que cette méthode n'est en aucun cas entraînée sur des données spécifiques de microscopie. Le noyau et la cellule sont quant à eux parfaitement segmentés. Pour l'image *I3*, la segmentation permet de segmenter correctement le noyau et la cellule, mais les autres objets ne sont pas détectés. Il est important de rappeler que ces segmentations ont été produites à l'aide de la fonction de segmentation automatique de la méthode SAM, donc sans aucune interaction utilisateurs.

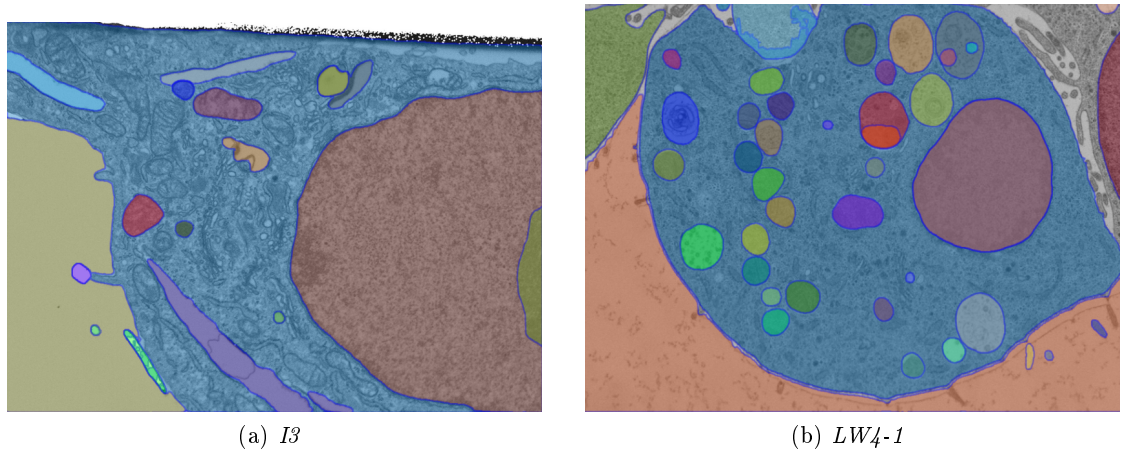


FIGURE 4.39 – Exemples de coupes segmentées en utilisant SAM [Kirillov et al., 2023].

Enfin, la méthode utilisant les annotations faibles permet une autre amélioration, celle de pouvoir adapter la quantité et la qualité des segmentations de références aux différentes classes. Nous avons vu dans la sous-section 4.2.1 que l'un des avantages des modèles binaires est de permettre l'entraînement sur des tranches où une seule organelle est annotée. Il est dans ce cas possible d'adapter la quantité d'annotation produite pour chaque classe. Dans le cas des annotations faibles, on peut bénéficier simultanément des avantages des modèles multi-classes et binaires en utilisant un seul modèle, sans avoir à annoter toutes les coupes pour toutes les organelles.

4.4 Morphologie mathématique et apprentissage profond

La morphologie mathématique est une théorie pour l'analyse de structure (voir sous-section 2.2 et [Soille, 2004]). Dans cette section, nous verrons les différentes techniques que nous avons mises en œuvre pour évaluer l'utilisation de méthodes issues de la morphologie mathématique, mise en relation avec les méthodes d'apprentissage profond. Nous verrons tout d'abord les combinaisons utilisant les opérations élémentaires et leurs compositions, puis nous nous intéresserons aux opérateurs basés sur les structures hiérarchiques [Salember and Wilkinson, 2009]. Nous présenterons dans chaque sous-section nos différentes expériences qui mélangent les concepts de morphologie mathématique présentés et apprentissage profond. Nous ne nous intéresserons qu'au cas des images finies, discrètes en niveaux de gris.

4.4.1 Opérateurs morphologiques

Dans cette première sous-section, nous verrons tout d'abord comment sont construits les opérateurs classiques de la morphologie mathématique reposant sur les opérations élémentaires de l'érosion et la dilatation.

Une des idées de base de la morphologie mathématique est de traiter un ensemble, dans notre cas une image, à l'aide d'un autre ensemble, appelé élément structurant, qui sert de sonde. Pour chaque position de l'élément structurant, on regarde s'il est inclus dans l'image initiale et on construit un ensemble de sortie, notre image traitée. Le pixel central de l'élément structurant, appelé origine, identifie le pixel dans l'image en cours de traitement.

On définit une image de dimension n comme une fonction $f : E \rightarrow T$, où E est l'espace des points (\mathbb{Z}^n ou \mathbb{R}^n) et T l'échelle des niveaux de gris (inclus dans $\overline{\mathbb{R}}$). L'élément structurant est représenté par une fonction structurante $b : B \rightarrow T$, $B \subseteq E$.

On peut différencier deux types d'éléments structurants, les éléments plats et non-plats. Un élément structurant plat est un cas particulier où la fonction structurante vaut 0 pour les points appartenant au support de l'élément structurant, et $-\infty$ ailleurs :

$$b(p) = \begin{cases} 0, & \text{pour } p \in B \\ -\infty & \text{sinon} \end{cases}$$

L'érosion et la dilatation sont les deux opérations élémentaires de la morphologie mathématique. On note ε l'opération d'érosion et δ la dilatation.

Pour chaque élément de l'image, la valeur de l'élément érodé est l'infimum dans la fenêtre définie par l'élément structurant. L'équation (4.1) correspond à la définition de l'érosion ε par une fonction structurant b .

$$[\varepsilon_b(f)](p) = \inf_{q \in B} f(p+q) - b(q) \tag{4.1}$$

$$\varepsilon_b(f) = f \ominus b \tag{érosion}$$

Pour chaque élément de l'image, la valeur de l'élément dilaté est le supremum dans la fenêtre définie par l'élément structurant. L'équation (4.2) correspond à la définition de la dilatation.

$$[\delta_b(f)](p) = \sup_{q \in B} f(p - q) + b(q) \quad (4.2)$$

$$\delta_b(f) = f \oplus b \quad (\text{dilatation})$$

Une dilatation agrandit les éléments clairs de f , tandis qu'une érosion les réduit. On peut utiliser ces deux opérations élémentaires pour construire d'autres opérations comme l'ouverture et la fermeture. L'ouverture γ d'une image f par un élément structurant B noté $\gamma_B(f)$ correspond à la dilatation de l'érosion de f par B . Respectivement, la fermeture φ d'une image f par un élément structurant B noté $\varphi_B(f)$ correspond à l'érosion de la dilatation de f par B . Les équations (4.3) et (4.4) qui suivent sont équivalentes dans le cas où B est non-plat (B_v)

$$\gamma_b(f) = (f \ominus b) \oplus b \quad (4.3)$$

$$\gamma_b(f) = f \circ b \quad (\text{ouverture})$$

$$\varphi_b(f) = (f \oplus b) \ominus b \quad (4.4)$$

$$\varphi_b(f) = f \bullet b \quad (\text{fermeture})$$

Une ouverture supprime les parties étroites de f et une fermeture bouche les trous étroits de f . Le terme étroit est à prendre au sens de l'élément structurant.

On peut également utiliser le résidu d'une transformation, la différence entre l'image originale et l'image transformée par une opération morphologique. L'opération de chapeau haut de forme blanc (white top-hat, (4.5)) correspond au résidu de l'ouverture. Respectivement, l'opération de chapeau haut de forme noir (black top-hat, (4.6)) correspond au résidu de la fermeture.

$$WTH_b(f) = f - \gamma_b(f) \quad (4.5)$$

(white top-hat)

$$BTH_b(f) = \varphi_b(f) - f \quad (4.6)$$

(black top-hat)

L'opération de gradient morphologique correspond à la différence entre la dilatation et l'érosion.

$$Grad_b(f) = \delta_b(f) - \varepsilon_b(f) \quad (4.7)$$

(gradient morphologique)

Nous nous limitons ici aux définitions nécessaires pour la compréhension de nos travaux ; de plus amples informations peuvent être trouvées dans les livres suivants : [Soille, 2004, Najman and Talbot, 2008, Talbot and Najman, 2010, Najman and Talbot, 2013].

Nous allons présenter 3 expérimentations combinant morphologie mathématique classique et apprentissage profond. Une première expérience où nous proposons de pré-traiter les images avec des opérations morphologiques, une suivante où nous combinons images pré-traitées et images originales et une dernière utilisant des couches de traitement morphologique dans un réseau de neurones.

4.4.2 Pré-traitements des images

Les traitements utilisant des opérateurs morphologiques permettent de modifier l'aspect des objets de l'image. Ces modifications peuvent améliorer ou non la perception des objets par un œil humain, la figure 4.40 représente une coupe de l'image $I\mathcal{B}$ ayant subi différentes transformations ; on peut par exemple noter que l'opération de fermeture (figure 4.40d) met légèrement en avant la structure interne des mitochondries. En outre, les opérateurs morphologiques sont non linéaires. Dans un réseau de neurones convolutifs, les convolutions sont des opérations linéaires et ce sont les fonctions d'activations non linéaires qui permettent au réseau de traiter des problèmes complexes. Les opérateurs morphologiques, non linéaires, ont donc un intérêt certain lorsque l'on s'intéresse à leurs applications aux réseaux de neurones.

La première expérience que nous avons menée consiste à pré-traiter les images avec des opérateurs morphologiques afin d'observer l'impact sur la classification. La configuration de l'expérience est similaire à celle de la sous-section 4.2.4 ; on utilise un CNN pour classifier le pixel central de l'image.

Expérience

Nous avons utilisé 5 pré-traitements différents et pour chaque traitement 4 éléments structurants. Les éléments structurants proposés sont des disques de rayons croissants. Pour toutes les expériences de cette section, un paramètre générique que l'on nomme λ , représente les variations d'élément structurant. Les opérateurs morphologiques utilisés sont les suivants :

- érosion ;
- dilatation ;
- ouverture ;
- fermeture ;
- Chapeau haut de forme noir.

La valeur λ représente le rayon du disque et utilise les valeurs entières de l'intervalle $[1, 5]$.

Les résultats de notre expérience indiquent de manière concluante que tous les pré-traitements utilisés ont un impact négatif sur la classification. La figure 4.41 représente

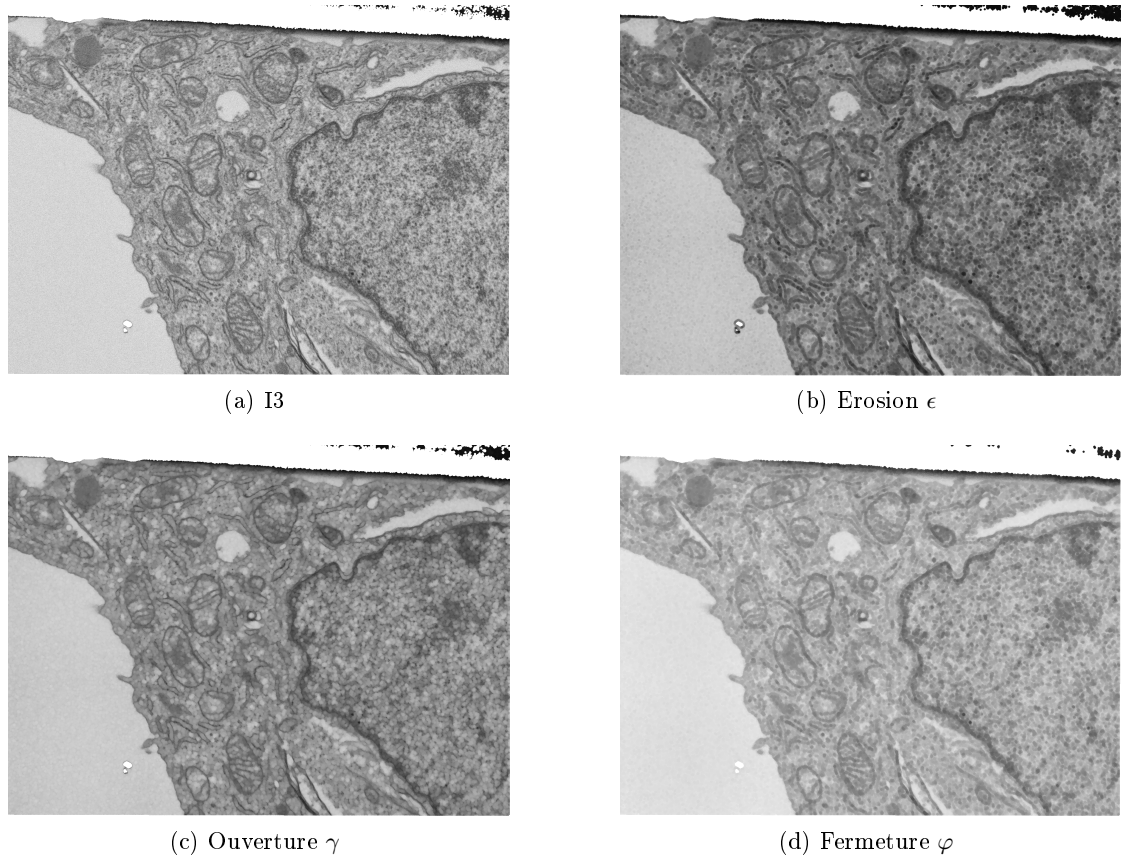


FIGURE 4.40 – Exemple de pré-traitement d’une coupe de l’image I_3 avec un élément structurant en forme de disque de rayon 4.

l’exactitude de classification sur le jeu de test pour les deux meilleurs cas, l’ouverture et la fermeture, la valeur $\lambda = 0$ correspond au cas contrôle. Dans les 3 autres cas, les résultats sont encore plus fortement impactés.

Les pré-traitements ne sont donc pas utiles, ils ont même un effet négatif. Cela pourrait être attribué au fait que les transformations morphologiques détruisent une partie de l’information. Pour éviter cette destruction, nous proposons donc de combiner ces images pré-traitées avec les images originales.

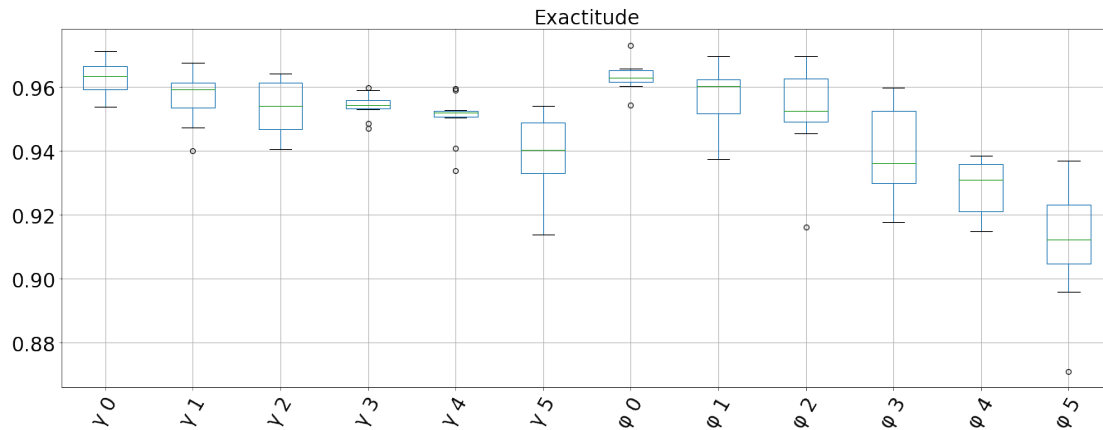


FIGURE 4.41 – Exactitude sur le jeu de test avec un pré-traitement utilisant les opérations d’ouverture et de fermeture.

La seconde expérience que nous avons menée consiste à utiliser en entrée les images pré-traitées et les images originales en même temps. La combinaison est effectuée en concaténant les images en utilisant le canal généralement utilisé pour les couleurs. Cette opération peut se rapprocher d’une forme d’extraction d’attributs non linéaire, même si ce n’est pas dans cette optique que l’expérience a originalement été pensée.

Expérience

La configuration de l’expérience est similaire à celle de la sous-section précédente, mais on concatène plusieurs images en entrée du réseau. Pour cette expérience, seuls les opérateurs ayant donné les meilleurs résultats précédemment ont été sélectionnés : l’ouverture et la fermeture. La valeur λ représente le rayon du disque, la valeur 0 correspond à l’image originale, les autres valeurs, les différents pré-traitements. Par exemple, dans le cas $\gamma[0\ 2\ 4]$, l’entrée est composée de l’image originale et de deux images pré-traitées avec des éléments structurants en forme de disque de rayon 2 et 4.

Une des idées que nous cherchions à développer était d’utiliser ces traitements pour réduire la taille nécessaire des réseaux de neurones utilisés ou la durée d’apprentissage. Le modèle a été légèrement modifié afin de prendre plusieurs images en entrée, mais le nombre de paramètres a également été légèrement réduit, ce qui explique une légère baisse de performance dans le cas contrôle.

Les résultats de cette expérience sont moins extrêmes, mais les pré-traitements utilisés n’ont jamais d’impact positif. La figure 4.42 représente l’exactitude de classification sur le jeu de test pour le meilleur cas, celui de l’ouverture. La figure 4.43 représente des exemples d’évolution de loss sur plusieurs époques pour le jeu d’entraînement et le jeu de validation. Le bruit sur les courbes du jeu de validation peut être expliqué à cause du nombre très réduit d’exemples utilisés pour calculer le score de validation.

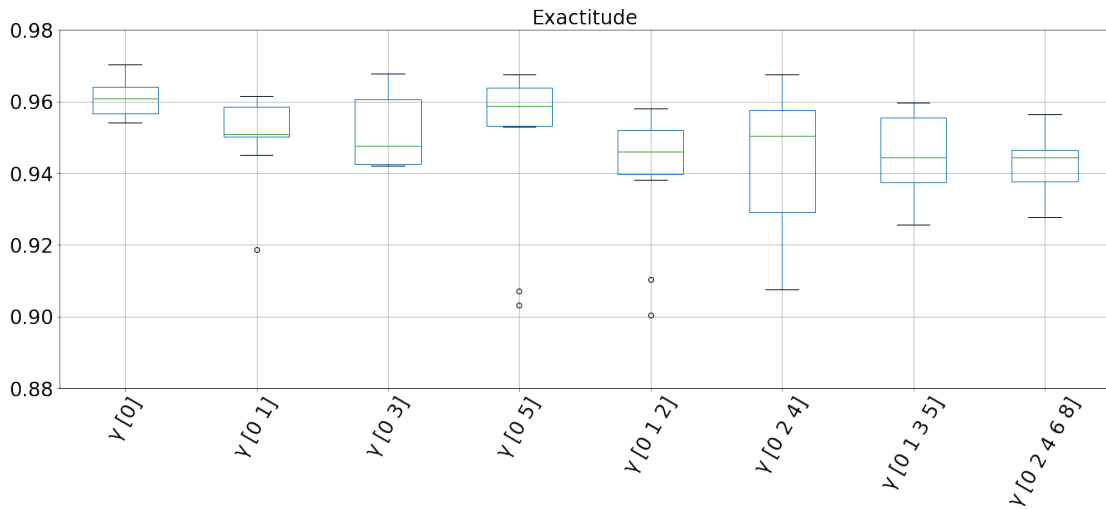


FIGURE 4.42 – Exactitude sur le jeu de test avec une combinaison d’images et de pré-traitement utilisant les opérations d’ouverture.

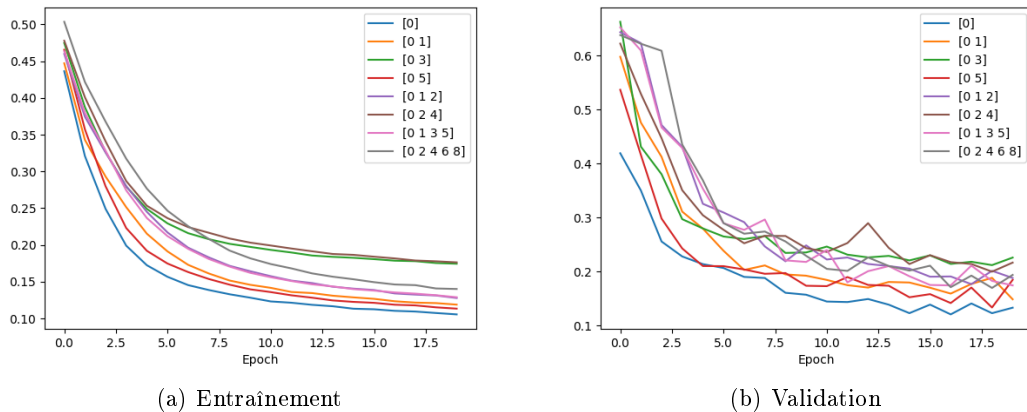


FIGURE 4.43 – Exemple d’entraînement pour l’expérience de combinaisons d’images originales et de pré-traitements utilisant les opérations d’ouverture.

Les résultats de cette expérience nous ont bien plus étonnés que ceux de la première. On pourrait imaginer que dans le pire des cas, ajouter de l’information devrait ne pas avoir d’impact, mais au contraire, il y a bien souvent un impact négatif. A la suite de ces résultats, nous avons décidé d’utiliser d’autres traitements que nous verrons dans la sous-section 4.4.4, mais restons pour l’instant dans le cadre des opérateurs morphologiques classiques.

4.4.3 Réseaux de neurones et couches morphologiques

En plus des expérimentations de pré-traitements des images, nous avons développé une autre idée, l'utilisation de couches morphologiques dans le réseau de neurones. L'objectif est d'utiliser des couches de traitement non pas composées de convolutions, mais d'opérations morphologiques élémentaires d'érosion et de dilatation. La première étape est de vérifier l'utilisation et la faisabilité technique de ces couches en expérimentant **l'apprentissage d'un élément structurant**. De telles expérimentations ont déjà été menées, et de nombreux chercheurs s'intéressent à ces problèmes, on peut par exemple citer les travaux suivants : [Mondal et al., 2019, Mondal et al., 2020, Kirszenberg et al., 2021, Hermary et al., 2022, Velasco-Forero et al., 2022a, Velasco-Forero et al., 2022b]. Dans cette expérience, nous nous intéressons uniquement à la faisabilité technique de la méthode sans chercher à l'améliorer. L'implémentation utilisée pour les couches morphologiques provient des travaux de [Mondal et al., 2019].

L'objectif est d'entraîner un modèle à reproduire une opération morphologique connue, dans notre cas, une érosion ou une dilatation. Nous effectuons nos expérimentations sur un jeu de données synthétique ; nous générons une image en niveau de gris à partir de formes aléatoires, dans notre cas, des disques. La figure 4.44 montre l'image d'entraînement générée et ses versions dilatée et érodée. La durée totale de l'entraînement de ces modèles ne dépasse pas la minute.

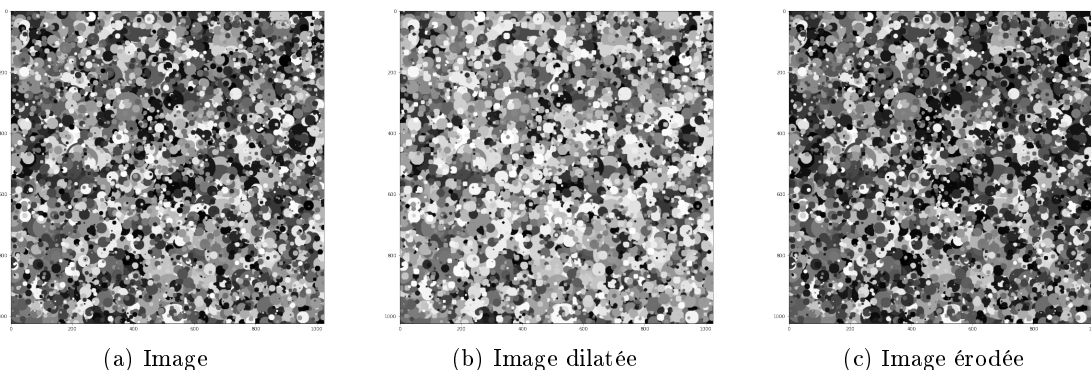
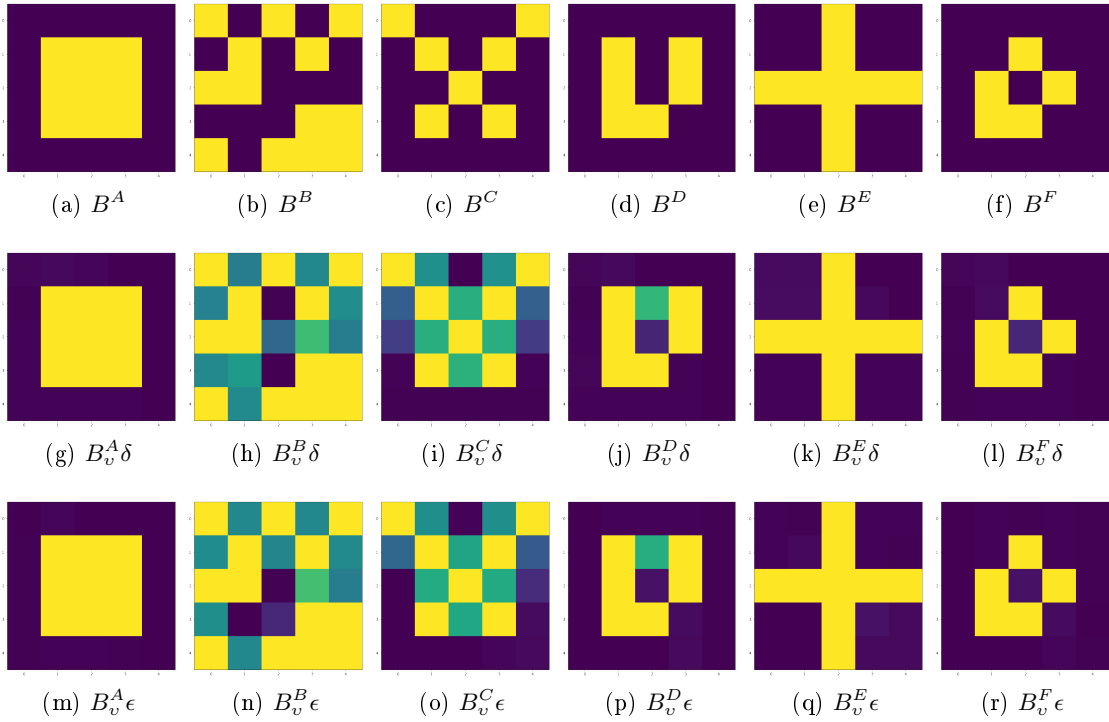


FIGURE 4.44 – Image d'entraînement synthétique utilisée pour l'entraînement.

Le « réseau » si on peut l'appeler ainsi est composé d'une unique couche d'érosion (ou de dilatation) et les poids représentent un élément structurant non-plat 5×5 . Les poids représentant l'élément structurant sont initialisés à 0. On entraîne le modèle avec des patchs composés de l'image originale et de l'image modifiée pour lui apprendre à reproduire l'opération souhaitée. Après l'entraînement, on regarde l'état des poids de la couche afin de voir si l'on a correctement reproduit l'élément structurant. Cinq éléments structurants différents ont été utilisés, comprenant un carré et une croix. La figure 4.45 représente les résultats d'apprentissage des éléments structurants pour une érosion ou une dilatation.

FIGURE 4.45 – Résultats de l'apprentissage des éléments structurants U .

On remarque que pour les formes d'élément structurant les plus simples (B^A , B^E , B^F), les poids correspondent exactement au résultat attendu. Les résultats sont moins bons pour les formes complexes (B^B , B^C) mais on reconnaît tout de même la forme initiale dans le résultat. Il n'y aucune différence de résultats entre apprentissage de l'érosion et de la dilatation, résultats attendus, les deux opérations étant duales. Après cette validation technique où nous avons montré que les couches peuvent bien effectuer les opérations non-linéaires d'érosion et de dilatation tout en apprenant les poids, nous pouvons essayer d'utiliser ces couches afin de traiter nos images.

Nous avons effectué plusieurs expériences préliminaires afin de déterminer comment utiliser ces nouvelles couches de manière à segmenter nos images. Tout d'abord, nous avons essayé de créer un réseau morphologique pour la classification du pixel central. Le réseau ressemble à un CNN classique, mais les opérations habituelles de convolutions sont remplacées par des opérations d'érosion et de dilatation. De la même manière, nous avons créé un U-Net morphologique, remplaçant les habituelles convolutions par des érosions et dilatations. Dans les deux cas, nous n'avons pas réussi à entraîner correctement ces réseaux de manière à obtenir des segmentations convaincantes.

Expérience

Dans la dernière expérience que nous avons menée, nous utilisons les couches morphologiques en supplément des couches habituelles dans un modèle U-Net. Cette expérience correspond à la même installation expérimentale que celle de la sous-section 4.2.3 où nous comparons différentes variations dans le modèle U-Net. Les blocs morphologiques correspondent à une couche d'érosion ou de dilatation avec un élément structurant non plat, où l'élément structurant correspond aux poids de la couche. Ces ajouts morphologiques sont uniquement présents dans la partie encodeur du modèle et non dans la partie décodeur.

Nous comparons les résultats de segmentation des 5 modèles suivants ; nous notons X l'entrée de la couche courante et C_M le module de traitement classique :

- U-Net, couches de traitement classique, figure 4.8a : $C_M(X)$
- U-Net-Res, couche de traitement résiduel, figure 4.8b : $C_M(X) + X$
- U-Net Morpho α , $C_M(\epsilon(X) + \delta(X))$
- U-Net Morpho β , $C_M(X) + Conv(\epsilon(X) + \delta(X))$
- U-Net Morpho γ , $C_M(X) + Conv(Concat(\epsilon(X), \delta(X)))$

La couche de convolution supplémentaire présente dans les modèles β et γ utilise un noyau 1×1 et est uniquement utilisée pour faire correspondre les tailles des éléments à additionner.

La figure 4.46 montre les modules utilisés, les blocs bleus correspondent au bloc C_M . La figure 4.47 présente les résultats de segmentation sur le jeu de test.

Les résultats sont très clairs, si les ajouts proposés n'ont pas d'impact positif, les scores d'évaluation nous montrent qu'un modèle sans aucun ajout est meilleur et qu'un modèle utilisant des connexions résiduelles simples est strictement supérieur. Similairement à l'expérience de la sous-section 4.4.2, nous ne pensions pas que de tels ajouts pourrait détériorer les résultats.

Après avoir mené ces différentes expériences, nous souhaitons toujours apporter des améliorations à nos modèles en utilisant des méthodes issues de la morphologie mathématique, mais nous arrivions à court d'outils issus de la morphologie classique.

4.4.4 Structures hiérarchiques et opérateurs connexes

Dans cette sous-section, nous nous intéresserons aux structures hiérarchiques et aux opérateurs connexes sur les images en niveaux de gris. Cette sous-section s'inspire du livre de Najman et Talbot : [Najman and Talbot, 2008]. Les opérateurs connexes ont une propriété très intéressante, ils ne peuvent pas créer de nouveaux contours ou déplacer les frontières entre des régions existantes. Un opérateur Ψ est dit connexe si la partition des composantes connexes de son entrée (on note f , l'image d'entrée) est toujours plus fine que la partition des composantes connexes de sa sortie.

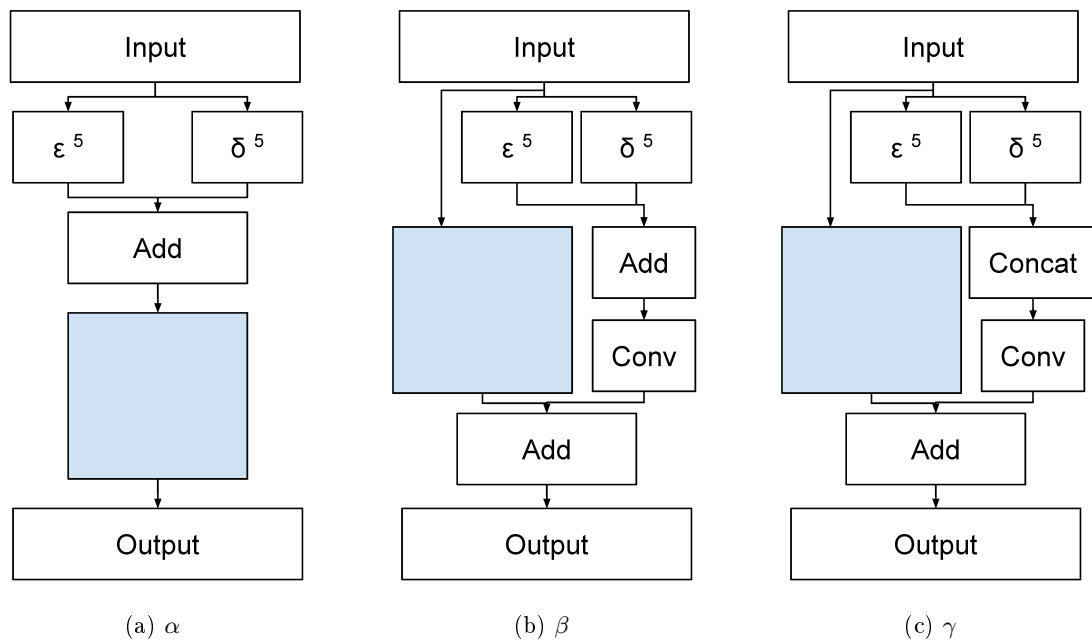


FIGURE 4.46 – Les blocs bleus représentent l'opération de calcul classique de la couche.

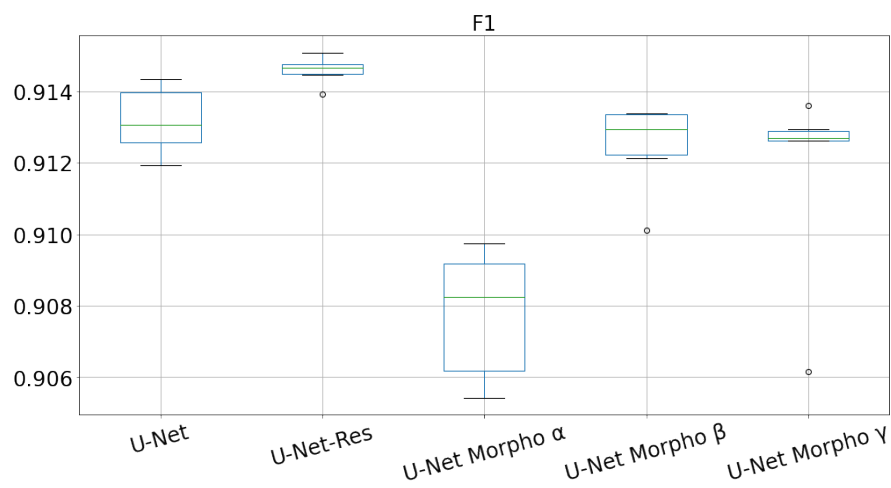


FIGURE 4.47 – Résultats de l'expérience utilisant U-Net et couches morphologiques, F1-Score.

$$\mathcal{P}_f \sqsubseteq \mathcal{P}_{\Psi(f)} \forall f \quad (4.8)$$

(opérateur connexe)

Il existe plusieurs méthodes pour construire des opérateurs connexes comme les méthodes par reconstruction (voir [Salembier and Wilkinson, 2009]) ou les méthodes utilisant des structures hiérarchiques. Nous présenterons uniquement les méthodes utilisant les structures hiérarchiques. Une structure hiérarchique est une représentation hiérarchique (un arbre) des composantes connexes de l'image.

Une des représentations hiérarchiques les plus simples est le max-tree (arbre des composantes, arbre des coupes) [Salembier et al., 1998]. En utilisant cette représentation, les nœuds représentent les zones plates de l'image. Un nœud de l'arbre est une composante connexe parmi les ensembles de niveaux de l'image. La relation de parenté entre nœuds correspond à la relation d'inclusion spatiale entre les composantes de différents niveaux. L'élagage de cette structure permet d'agir sur les maxima de l'image. La structure duale du max-tree est appelée min-tree, une manière de la construire est de construire le max-tree de l'image $-f$. Un algorithme linéaire [Salembier et al., 1998] pour le calcul de cette structure existe ainsi que plusieurs algorithmes utiles en fonction des caractéristiques des images et systèmes utilisés (bits par pixel, mémoire limitée) [Carlinet and Géraud, 2013].

Une autre représentation existante est l'arbre des formes (tree of shapes, inclusion-tree) [Monasse and Guichard, 1999, Monasse, 2000, Salembier and Wilkinson, 2009, Géraud et al., 2013], qui représente la relation d'inclusion des formes de l'image. L'arbre des formes est invariant aux changements et inversions de contrastes. Cette structure permet d'agir simultanément sur les maxima et les minima d'une image. La relation de parenté entre nœuds correspond à la relation d'inclusion des composantes connexes complètement entourées. Techniquement, l'arbre des formes peut être calculé avec un algorithme quasi linéaire [Géraud et al., 2013] ou en utilisant l'algorithme de calcul du max-tree [Carlinet et al., 2018].

La figure 4.48 représente une image d'exemple ainsi que les différentes représentations hiérarchiques mentionnées.

Le vocabulaire utilisé est changeant en fonction des auteurs. Certains utilisent le terme d'arbre des composantes pour décrire l'arbre des formes [Monasse and Guichard, 1999, Monasse, 2000, Najman and Talbot, 2008, Najman and Talbot, 2013], dans notre cas, nous utilisons le terme arbre des composantes pour décrire la structure la plus simple, le max-tree. Dans d'autre cas, les auteurs considèrent le max-tree et l'arbre des composantes comme deux entités distinctes, le max-tree étant une version plus compacte et non redondante de la structure [Berger et al., 2007]. De notre point de vue, il n'y a pas matière à faire de distinctions, les deux représentant théoriquement la même hiérarchie, ce n'est qu'un détail d'implémentation.

Une fois cette structure hiérarchique construite, on peut calculer des attributs pour chaque nœud de l'arbre. Dans certains cas, ces attributs peuvent être croissants comme l'aire ou le contraste. Dans d'autre cas, les attributs ne sont ni croissants, ni décroissants,

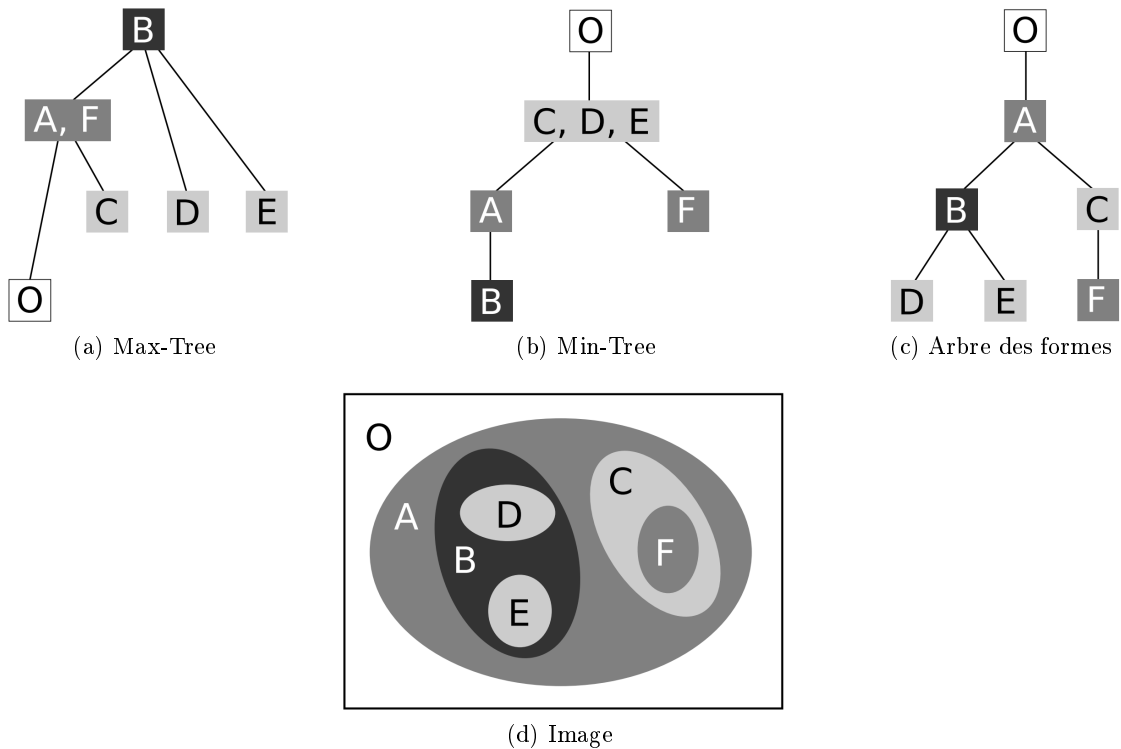


FIGURE 4.48 – Une image et ses différentes représentations hiérarchiques [Carlinet and Géraud, 2015].

comme le périmètre ou la complexité. Pour le max-tree, la hauteur correspond au niveau de gris minimum dans le nœud, pour le min-tree, c'est la valeur maximum. La table 4.4 liste plusieurs attributs pouvant être calculés pour les nœuds de l'arbre.

Une fois la structure construite et les attributs calculés, la stratégie de filtrage consiste à élaguer cette structure avant de reconstruire l'image. L'élagage consiste à simplifier l'image en éliminant l'information représentée par certaines branches, certains nœuds. Il existe plusieurs règles permettant de filtrer les arbres [Urbach and Wilkinson, 2002].

- direct : un nœud est supprimé s'il ne satisfait pas la condition, la valeur de ces pixels correspond à la valeur de niveau de gris de son ancêtre ;
- soustractif : direct, mais la valeur des descendants est soustraite à la valeur du nœud courant ;
- max : un nœud est supprimé s'il ne satisfait pas la condition ou si un ancêtre est supprimé ;
- min : un nœud est supprimé s'il ne satisfait pas la condition ou si un descendant est supprimé.

Attributs	Description	Croissant
Aire (A)	Le nombre de pixels dans le nœud	✓
Volume (V)	La somme de la différence entre la valeur des pixels du nœud et la hauteur du nœud	✓
Contraste (C)	Différence entre les valeurs minimales et maximales de la composante	✓
Contours (CT)	Nombre de pixels avec un voisin à l'intérieur et à l'extérieur de la composante	
Complexité (CPL)	CT/A	
Compacité (CPA)	A/CT^2	
Gradient aux Bords (MGB)	La valeur moyenne du gradient pour les pixels du contour	

TABLE 4.4 – Exemples d'attributs pouvant être calculés pour l'élagage des structures hiérarchiques.

C'est en élaguant l'arbre construit puis en reconstituant l'image que l'on applique l'opérateur connexe. La figure 4.49 représente les étapes nécessaires à la création d'un opérateur connexe.

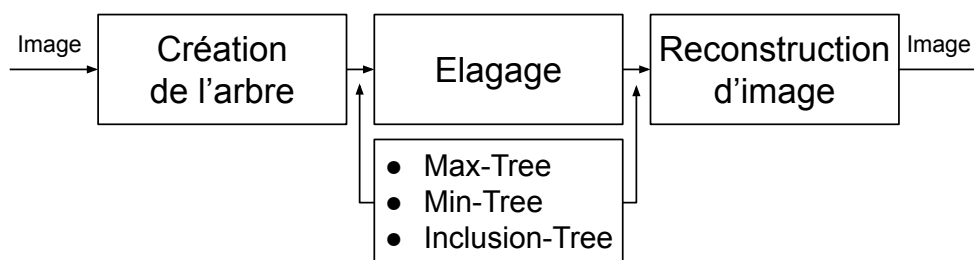


FIGURE 4.49 – Opérateurs connexes et représentation hiérarchique.

4.4.5 Pré-traitement des images par opérateurs connexes

Nos premières expériences ont consisté à appliquer des pré-traitements morphologiques (de manière similaire aux expériences de la sous-section 4.4.2) sur nos images. Plutôt que d'utiliser des opérateurs classiques, nous avons utilisé un opérateur connexe, l'opérateur d'ouverture d'aire. L'ouverture d'aire consiste à supprimer les nœuds dont la valeur d'attribut de l'aire est inférieure à un seuil, avec une règle de filtrage directe. Comme l'aire est un critère croissant, cela revient à donner au nœud courant la valeur de niveau du gris du premier ancêtre dont l'aire satisfait la condition. L'objectif est de simplifier l'image en supprimant les petites zones bruitées. Nous avons donc utilisé l'arbre des formes, qui permet de simplifier les petites zones, qu'elles soient claires ou sombres.

La première expérience consiste à classifier le pixel central de patches ayant été pré-traités. Nous nous intéressons uniquement à l'image $I\beta$ et aux mitochondries. Un exemple de coupe ayant subi les pré-traitements est disponible en figure 4.51. La figure 4.50 représente les résultats de cette expérience. De manière similaire aux expériences utilisant un filtre morphologique, l'utilisation d'un pré-traitement a un impact négatif croissant.

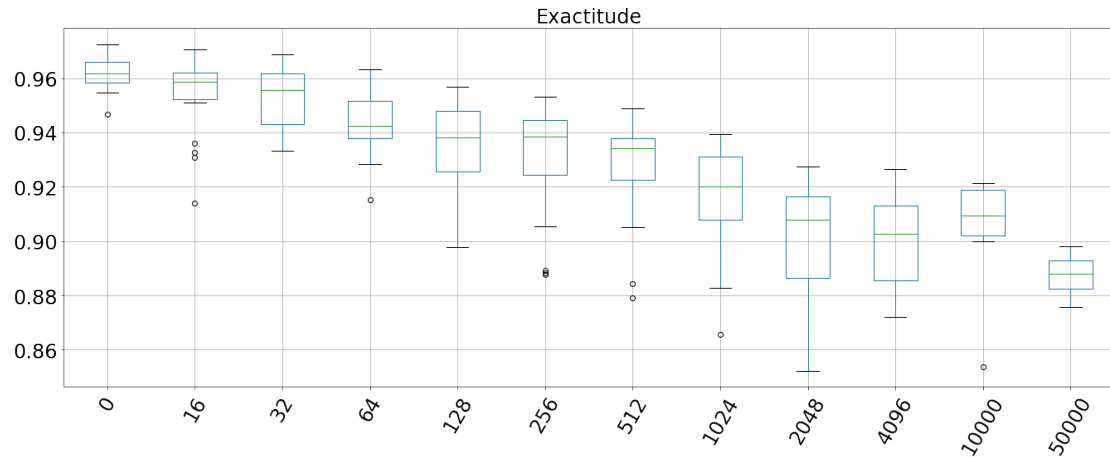


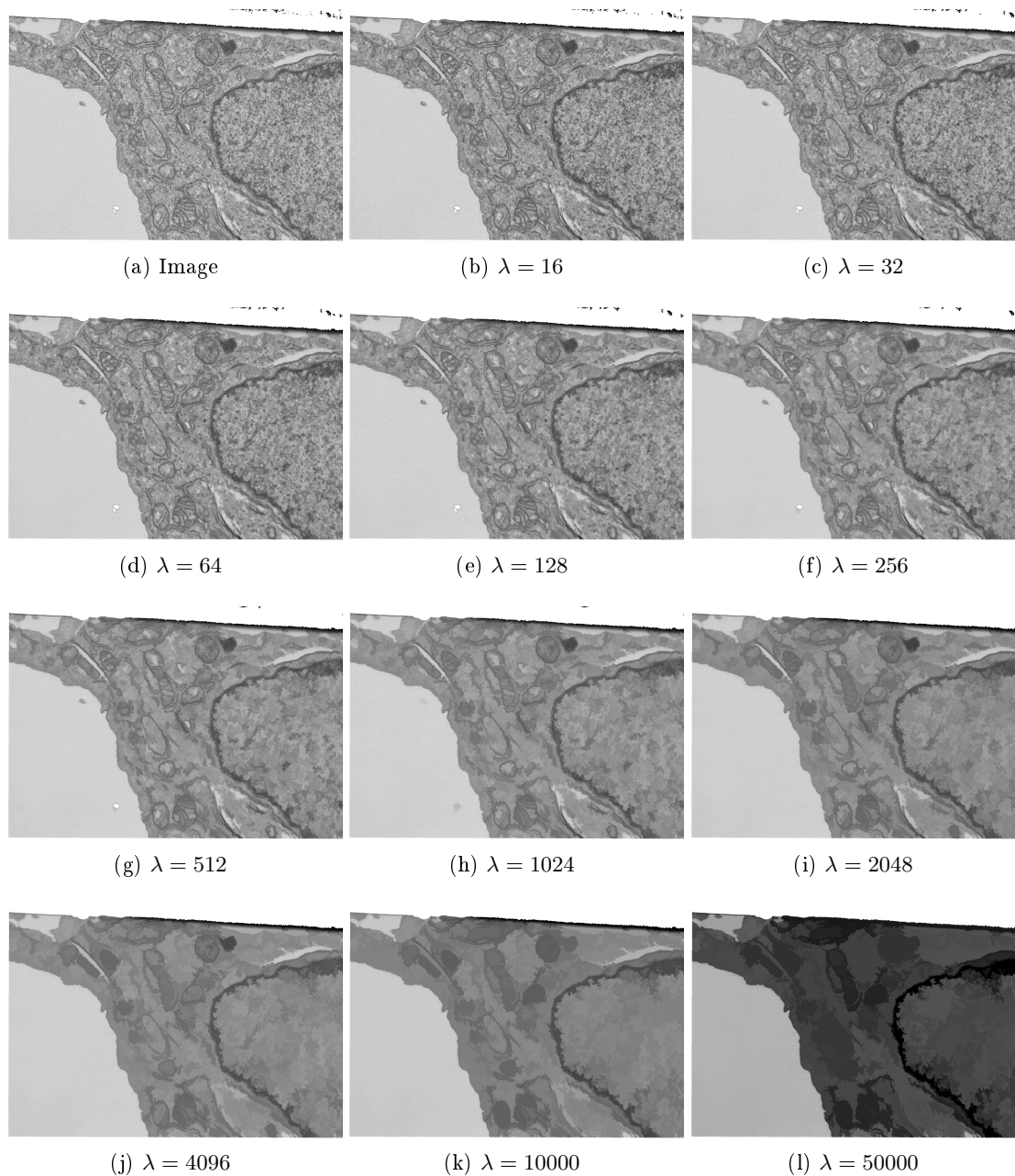
FIGURE 4.50 – Résultat de l'expérience de pré-traitement utilisant l'opération d'ouverture d'aire.

Comme les résultats n'étaient pas satisfaisants, nous avons agi de la même manière que dans la seconde expérience de la sous-section 4.4.2 ; nous avons combiné image originale et pré-traitements. Pour sélectionner la valeur λ correspondant à l'ouverture d'aire, nous avons choisi manuellement des valeurs d'aire qui correspondaient approximativement à la mise en évidence des organelles. Les valeurs retenues sont 650 (réticulum endoplasmique) et 3200 (mitochondrie). La figure 4.52 représente un patch ainsi que les deux pré-traitements utilisés.

Ces résultats ont fait l'objet de la publication suivante : [Meyer et al., 2021]. Dans le cadre de cette publication, nous avons également étudié l'amélioration des performances en utilisant un encodeur EfficientNet. Les figures 4.53 et 4.54 représentent les résultats de cette expérience.

Dans tous les cas, l'utilisation de l'encodeur EfficientNet améliore les résultats de segmentation. On remarque que dans le cas du réticulum endoplasmique sur l'image $I\beta$ (figure 4.54a), l'application des pré-traitements semble légèrement améliorer le score médian, mais en réalité, le score maximum est toujours atteint par le modèle « Eff U-Net ».

En conclusion de cette expérience, les résultats ne permettent pas de mettre en évidence une amélioration significative, mais au moins, on peut remarquer que pour la première fois, il n'y a pas de dégradations des résultats en utilisant un pré-traitement morphologique.

FIGURE 4.51 – Pré-traitements de l'image $I_{\mathcal{B}}$, ouverture d'aire sur l'arbre des formes.

4.4.6 Pré-traitement des images par extraction d'attributs

Bien que les résultats précédents soient meilleurs que ceux utilisant la morphologie classique, nous n'obtenons toujours pas d'améliorations significatives des résultats. Nous

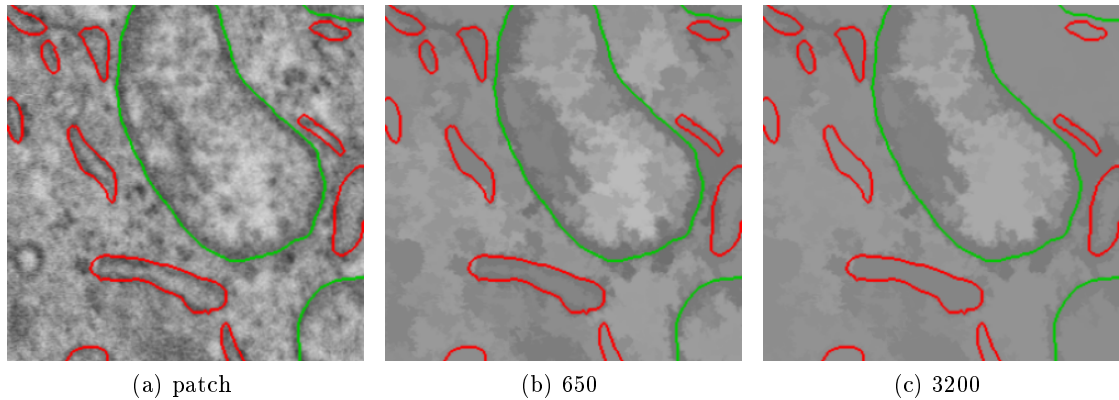


FIGURE 4.52 – Pré-traitements d’un patch de l’image I_3 , ouverture d’aire sur l’arbre des formes.

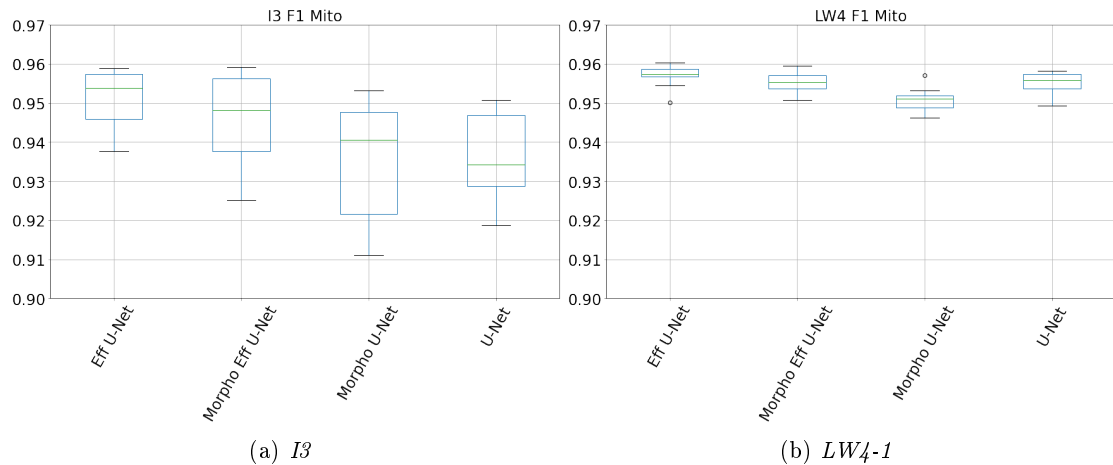


FIGURE 4.53 – Résultats d’entraînement pour l’expérience de pré-traitement par opérateurs connexes, mitochondries, F1-Score.

proposons une nouvelle approche : utiliser la structure hiérarchique représentant l’image comme extracteur d’attribut. Plutôt que de filtrer l’image à l’aide des attributs et de modifier les valeurs de niveaux de gris, nous utilisons directement la valeur des attributs comme valeur de sortie. Pour ces travaux, nous avons choisi de travailler avec le max-tree. Cette idée est inspirée des travaux de Farfan et al. [Farfan Cabrera et al., 2021a, Farfan Cabrera et al., 2021b], eux-mêmes découlant des recherches de Grossiord et al. [Grossiord et al., 2015, Grossiord et al., 2017]. Les travaux de Farfan et al. utilisent l’arbre des composantes afin d’extraire des cartes d’attributs d’images PET.

Un des problèmes lorsque l’on utilise cette méthode est de décider à quel nœud rattacher un pixel. Le pixel appartient au minimum à un nœud, la racine, mais dans

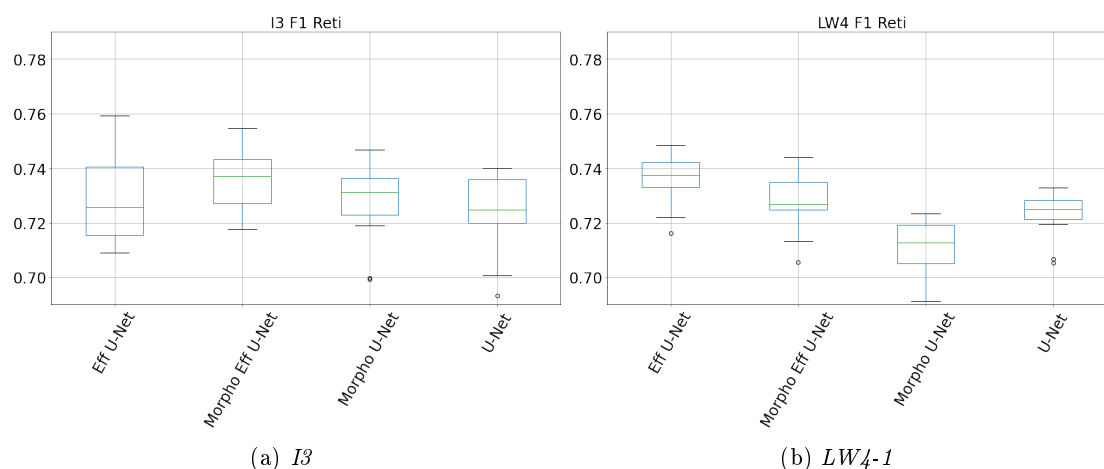


FIGURE 4.54 – Résultats d'entraînement pour l'expérience de pré-traitement par opérateurs connexes, réticulum endoplasmique, F1-Score.

la plupart des cas, le pixel appartient également à toute une branche, partant du nœud courant vers la racine. Dans leurs articles, les auteurs de [Farfan Cabrera et al., 2021b] définissent plusieurs stratégies possibles, garder le maximum, le minimum ou encore la moyenne dans la branche. Nous avons tout d'abord évalué cette méthode mais après plusieurs essais, les résultats n'étaient pas satisfaisants.

Pour résoudre ce problème, nous avons défini une méthode générale d'extraction d'attributs utilisant 3 attributs.

- un attribut de valeur ;
- un attribut de critère ;
- un attribut de restriction.

Pour chaque pixel p , l'ensemble des nœuds appartenant à la branche de p est récupéré, et seul un sous-ensemble de nœuds ayant une valeur d'attribut dans une certaine plage (attribut de restriction) est conservé. De cet ensemble, le nœud N_{best} optimisant un certain critère est conservé. Enfin, la valeur de p dans l'image résultante est définie par la valeur de l'attribut du nœud N_{best} . L'image résultante est normalisée dans la plage $V = \llbracket 0, 255 \rrbracket$.

L'attribut de restriction est un attribut optionnel qui permet de limiter la recherche dans la branche du nœud courant. Dans notre expérience, on limite par exemple la recherche aux nœuds ayant une aire comprise entre certaines valeurs. Seul le nœud courant est exempté de cette restriction. Pour le critère, nous avons développé un critère de stabilité Δ_A , basé sur le concept de régions extrêmes stables au maximum (Maximally Stable Extremal Regions [Matas et al., 2004]). L'idée est de récupérer les régions les plus stables en fonction de la variation de surface entre nœuds successifs puisque ces régions représentent les objets saillants de l'image. On note $Par(N)$ le nœud parent de N .

$$\nabla_A(N) = \frac{|A(\text{Par}(N)) - A(N)|}{|H(\text{Par}(N)) - H(N)|} \cdot \frac{1}{A(N)}$$

$$\Delta_A(N) = \nabla_A(\text{Par}(N)) - \nabla_A(N)$$

Pour imaginer cet attribut, nous avons développé une solution logicielle permettant de visualiser les différentes composantes connexes et les valeurs des attributs pour les branches. Ce logiciel consiste en une amélioration de la bibliothèque logicielle LibTIM ainsi qu'à l'ajout d'une interface graphique pour l'analyse manuelle des images. La figure 4.55 montre un exemple d'utilisation du logiciel GUILibTIM. L'image à gauche représente l'image source, le carré rouge le pixel sélectionné par l'utilisateur. Le graphique en bas à droite représente les valeurs de l'attribut (le contraste, en vert) et du critère ($\Delta_A(N)$, en bleu) sélectionné pour la branche du pixel. L'axe des abscisses représente le niveau de gris, l'axe des ordonnées de droite la graduation pour la valeur de l'attribut et l'axe des ordonnées de gauche la graduation pour la valeur du critère. Le point rouge en bas à droite du graphique correspond au nœud du pixel courant sélectionné. Le second point rouge correspond au nœud parent sélectionné par le critère (N_{best}). En haut à droite, sur l'image, le rouge correspond aux pixels du nœud N_{best} , en bleu, son contour. Une explication plus complète des fonctionnalités de cette solution logicielle est détaillée dans la sous-section A.1.3.

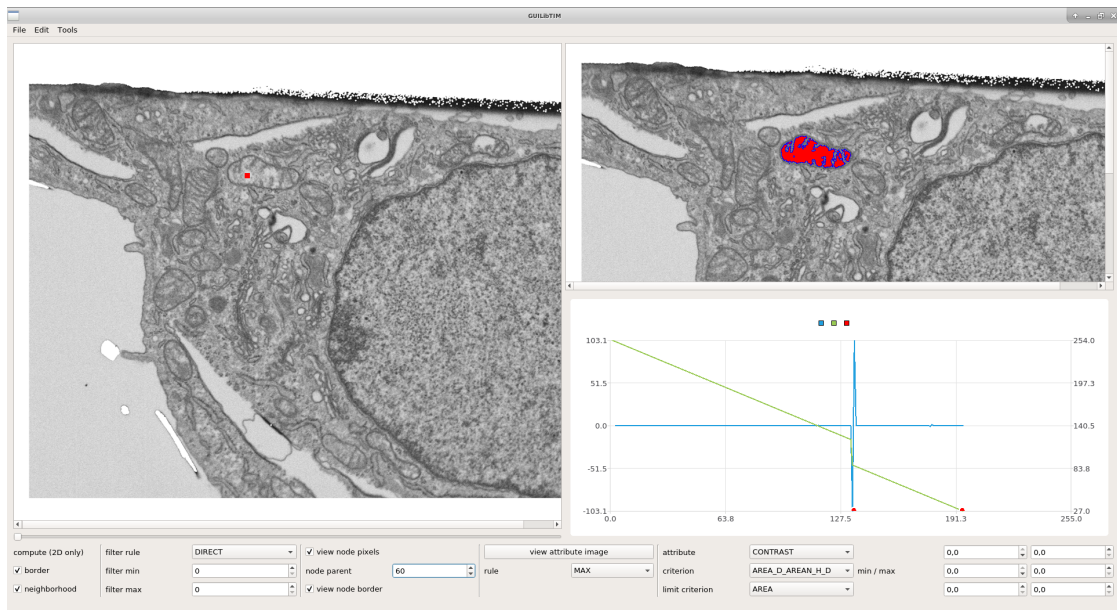


FIGURE 4.55 – Exemple d'utilisation de GUILibTIM pour analyser une image.

Pour résumer le fonctionnement de notre algorithme d'extraction d'attribut, on affecte à p la valeur de l'attribut A_{val} du nœud N_{best} optimisant le critère A_{crit} dans la branche restreinte par un attribut A_{lim} . Plusieurs combinaisons d'attributs ont été sélectionnées pour cette expérience afin de construire des extracteurs d'attributs. La table 4.5 liste ces

extracteurs ainsi que les attributs utilisés pour les construire. Les sous-figures 4.56c à 4.56g de la figure 4.56 représentent l'extraction d'attribut sur une coupe.

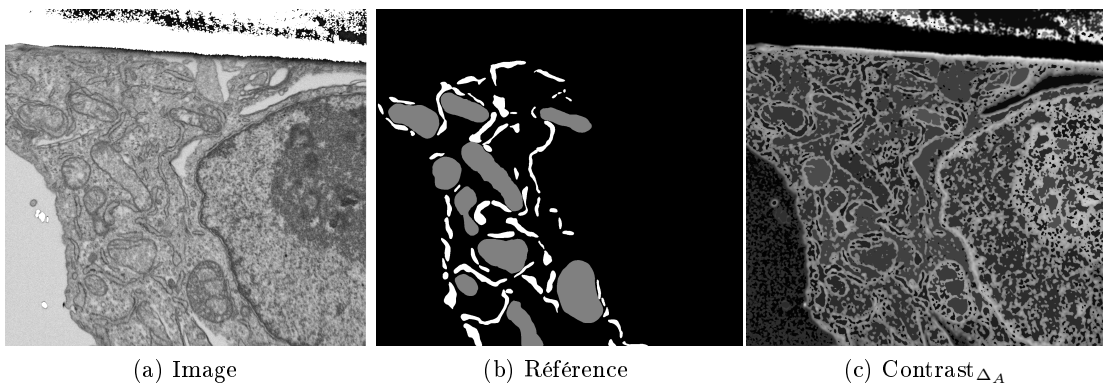
Nom	Attribut	Critère	Limite
Contrast $_{\Delta_A}$	C	Δ_A	$0 \leq A \leq 1024^2$
Complexity $_{\Delta_A}$	CPL	Δ_A	$0 \leq A \leq 1024^2$
Compacity $_{\Delta_A}$	CPA	Δ_A	$CPA \leq 50$
Volume $_{\Delta_A}$	V	Δ_A	$0 \leq A \leq 1024^2$
MGB	MGB	MGB	$0 \leq A \leq 1024^2$

TABLE 4.5 – Liste des extracteurs d'attributs utilisés pour l'expérience.

Nous avons également utilisé des opérateurs connexes comme dans l'expérience précédente, mais comme pour les extracteurs d'attributs, nous utilisons le max-tree. La table 4.6 liste les opérateurs utilisés. La colonne valeurs correspond aux intervalles de valeurs valides pour l'attribut, si le nœud ne satisfait pas la condition, il est filtré par règle directe. La colonne inverse signifie que l'opération a été réalisée sur l'image inversée (min-tree). Les sous-figures 4.56h à 4.56l de la figure 4.56 représentent l'extraction d'attribut sur une coupe.

Nom	Attribut	Valeurs	Inverse
Contrast α	C	$[10, 150]$	
Contrast β	C	$[10, 150]$	✓
Area α	A	$[2 \times 32^2, 1024^2]$	
Area β	A	$[64^2, +\infty]$	
Area γ	A	$[16^2, 512^2]$	✓

TABLE 4.6 – Liste des opérateurs connexes utilisés pour l'expérience.



Cette expérience a été répétée 10 fois, le jeu de données utilise l'image I_3 et l'expérience concerne les classes mitochondries et réticulum endoplasmique. Les figures 4.57

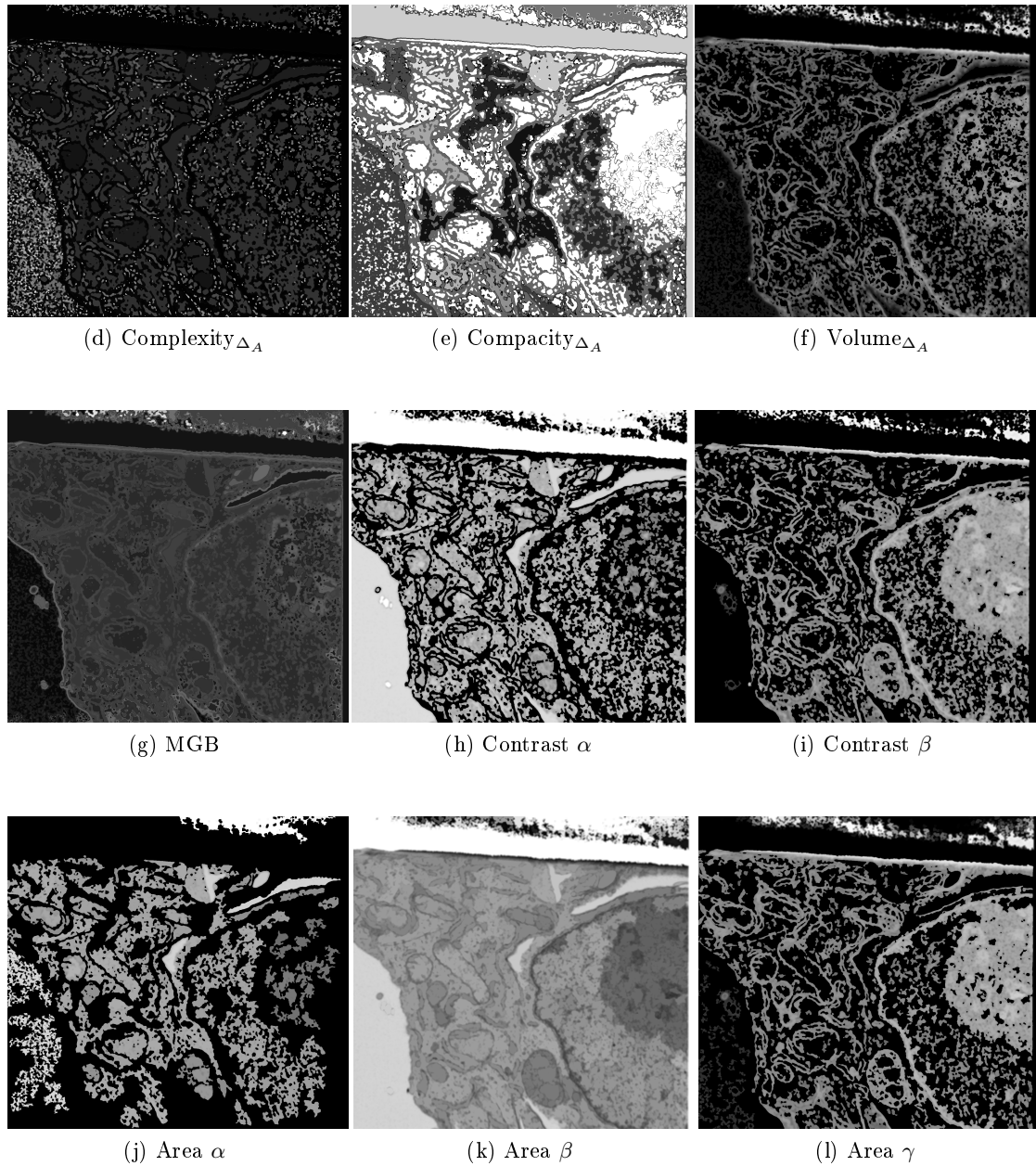


FIGURE 4.56 – Exemples des pré-traitements appliqués sur une coupe de l'image I_3 .

et 4.58 représentent les résultats de segmentation sur le jeu de test. Ces résultats ont fait l'objet d'une publication, et les figures sont directement tirées de l'article en question [Meyer et al., 2023].

Tout d'abord, on remarque que dans la plupart des cas, il n'y a pas de dégradation liée à l'ajout de pré-traitements. Pour la classe des mitochondries, ce sont les extrac-

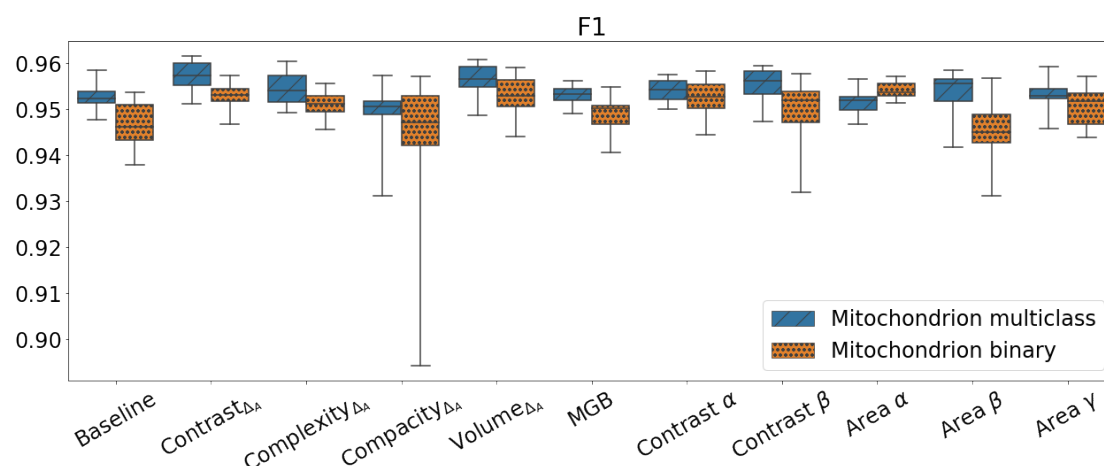


FIGURE 4.57 – Résultats de l'expérience utilisant l'extraction d'attributs, mitochondries, F1-Score.

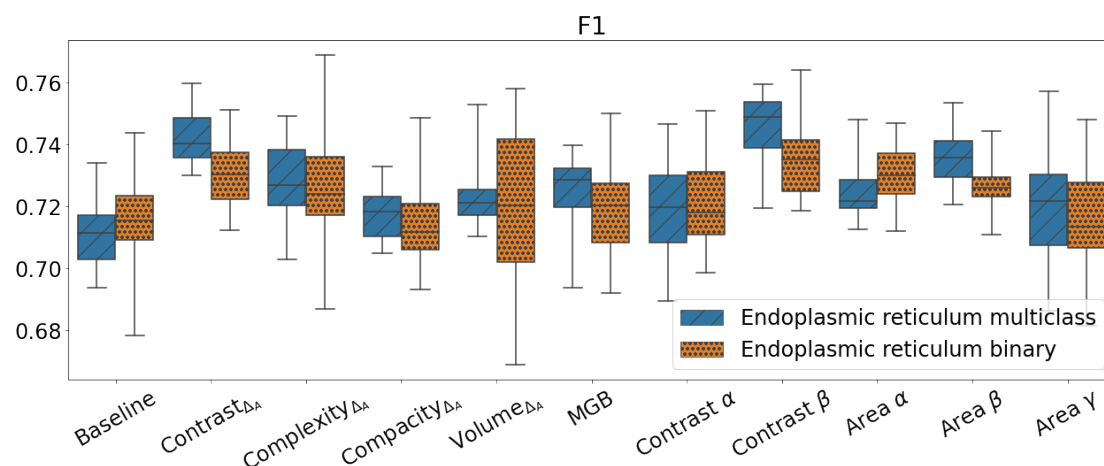


FIGURE 4.58 – Résultats de l'expérience utilisant l'extraction d'attributs, réticulum endoplasmique, F1-Score.

teurs $\text{Contrast}\Delta_A$ et $\text{Volume}\Delta_A$ qui donnent les meilleurs résultats, ainsi que le filtre $\text{Contrast}\beta$. Pour le réticulum endoplasmique, c'est $\text{Contrast}\Delta_A$ et $\text{Contrast}\beta$. Bien que les résultats visuels des filtres $\text{Contrast}\beta$ et $\text{Area}\gamma$ soient très proches, le second n'est clairement pas aussi efficace.

Dans la majorité des cas, l'approche multi-classes est plus performante que la méthode binaire. La différence est surtout visible pour les mitochondries.

Notre méthode permet donc d'améliorer les résultats de segmentations, mais au prix d'une extraction d'attributs. Si cette extraction n'est pas coûteuse en temps, le choix des attributs et des paramètres d'extraction pour notre image ne garantit pas une améliora-

tion similaire sur d'autres images. Au final, la différence de score est significative, mais reste faible et qualitativement (visuellement) négligeable.

4.4.7 Conclusion

Dans cette dernière partie, nous avons présenté nos contributions consistant à combiner apprentissage profond et morphologie mathématique. Nous avons pu remarquer que dans la plupart des cas, l'ajout en entrée du réseau de neurones d'image traitée par des filtres morphologiques ne permet pas d'amélioration des résultats de la segmentation. Nous avons également évalué l'utilisation de couches morphologiques dans nos modèles de segmentation, mais nous n'avons pas pu mettre en évidence d'amélioration des résultats en ajoutant différentes couches morphologiques aux réseaux. Enfin, nous avons présenté une méthode d'extraction d'attribut utilisant les structures hiérarchiques, permettant de pré-traiter nos images. Grâce à cette méthode, les pré-traitements appliqués aux images permettent d'améliorer la qualité des segmentations produites.

En l'état de nos connaissances, nous n'avons pas d'explications définitives expliquant la détérioration des résultats lorsque l'on utilise des images pré-traitées en plus des images originales en entrée de nos modèles. On pourrait initialement penser que l'ajout d'informations devrait dans le pire des cas, n'avoir aucun effet, comme c'est le cas dans notre communication. Nous pouvons cependant émettre des hypothèses.

Tout d'abord, lors de nos expériences, nous avons toujours intégré ces pré-traitements dès la première couche du réseau, la couche d'entrée combinant image originale et image pré-traitée. Ce faisant, il est envisageable qu'une modification faible des poids de la première couche entraîne une combinaison prématurée des images d'entrées. On peut supposer que les poids de ces premières couches ne varieraient alors pas suffisamment pour éliminer l'impact d'un pré-traitement négatif.

Ensuite, il est envisageable qu'en agrandissant la taille de l'entrée de notre modèle, nous augmentons aussi de manière significative la taille du problème d'optimisation, entraînant de ce fait une moins bonne optimisation des paramètres du réseau.

Enfin, dans le cas où nous n'améliorons pas les résultats, il est également possible qu'une sorte d'optimum ait déjà été atteint au vu des données d'apprentissage fournies pour l'entraînement du modèle.

Conclusion

5.1	Bilan	133
5.2	Perspectives	135
5.2.1	Annotations faibles et apprentissage profond	135
5.2.2	Représentation hiérarchique et apprentissage machine	136
5.2.3	Applications aux images non FIB-SEM	136
5.2.4	OpenData et base de données d'imagerie électronique cellulaire . .	137
5.2.5	Analyse morphologique de l'ultrastructure cellulaire	137

Dans ce dernier chapitre, nous rappelons les contributions majeures de nos travaux qui ont ouvert de nouvelles perspectives dans notre domaine de recherche.

5.1 Bilan

Dans cette thèse, nous avons étudié les méthodes de segmentation sémantique pour les images de microscopie électronique FIB-SEM cellulaire. La segmentation des images est un prérequis nécessaire afin de permettre une analyse quantitative et comparative de l'ultrastructure cellulaire. Nous nous sommes concentrés sur la segmentation des organelles, en apportant une attention particulière aux mitochondries et au réticulum endoplasmique. Nous avons également pris des dispositions pour rendre le résultat de nos travaux le plus aisé à diffuser et réutiliser. Premièrement, nous distribuons les sources de nos logiciels sur GitHub et nous séparons au maximum les fonctionnalités en modules indépendants, permettant leur utilisation dans d'autres contextes. Deuxièmement, nous proposons l'utilisation d'une configuration matérielle minimale correspondant à une station de travail classique. Notre objectif était de développer une approche permettant de segmenter précisément et automatiquement l'ultrastructure cellulaire à partir de ces images. Voici le rappel de nos contributions.

Tout d’abord, nous avons étudié l’utilisation des réseaux de neurones convolutifs (CNN) pour la segmentation d’images, en particulier des réseaux entièrement convolutifs (FCN) utilisant l’architecture U-Net. Nous avons cherché à adapter au mieux ces modèles et leurs méthodes d’entraînement aux images de microscopie électronique FIB-SEM cellulaire. Nous avons proposé des variations pour plusieurs paramètres :

1. l’utilisation de modèles multi-classes ou binaires ;
2. l’emploi de modèles 2D ou 3D ;
3. les variations de composition des couches pour l’architecture U-Net ;
4. le choix de la taille des patchs pour l’entraînement ;
5. la taille du jeu de données employé pour l’entraînement ;
6. le choix de la fonction de perte utilisée pour l’optimisation ;
7. l’utilisation de modèles pré-entraînés.

Nous avons pu définir quels paramètres sont importants ainsi que leurs valeurs optimales dans notre contexte. Nous avons en l’occurrence mis en évidence un résultat contraire à ce qu’indiquait alors l’état de l’art : les architectures 2D sont plus performantes que leurs variations 3D, avec de petites bases de données d’images et une configuration matérielle limitée.

Nos modèles ont démontré leur efficacité pour la segmentation des différentes structures cellulaires : les mitochondries, le réticulum endoplasmique, le noyau, la membrane cellulaire ou encore les endosomes. Nos travaux ont été valorisés dans la communication suivante : [Meyer et al., 2021] ; dans laquelle nous avons également mis en évidence l’absence d’amélioration lors de l’utilisation de pré-traitement morphologique.

À la suite de ces premières contributions et durant toute notre étude, de nouveaux travaux sont venus enrichir l’état des connaissances du domaine. On peut en particulier évoquer le projet OpenOrganelle [Heinrich et al., 2021]. Une première version de cet article « Automatic whole cell organelle segmentation in volumetric electron microscopy », a été mise en ligne en novembre 2020. Une partie des réponses aux questions que nous nous étions posées furent confirmées. Une autre partie fut remise en cause, c’est par exemple la raison pour laquelle nous avons répété nos expériences sur les modèles 2D et 3D qui ont confirmé nos premières conclusions.

Suite à nos premiers travaux, nous avons identifié deux points d’amélioration sur lesquels nous souhaitons travailler.

Le premier point d’amélioration concerne l’étape de création de segmentation de référence, la phase la plus coûteuse. En l’absence de généralisation satisfaisante des modèles entraînés, nous souhaitons réduire le temps d’annotation nécessaire à la production d’une segmentation fiable. Deuxièmement, l’application de pré-traitements morphologiques, qu’ils soient classiques ou utilisant des opérateurs connexes, ne permettait pas d’améliorer la qualité des segmentations. Notre second objectif fut alors d’améliorer nos segmentations en nous inspirant d’autres outils utilisant la morphologie mathématique.

Tout d’abord, pour réduire le temps nécessaire à la création de données pour l’entraînement de nos modèles, nous avons étudié les modèles interactifs et les modèles entraînés à l’aide de segmentations de référence faibles.

Nous avons proposé l'utilisation de modèles interactifs. Nos expérimentations sur cette méthode ont montré qu'il est possible d'entraîner un modèle interactif avec très peu de données, tout en ayant de bonnes performances. Ce modèle peut ensuite être utilisé interactivement afin d'accélérer le processus de création de nouvelles segmentations de référence, et d'entraîner un modèle automatique plus performant.

Nous avons également proposé une seconde approche, utilisant des segmentations faibles pour entraîner notre réseau de neurones. Cette méthode permet d'entraîner un modèle en utilisant des annotations nécessitant moins de temps à être produites par l'expert humain. Une part importante du temps nécessaire à l'annotation manuelle est liée à la difficulté de prendre une décision pour certains objets. En plus de réduire le temps d'annotation lié au remplissage et à la délimitation précise des objets, cette méthode permet aussi de soulager les situations où la décision est difficile à prendre. L'expert annoté uniquement une partie des données, afin de ne pas perdre de temps dans les cas incertains. Les segmentations que nous avons pu obtenir avec cette méthode ne sont pas parfaites, mais très encourageantes pour le noyau, la membrane cellulaire et les mitochondries.

Enfin, nous avons développé une méthode mélangeant apprentissage profond et morphologie mathématique à l'aide d'une extraction d'attributs basée sur le max-tree. Premièrement, nous avons proposé un algorithme général d'extraction d'attributs pouvant s'adapter aux structures hiérarchiques. Ensuite, nous avons développé un attribut adapté à la détection des composantes connexes dans le contexte d'une de nos images. Finalement, nous avons évalué les différences de performances de modèles utilisant ces extractions d'attributs, et mis en évidence plusieurs cas où une amélioration est présente. Ces travaux représentent la première étape vers une amélioration de la qualité des segmentations à l'aide de pré-traitements morphologiques. Ces travaux ont été valorisés au travers d'une communication scientifique [Meyer et al., 2023].

Cette thèse a apporté des contributions dans le domaine de l'analyse d'images de microscopie électronique cellulaire. Nos travaux ont permis de réaliser des avancées dans la segmentation des images, étape nécessaire afin d'obtenir une analyse quantitative de l'ultrastructure cellulaire.

5.2 Perspectives

Notre étude nous permet d'envisager plusieurs perspectives que nous présentons dans cette partie.

5.2.1 Annotations faibles et apprentissage profond

Lors de nos expériences sur les annotations faibles non simulées, détaillées dans la sous-section 4.3.3, nous avons montré qu'un modèle entraîné sur des annotations faibles produites par un expert permet de générer une segmentation correcte, mais imparfaite.

Premièrement, il serait intéressant d'évaluer la robustesse des résultats de la méthode en répétant ces expériences plusieurs fois, et sur d'autres images. On pourrait alors évaluer

quantitativement la réduction de qualité de la segmentation par rapport à un modèle automatique.

Ensuite, on pourrait étudier la robustesse de la méthode aux variations d’annotations en demandant à l’expert d’annoter les images de différentes manières (contours plus ou moins précis, densités variables en fonction des organelles), dans le but d’étudier les effets de ces variations sur nos modèles. Une telle étude permettrait de définir la meilleure méthode d’annotation des données, afin de maximiser les performances des modèles tout en minimisant le temps requis pour la création des données.

Enfin, il serait intéressant de se pencher sur les méthodes de segmentation interactive, notamment les méthodes qui ne demandent pas un entraînement initial, et qui pourraient alors accélérer ce processus d’annotation faible.

5.2.2 Représentation hiérarchique et apprentissage machine

Dans la sous-section 4.4.6, nous avons détaillé notre méthode d’extraction d’attributs utilisée dans le but d’enrichir l’entrée de nos modèles de segmentation. Notre idée initiale de pré-traitement des images à l’aide d’opérateurs morphologiques a évolué jusqu’à devenir une méthode générique d’extraction d’attributs utilisant la représentation hiérarchique de l’image. Notre méthode d’extraction d’attribut pourrait être utilisée avec toutes les méthodes où un classifieur utilise une extraction des caractéristiques de l’image, par exemple, les forêts d’arbres décisionnels.

Ensuite, nous pourrions utiliser ces classifieurs afin de détecter automatiquement et plus rapidement les pré-traitements intéressants pour une image donnée. Dans leur article, les auteurs de [Farfan Cabrera et al., 2021a] utilisent des forêts d’arbres décisionnels dans le but de calculer l’importance relative de chacun des attributs. On pourrait réutiliser cette idée dans le but d’évaluer de manière systématique des plages plus larges d’attributs et de paramètres, pour les présélectionner.

Enfin, plutôt que d’utiliser nos pré-traitements uniquement en entrée du réseau de neurones, il pourrait être intéressant d’intégrer ces pré-traitements à d’autres niveaux, plus profond de l’encodeur ou dans la partie décodeur du modèle.

5.2.3 Applications aux images non FIB-SEM

Nous avons développé des méthodes de segmentation générales, mais adaptées à notre contexte, donc spécialisées pour les images FIB-SEM. Cependant, il est tout à fait envisageable d’adapter ces méthodes à de nouveaux types d’images, en particulier d’autres images de microscopie électronique cellulaire. Dans le cadre de notre collaboration avec le GoetzLab, nous avons eu accès à des images FIB-SEM, mais la majorité des images en leur possession sont de type ssTEM (Serial Sectioning for Transmission Electron Microscopy, détails dans la section 3.1). Une grande partie de nos travaux devrait pouvoir être réutilisée pour cette autre modalité. Plus spécifiquement, notre approche se reposant sur des modèles 2D, la grande différence d’anisotropie dans l’axe des coupes ne devrait pas avoir d’impact négatif sur nos méthodes. Néanmoins, il y a une différence fondamentale entre les deux processus d’acquisition. En microscopie à balayage, c’est la surface de

l'échantillon que l'on capture, alors qu'en microscopie à transmission, c'est une projection de la tranche imagée. Des adaptations seront nécessaires ainsi qu'une recherche des paramètres d'apprentissage optimaux, mais une grande partie de nos travaux pourra être réutilisée.

5.2.4 OpenData et base de données d'imagerie électronique cellulaire

Un des problèmes que nous avons rencontré pour entraîner nos modèles est qu'une grande quantité de données annotée est requise. C'est un manque de variations dans les images annotées que nous possédons qui est en partie à l'origine de notre incapacité à entraîner un modèle robuste et performant sur de nouvelles images. Cependant, il n'est pour l'instant pas envisageable de créer en interne un jeu de données suffisamment conséquent, composé de milliers d'images.

Nous avons fait un inventaire dans la section 3.1 des données publiques que nous avons identifiées et qui pourraient être utilisées dans le but d'entraîner ou de pré-entraîner des modèles. Parmi ces jeux de données, le jeu de données CEM-MitoLab est particulier, car il est principalement composé d'images et d'annotations obtenues publiquement et compilées dans un format commun.

Nous avons mis en ligne une partie de nos données, disponibles pour la recherche. Les images *I3*, *LW4-1* et leurs annotations utilisées dans nos travaux sont disponibles sur la plateforme Zenodo.

— *I3* : <https://zenodo.org/record/8341172>

— *LW4-1* : <https://zenodo.org/record/8344292>

La généralisation de ce partage de données et de l'OpenData au sein des différents laboratoires producteurs et utilisateurs de données pourrait à l'avenir améliorer de manière significative la quantité de données utilisables et à moyen terme permettre l'entraînement de modèles plus robustes grâce à la mise en commun des ressources.

5.2.5 Analyse morphologique de l'ultrastructure cellulaire

L'objectif de la segmentation est de permettre une analyse morphologique quantitative de l'ultrastructure cellulaire. Nos contributions ouvrent de nouvelles perspectives pour la compréhension des processus cellulaires et intracellulaires. La production automatique de segmentation de qualité nous permet d'envisager l'utilisation de ces segmentations pour effectuer une analyse des caractéristiques morphologiques des cellules, mais également des analyses comparatives. On peut par exemple envisager de comparer différentes cellules dont une partie aurait été altérée et ainsi mettre en évidence et quantifier les différences morphologiques résultantes. Ces perspectives offrent ainsi un potentiel considérable pour les recherches futures dans le domaine de la biologie cellulaire et de la recherche biomédicale.

Annexe

A.1	Productions logicielles	140
A.1.1	Segmentation d'images de microscopie électronique	140
A.1.2	Segmentation interactive et annotations faibles	143
A.1.3	Morphologie mathématique	145
A.1.4	Visualisation	147
A.1.5	Divers	148
A.1.6	Résumé des applications	151
A.2	Jeux de données	152
A.2.1	I3-ISBI et LW4-ISBI	152
A.2.2	IGBMC-S-multi	152
A.2.3	IGBMC-620-MITO	152
A.2.4	I3-ISO	153
A.2.5	LW4-ISO	153
A.2.6	JG1-Mito 5x384	153
A.2.7	MitoEM-H	154
A.3	Images FIB-SEM IGBMC	155
A.4	Figures	161

A.1 Productions logicielles

Tout au long de cette thèse, des solutions logicielles ont dû être conçues afin de réaliser nos expérimentations. Ces logiciels ont été développés de manière à permettre la production de nos expériences, la reproduction de nos résultats, mais également une réutilisation plus générale par la communauté. Dans ce chapitre, nous listons ces productions logicielles, disponibles en accès libre sur GitHub : <https://github.com/Cyril-Meyer/>.

A.1.1 Segmentation d’images de microscopie électronique

Cette première partie concerne les logiciels utilisés afin de permettre la segmentation d’images.

NeNIST

- <https://github.com/Cyril-Meyer/NeNIST>
- Python

NeNIST (Neural Network Image Segmentation toolkit for Electron Microscopy) est le projet regroupant les différents modules créés afin d’entraîner nos modèles. Cette bibliothèque, regroupe les scripts d’entraînement générique, les outils permettant le calcul des métriques, les fonctions de pertes, les modules de préparations et de chargement des données.

```
1 import utils.parser
2 import utils.gpu
3
4 import data.patch
5 import data.distance_transform
6
7 import models.unet
8
9 import loss.segmentation
10
11 # import datasets.I3
12 import datasets.reader
13
14 import metrics.segmentation
```

FIGURE A.1 – Exemple d’importation de modules NeNIST pour un script d’entraînement et d’évaluation.

Cutter

Le projet cutter est un sous-projet de la boîte à outils NeNIST. Nous avons déjà évoqué ce logiciel dans la sous-section 4.3.2, c’est un outil interactif de segmentation.

Le logiciel permet de segmenter des images ou des parties d'images à l'aide de modèles automatiques ou interactifs.

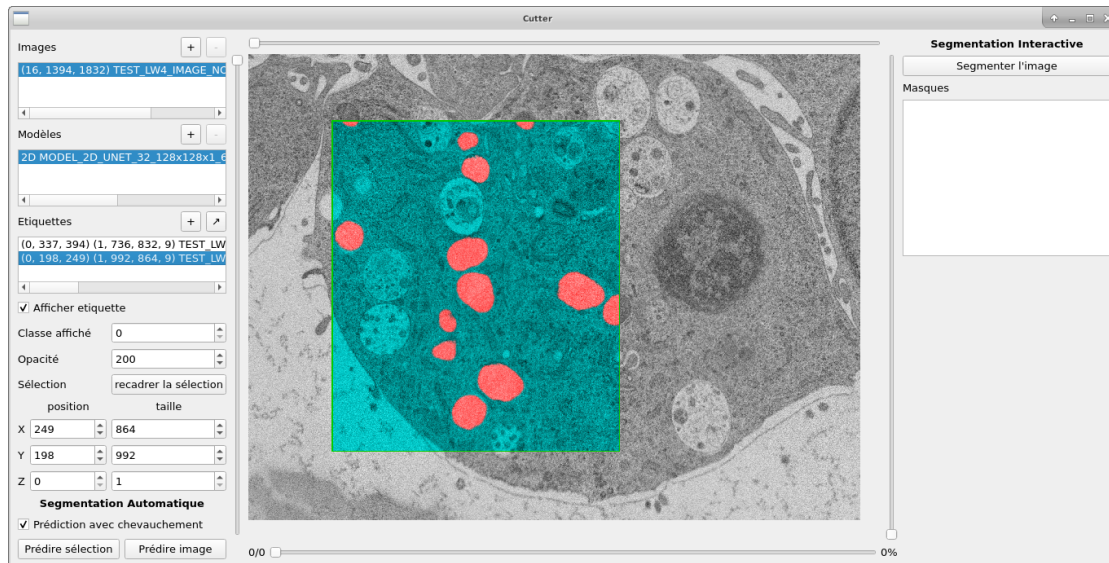


FIGURE A.2 – Exemple d'utilisation de l'application Cutter sur l'image LW_4 avec un modèle de segmentation automatique.

Une démonstration vidéo de ces fonctionnalités est disponible à l'adresse suivante : <https://youtu.be/3v9ET3jORFc>. Comme la vidéo contient peu de montage, il est conseillé de passer rapidement les moments de prédiction qui sont en temps réel.

TISM

- <https://github.com/Cyril-Meyer/tism>
- Python

La bibliothèque TISM (TensorFlow Image Segmentation Models) permet de créer et d'utiliser des modèles de segmentation TensorFlow. L'idée de la bibliothèque est de construire des modèles de segmentation d'image de manière procédurale avec un principe de séparation entre architecture et backbone (brique de construction).

Une architecture représente un enchaînement particulier de blocs élémentaires. Lorsqu'une architecture demande la construction d'un bloc élémentaire, elle lui fournit un unique paramètre, représentant le niveau correspondant à la « profondeur » du bloc. Dans les cas que nous avons implémentés, cela sert à définir le nombre de filtres du bloc en fonction de la profondeur.

Il est aisé d'implémenter une nouvelle architecture ou de nouvelles briques de constructions. Les blocs peuvent être imbriqués les uns dans les autres. Un bloc de traitement avec connexion résiduelle est construit à partir d'une brique de construction classique imbriquée dans un bloc résiduel.

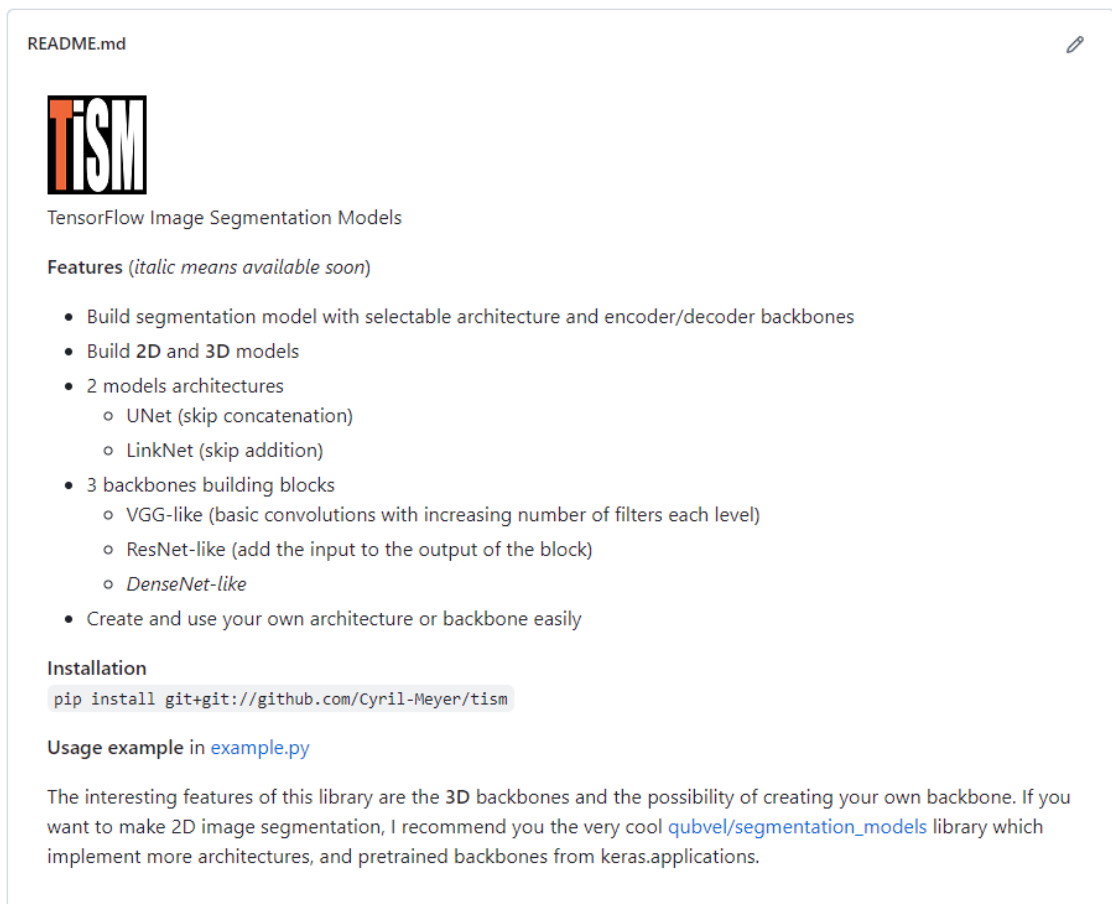


FIGURE A.3 – Page de présentation de la bibliothèque TISM.

LII

- <https://github.com/Cyril-Meyer/lii>
- Python

La bibliothèque LII (Large Image Inference) permet de simplifier la segmentation d'images de grandes dimensions en proposant une fonction paramétrable pour segmenter l'image à partir de patches. L'utilisateur donne en entrée une image (2D ou 3D), une fonction de prédiction et choisit la taille de patch à utiliser (2D ou 3D) ainsi que le chevauchement à appliquer. La bibliothèque est par exemple utilisée dans notre logiciel cutter, lorsque l'on souhaite segmenter toute l'image.



(a)



(b)

FIGURE A.4 – Exemple d’affichage de la bibliothèque LII lors de l’inférence sur une image de grande dimension.

A.1.2 Segmentation interactive et annotations faibles

Cette seconde partie concerne les logiciels conçus pour permettre les expériences de segmentation interactive et d’annotations faibles.

ISISIM

- <https://github.com/Cyril-Meyer/ISISIM>
- Python

ISISIM (Interactive Segmentation Interaction SIMulation) est un module de simulation d’interaction utilisateur pour la segmentation. Le module a particulièrement été utile pour nos expériences de segmentation interactive. Des exemples animés des interactions sont disponibles sur le dépôt GitHub.

SegmentationToScribble

- <https://github.com/Cyril-Meyer/SegmentationToScribble>
- Python

SegmentationToScribble est une bibliothèque logicielle de simulations d’annotations faibles à partir de segmentations denses. Historiquement, c’est l’ancêtre du projet ISISIM, la plupart des fonctionnalités ayant été portées et améliorées. Des exemples animés des résultats sont disponibles sur le dépôt GitHub.

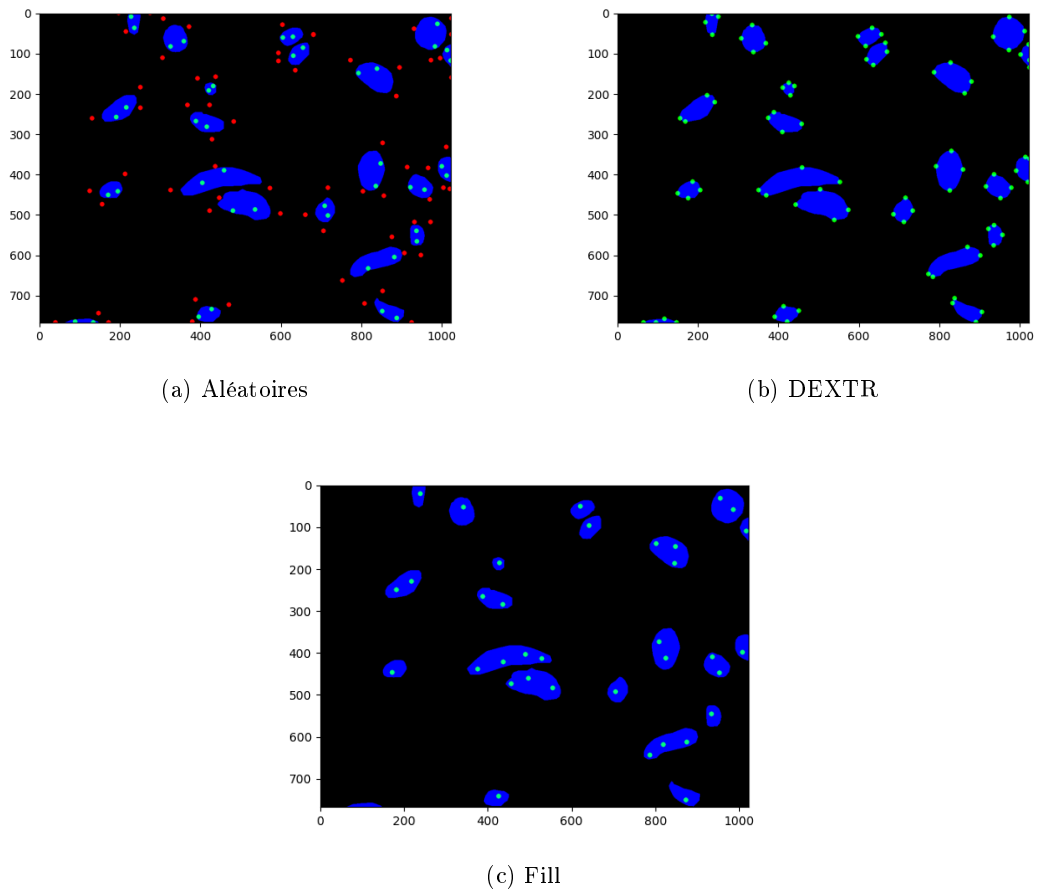


FIGURE A.5 – Exemples d’interactions simulées par ISISIM.

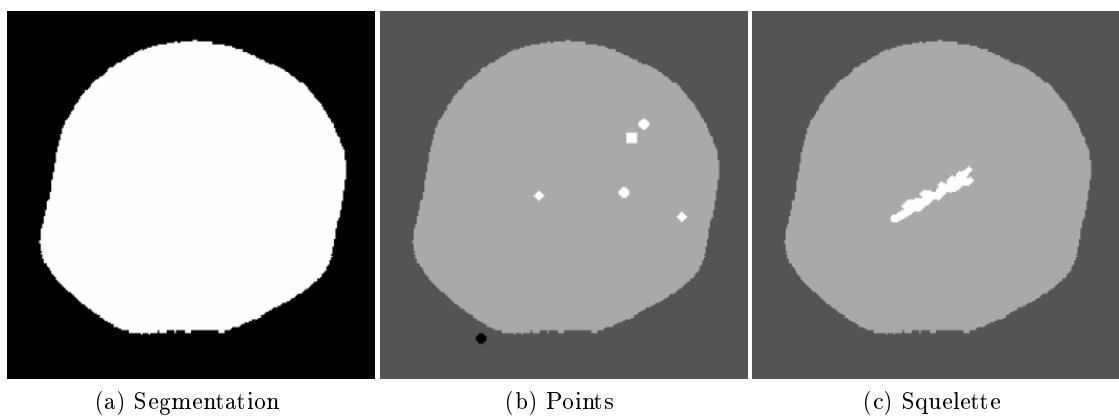


FIGURE A.6 – Exemples d’annotations faibles simulées par SegmentationToScribble.

A.1.3 Morphologie mathématique

LibTIM

- <https://github.com/Cyril-Meyer/libtim>
- C++

LibTIM est une bibliothèque logicielle de traitement d'image développée par Benoît Naegel (version originale <https://github.com/bnaegel/libtim>). La bibliothèque implémente les fonctions nécessaires au calcul et à l'utilisation de structures comme le max-tree. Dans le cadre de nos travaux, nous avons modifié la bibliothèque afin d'ajouter le calcul de nouveaux attributs pour les structures hiérarchiques. Une interface graphique, GUILibTIM, permettant d'analyser des images a également été ajoutée dans le but de mieux identifier les effets des calculs d'attributs sur nos images.

Une démonstration vidéo de cette interface graphique est disponible à l'adresse suivante : <https://youtu.be/jNJtAiE-K0w>.

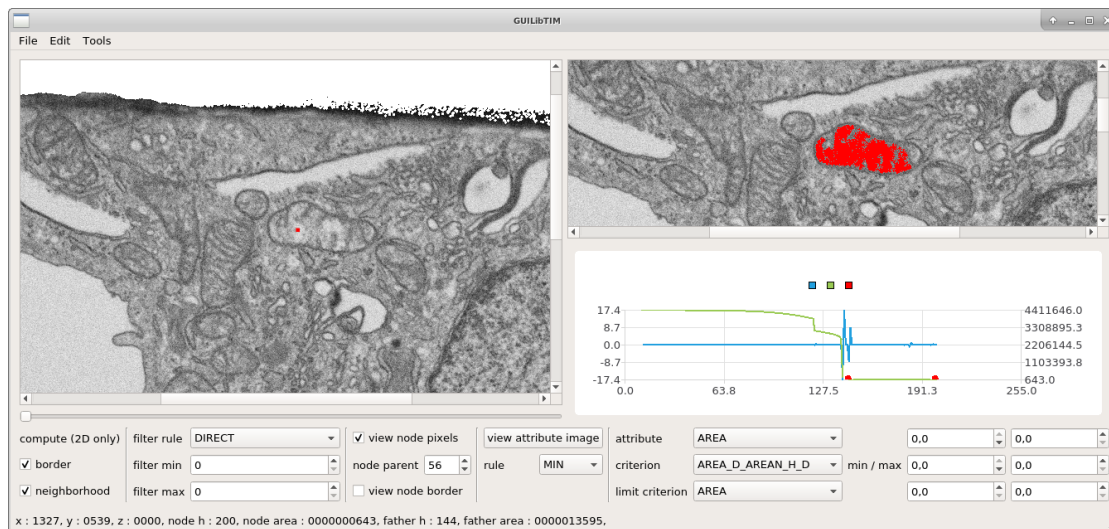


FIGURE A.7 – Exemple d'analyse de l'image I_3 avec l'interface graphique GUILibTIM.

PyLibTIM

- <https://github.com/Cyril-Meyer/PyLibTIM>
- Python / C++

PyLibTIM est une interface logicielle pour utiliser certaines fonctionnalités de la bibliothèque LibTIM en Python. L'interface Python / C++ est mise en place grâce à la bibliothèque pybind11. Les fonctionnalités implémentées actuellement permettent l'extraction d'attributs et l'utilisation d'opérateurs connexes utilisant l'arbre des coupes.

Les fonctionnalités de cette bibliothèque sont vouées à être remplacées par un ajout de notre algorithme d'extraction d'attribut dans la bibliothèque Higma [Perret et al.,

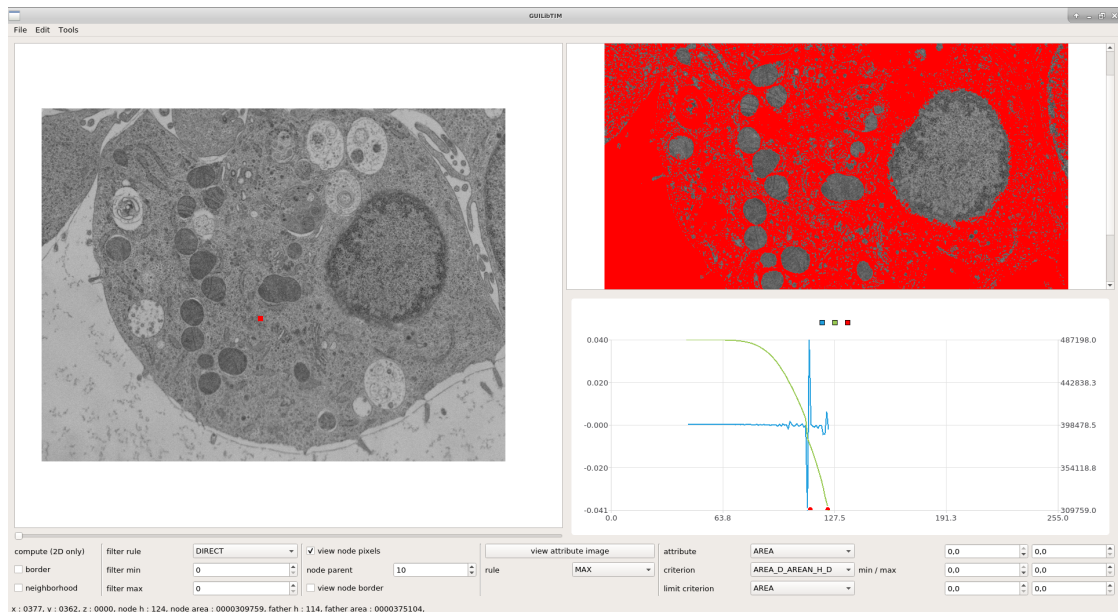


FIGURE A.8 – Exemple d’analyse de l’image LW_4 avec l’interface graphique GUILibTIM.

2019], cette dernière implémentant d’ores et déjà les différents opérateurs connexes et structures hiérarchiques.

CTAISegmentationCNN

- <https://github.com/Cyril-Meyer/DGMM2022-RRPR-CTAISegmentationCNN>
- Python / C++

Le projet CTAISegmentationCNN est issu de nos travaux pour le volet spécial RRPR de la conférence DGMM 2022. Ce dépôt contient l’intégralité des sources et données pour la reproduction des résultats de notre article [Meyer et al., 2023].

Le projet est composé d’une version simplifiée de LibTIM pour le traitement de nos images (<https://github.com/Cyril-Meyer/DGMM2022-RRPR-ComponentTreeAttributeImage>) ainsi que des images et données d’entraînement permettant la reproduction (<https://github.com/Cyril-Meyer/DGMM2022-RRPR-MEYER-DATA>).

MaxTreePy

- <https://github.com/Cyril-Meyer/MaxTreePy>
- Python

MaxTreePy est une bibliothèque Python pour le calcul de max-tree et le filtrage par opérateurs connexes.

A.1.4 Visualisation

VREMS

- <https://github.com/Cyril-Meyer/VREMS>
- Python / GLSL

VREMS (Volume Rendering toolkit for Electron Microscopy Segmentations) est un outil de visualisation et de création de vidéos d'images et de segmentation volumétriques.

Cet outil a été créé en collaboration avec Paul Viville, docteur en informatique et chercheur postdoctoral dans l'équipe Informatique Géométrique et Graphique du laboratoire ICube. Ce dernier était en charge du développement des shaders pour le rendu.

Des vidéos de résultats sont disponibles sur le dépôt GitHub et aux adresses suivantes : <https://www.youtube.com/watch?v=t5KbPGDv9hs>, <https://www.youtube.com/watch?v=YBD-Z2qRG-g>.

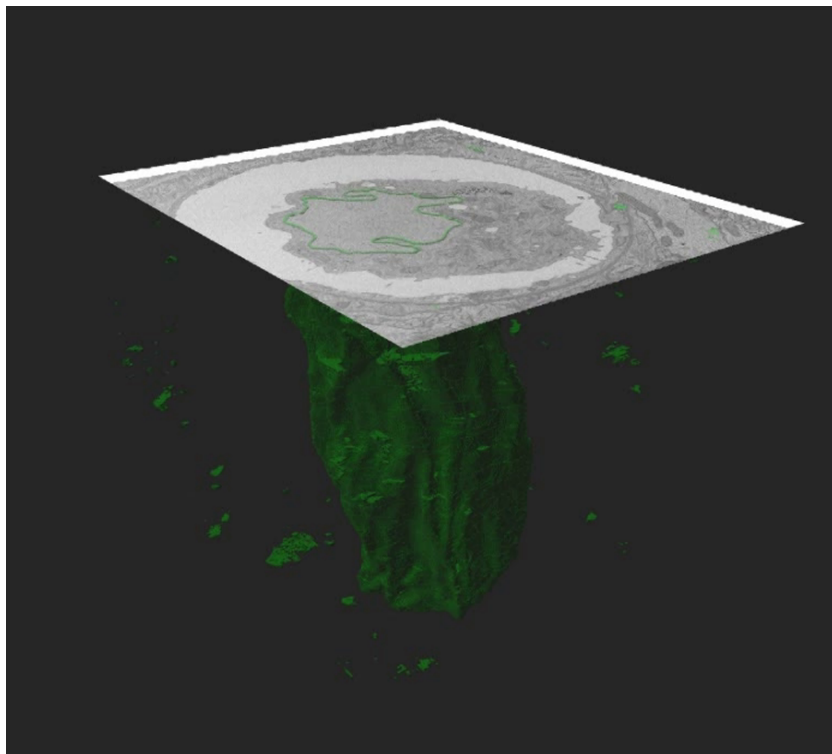
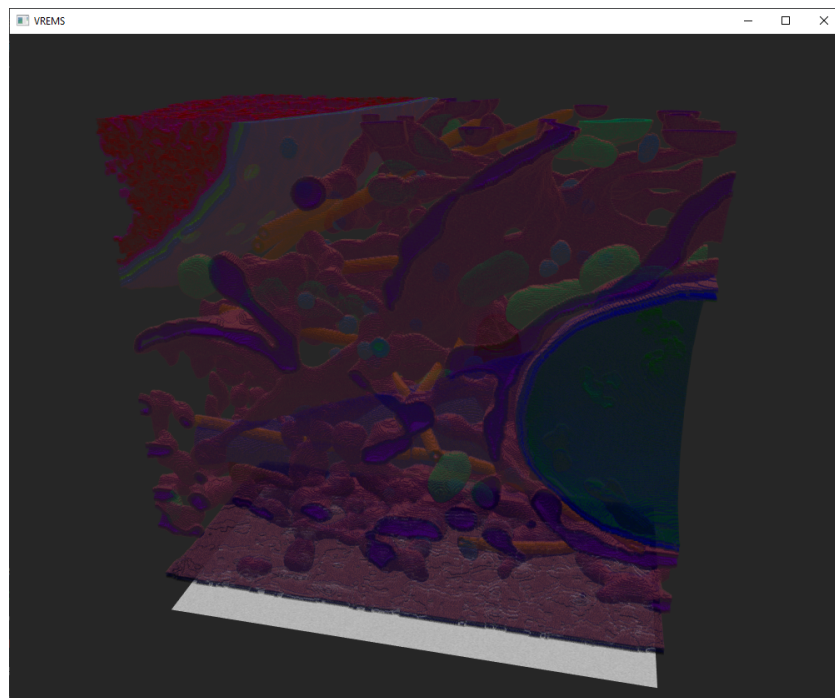
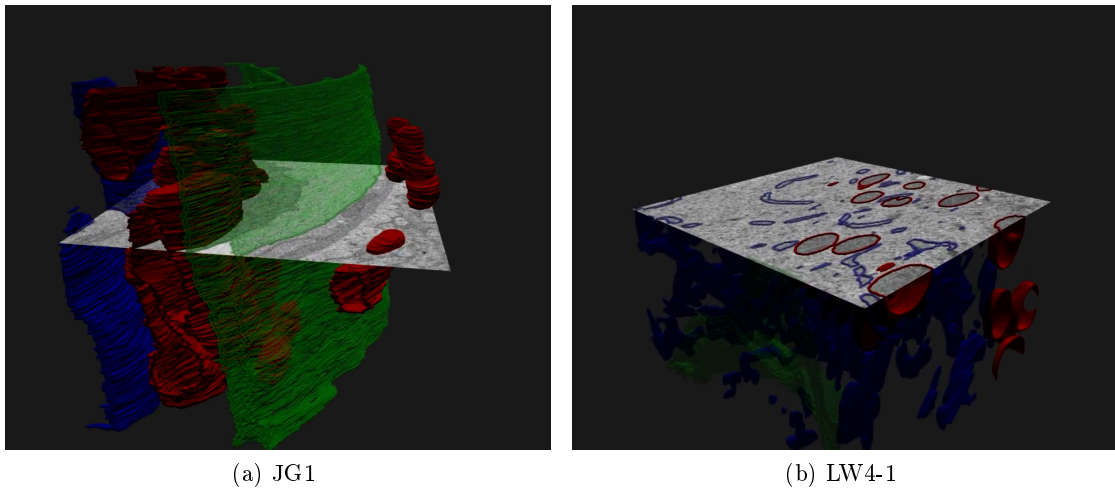


FIGURE A.9 – Exemple de visualisation d'une prédiction générée par VREMS sur une image du GoetzLab.



(c) jrc_hela-2 (crop 113)

FIGURE A.10 – Exemples de visualisation générée par VREMS.

A.1.5 Divers

NumPyDraw et NumPyRandomShapes3D

- <https://github.com/Cyril-Meyer/NumPyRandomShapes3D>
- <https://github.com/Cyril-Meyer/NumPyDraw>

— Python

NumPyRandomShapes3D est une bibliothèque logicielle pour la génération d'images volumétriques aléatoires paramétrables pour les tableaux NumPy [Harris et al., 2020]. L'image générée est composée d'ellipsoïdes déformées et texturées.

NumPyDraw est une bibliothèque logicielle pour le dessin de formes dans des tableaux NumPy, et une dépendance de NumPyRandomShapes3D. NumPyDraw implémente les ellipses (2D) et ellipsoïdes (3D) avec rotations, fonction absente du module `skimage.draw` [Van der Walt et al., 2014].

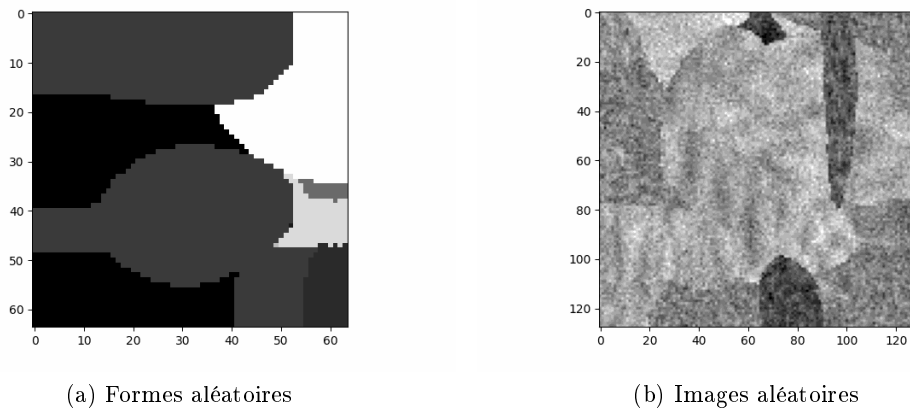


FIGURE A.11 – Exemples d'images aléatoires générées avec NumPyRandomShapes3D.

SeSeCoCo

— <https://github.com/Cyril-Meyer/SeSeCoCo>

— Python

SeSeCoCo (Semantic Segmentation Comparator using Components) est un outil de comparaison de segmentations sur les composantes connexes. L'outil permet de rendre compte du nombre de composantes connexes correctement détectées ou non.

Un exemple de résultat d'une analyse utilisant SeSeCoCo :

```
SeSeCoCo : Semantic Segmentation Comparator using Components
Segmentation 1 (y_true)
Connected Components 12
Correctly detected components 8
False negative components 1
Under detected components (too small or not enough recall) 3
Segmentation 2 (y_pred)
Connected Components 10
False positive components 1
```


Over detecting components (too large or not enough precision) 2

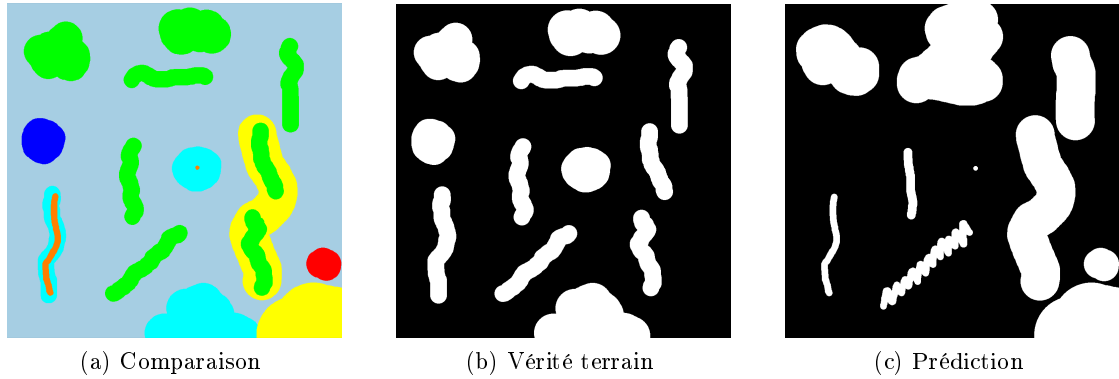


FIGURE A.12 – Exemple de comparaison faite avec SeSeCoCo. Les seuils de précision / rappel sont fixés à 0,40. Vert : TP, Rouge : FP, Bleu : FN, Jaune : sur-détection, Cyan : sous-détection, Orange : partie détectée d'un objet sous-déecté.

DistanceMap

- <https://github.com/Cyril-Meyer/DistanceMap>
- Python

DistanceMap est un outil de calcul de carte de distance, développé afin de reproduire les expériences des auteurs de [Xu et al., 2016] sur nos données. L'outil a été développé dans l'objectif de calculer des cartes de distance à partir de quelques points positifs et négatifs.

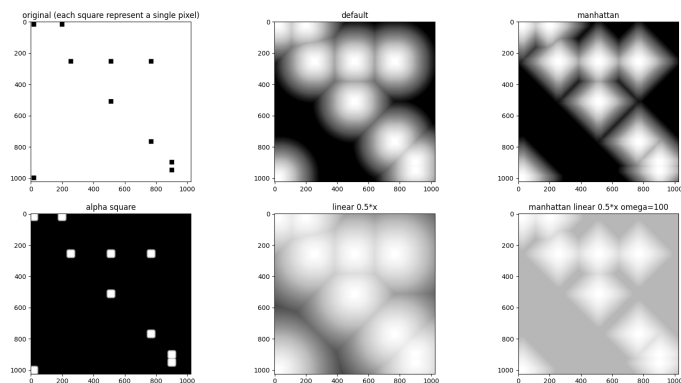


FIGURE A.13 – Exemple de calcul de carte de distance avec DistanceMap.

A.1.6 Résumé des applications

Nom	Description
NeNIST	Boîte à outils pour la segmentation d'images de microscopie électronique. Génération de patches, augmentation de données, entraînement et évaluation de modèles.
TISM	Générateur de modèles TensorFlow avec choix d'architecture générale (e.g. UNet, LinkNet) et de backbone (e.g. VGG, ResNet).
LII	Inférence automatique sur de grandes images avec patches recouvrants.
ISISIM	Simulateur d'interactions utilisateur pour les algorithmes de segmentation interactifs.
LibTIM	Extension de la bibliothèque LibTIM développée par Benoît Naegel. Ajouts de nouveaux attributs pour l'arbre des coupes et d'une interface graphique.
PyLibTIM	Bibliothèque python qui permet l'utilisation de certaines fonctionnalités de LibTIM en Python.
CTAISegmentationCNN	Dépôt principal pour notre article combining max-tree and CNN for segmentation of cellular FIB-SEM images.
VREMS	Outils de rendu volumique pour les segmentations de microscopie électronique.
NumPyDraw	Génération de forme pour tableau NumPy, extension de type skimage.draw pour les ellipsoïdes.
NumPyRandomShapes3D	Génération de tableau NumPy 3D avec des formes et textures aléatoires.

TABLE A.1 – Liste des productions logicielles open source principales.

A.2 Jeux de données

A.2.1 I3-ISBI et LW4-ISBI

- Images
 - *I3* (79 tranches, 171 → 250)
 - *LW4-1* (80 tranches, 0 → 40 + 80 → 120)
- Classes
 - mitochondrie
 - réticulum endoplasmique
 - membrane cellulaire
 - membrane nucléaire (I3-ISBI)
 - endosome (LW4-ISBI)
- Répartition
 - entraînement : 40 tranches
 - validation : 20 tranches
 - test : 19 tranches (I3-ISBI), 20 tranches (LW4-ISBI)

A.2.2 IGBMC-S-multi

- Images
 - *I3* (80 tranches, 171 → 251)
 - *LW4-1* (80 tranches, 0 → 40 + 80 → 120)
- Classes
 - mitochondrie
 - réticulum endoplasmique
- Répartition
 - entraînement : 2 × 40 tranches
 - validation : 2 × 20 tranches
 - test : 2 × 20 tranches

A.2.3 IGBMC-620-MITO

- Images
 - *I3* (500 tranches, 0 → 500)
 - *LW4-1* (120 tranches, 0 → 40 + 80 → 120 + 160 → 200)
- Classes
 - mitochondrie

- Répartition
 - entraînement : 600 tranches
 - validation : 2 10 tranches
 - test : 10 tranches

A.2.4 I3-ISO

- Images
 - *I3* (250 tranches, $0 \rightarrow 250$) rendus isotropes par interpolation sur l'axe Z (1000 tranches)
- Classes
 - mitochondrie
- Répartition
 - entraînement : 400 tranches (800 isotropes)
 - validation : 50 tranches (100 isotropes)
 - test : 50 tranches (100 isotropes)

A.2.5 LW4-ISO

- Images
 - *LW4-1* (80 tranches, $0 \rightarrow 40 + 80 \rightarrow 120$) rendus isotropes par interpolation sur l'axe Z (160 tranches)
- Classes
 - mitochondrie
 - réticulum endoplasmique
 - endosome
 - membrane cellulaire
- Répartition
 - entraînement : 40 tranches (80 isotropes)
 - validation : 20 tranches (40 isotropes)
 - test : 20 tranches (40 isotropes)

A.2.6 JG1-Mito 5x384

- Images
 - *JG1* (5 blocs de $384 \times 384 \times 384$)
- Classes
 - mitochondrie
- Répartition
 - entraînement : 4 blocs (1536 tranches)
 - validation : 1/2 bloc (192 tranches)
 - test : 1/2 bloc (192 tranches)

A.2.7 MitoEM-H

- Images
 - *MitoEM - H* (500 tranches)
- Classes
 - mitochondrie
- Répartition
 - entraînement : 400 tranches
 - validation et test : 100 tranches communes

A.3 Images FIB-SEM IGBMC

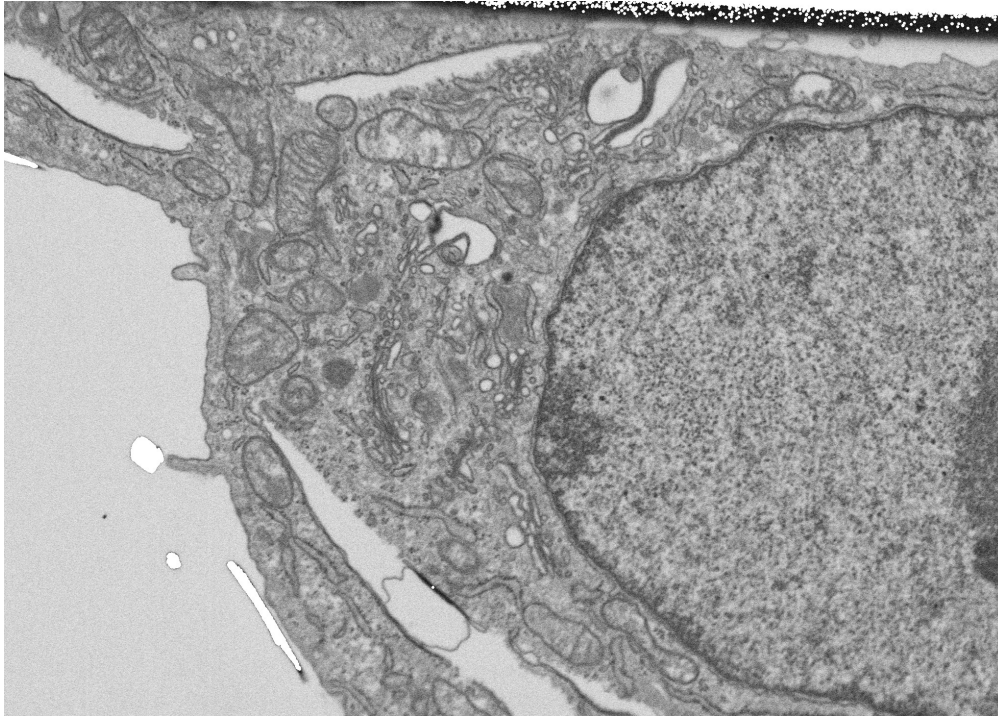


FIGURE A.14 – Image I_3 , coupe 001.

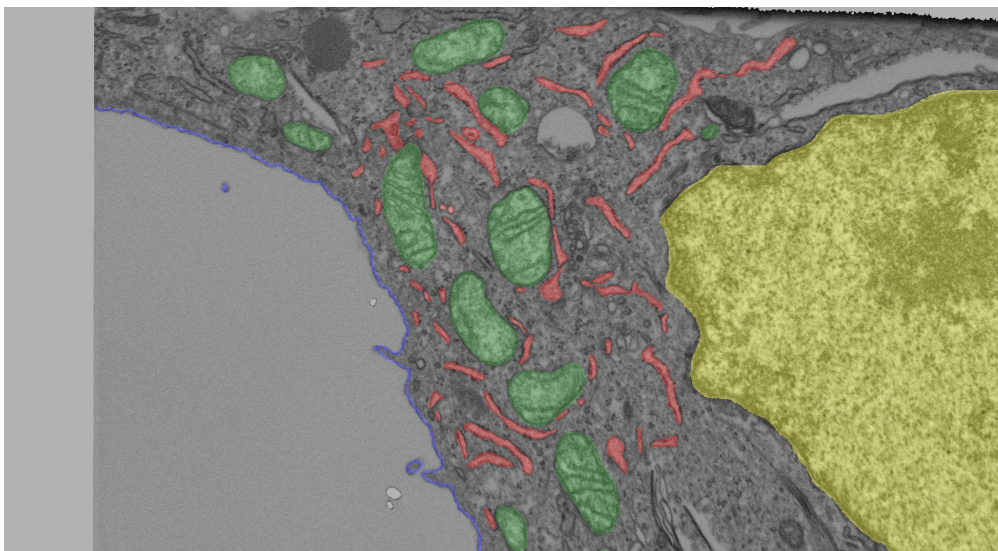


FIGURE A.15 – Image I_3 avec segmentation de référence, coupe 180.

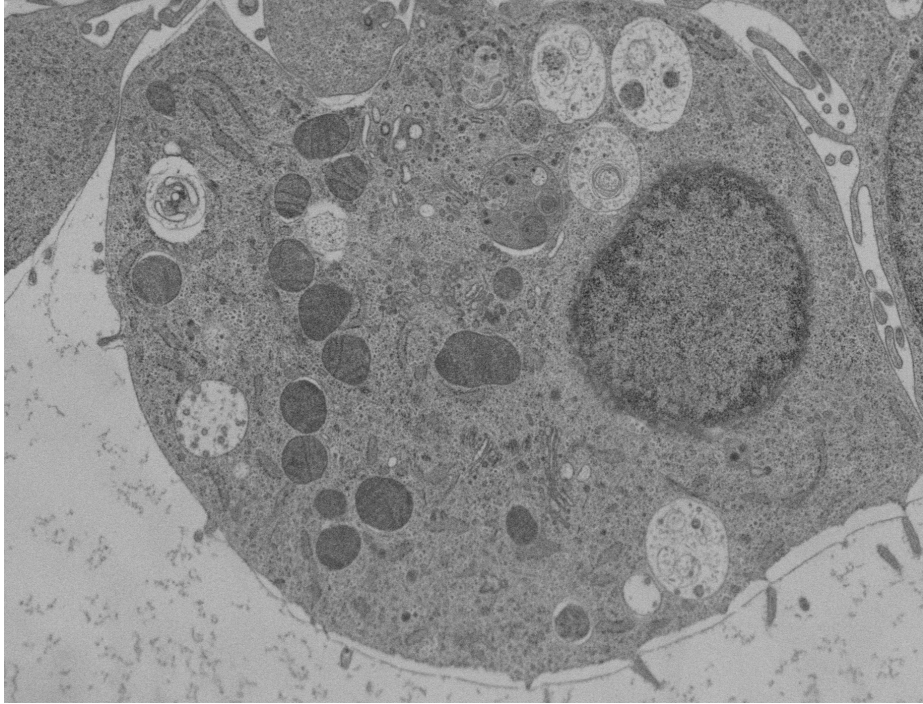


FIGURE A.16 – Image LW_4-1 , coupe 001.

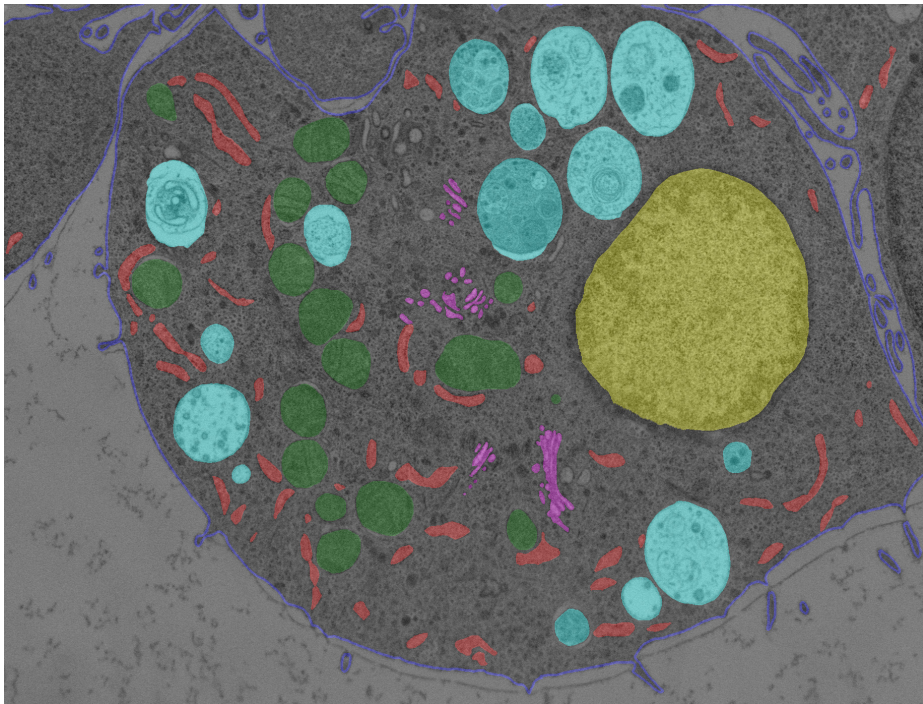


FIGURE A.17 – Image LW_4-1 avec segmentation de référence, coupe 001.

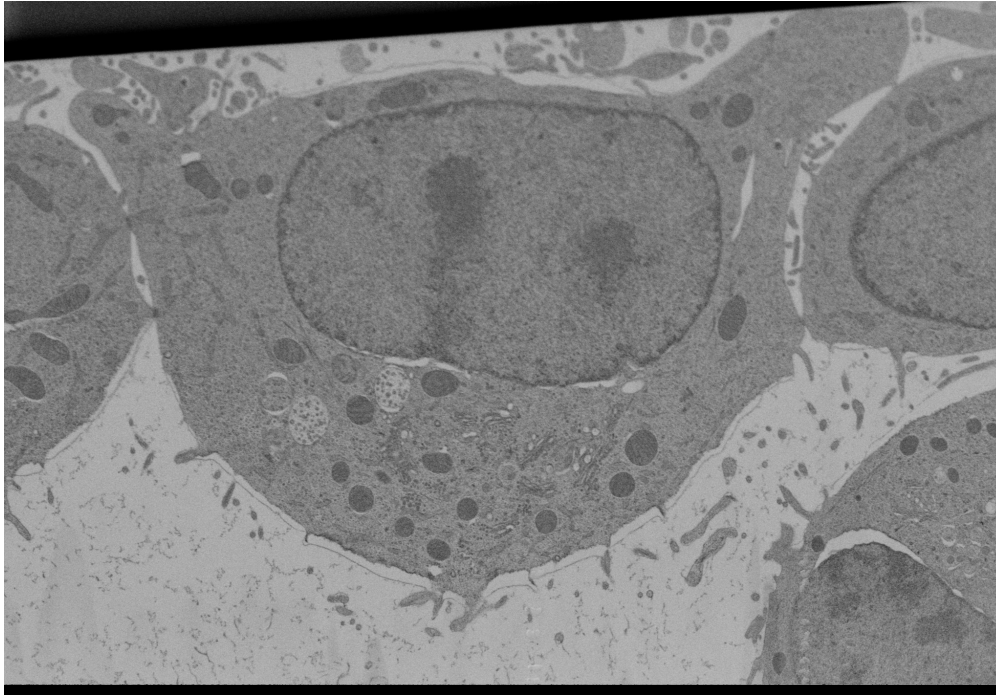


FIGURE A.18 – Image *LW4-2*, coupe 116.

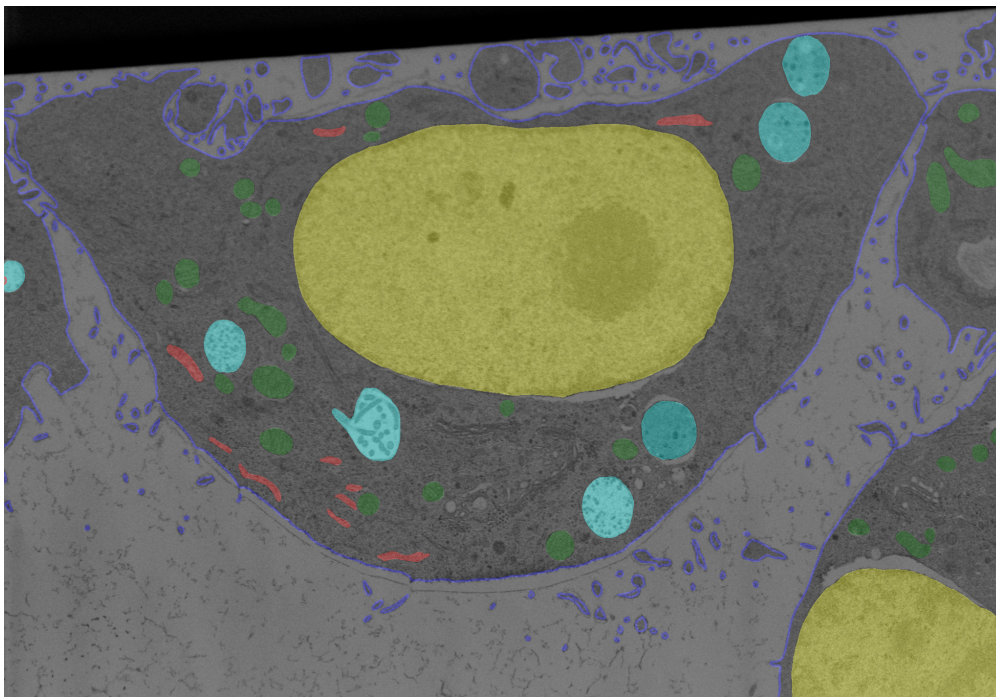


FIGURE A.19 – Image *LW4-2* avec segmentation de référence, coupe 013.

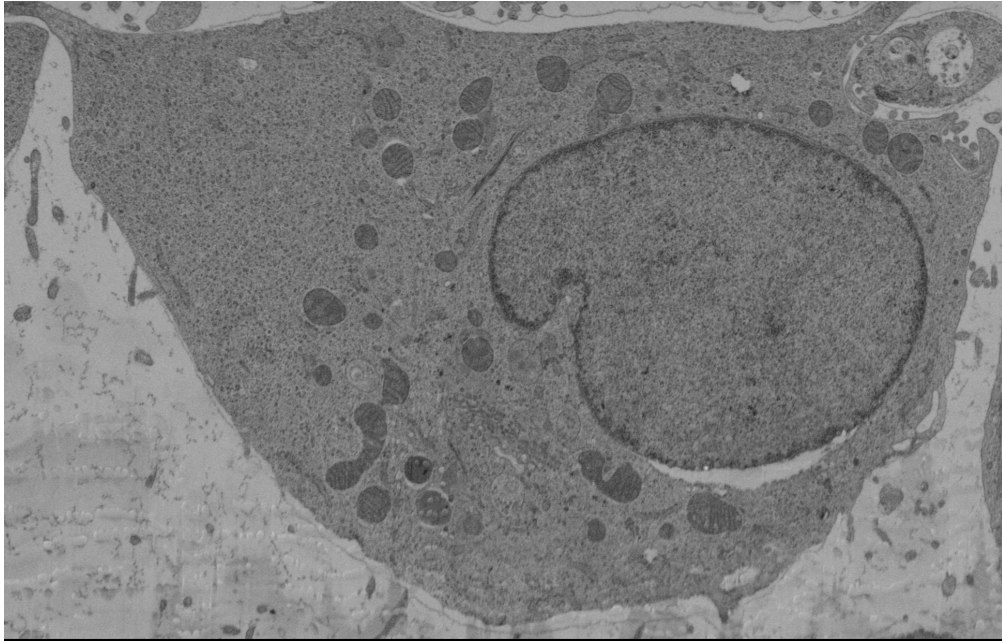


FIGURE A.20 – Image *LW4-3*, coupe 074.

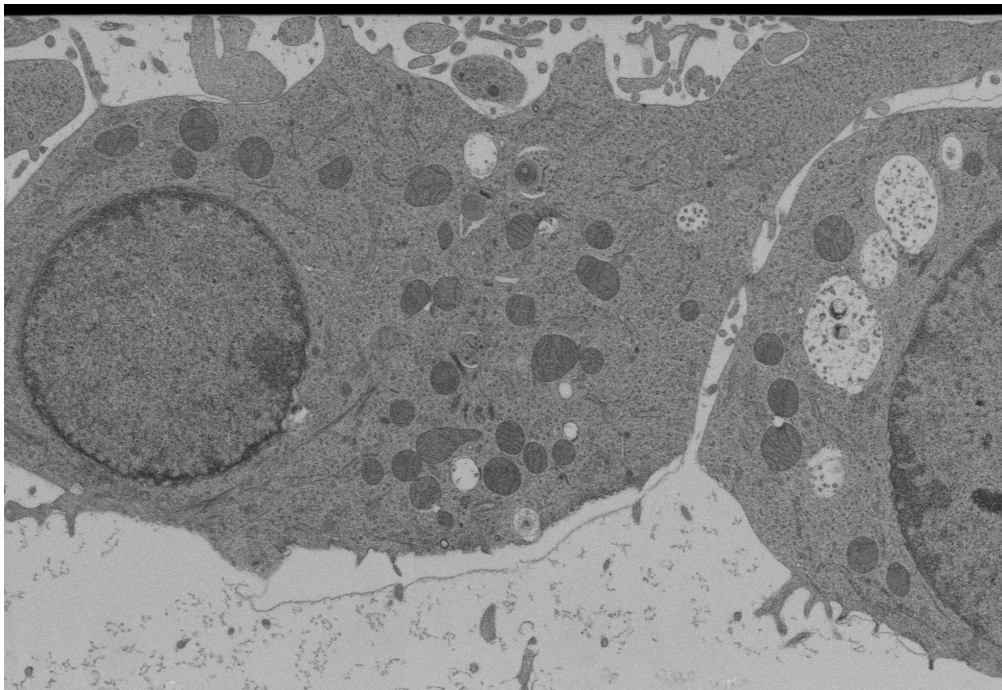


FIGURE A.21 – Image *LW4-4*, coupe 001.

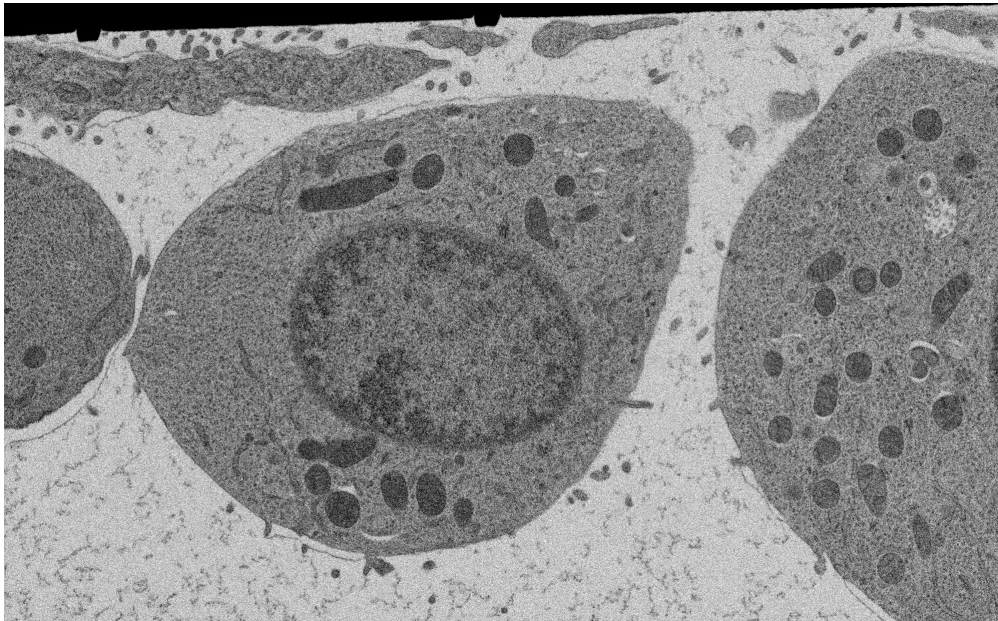


FIGURE A.22 – Image *LW2-1*, coupe 059.

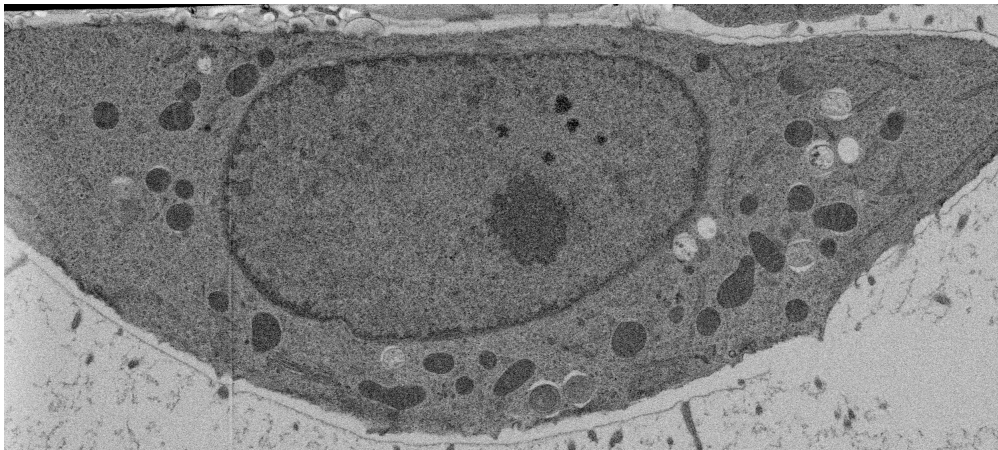


FIGURE A.23 – Image *LW2-4*, coupe 521.



FIGURE A.24 – Image *LW2-5*, coupe 096.

A.4 Figures

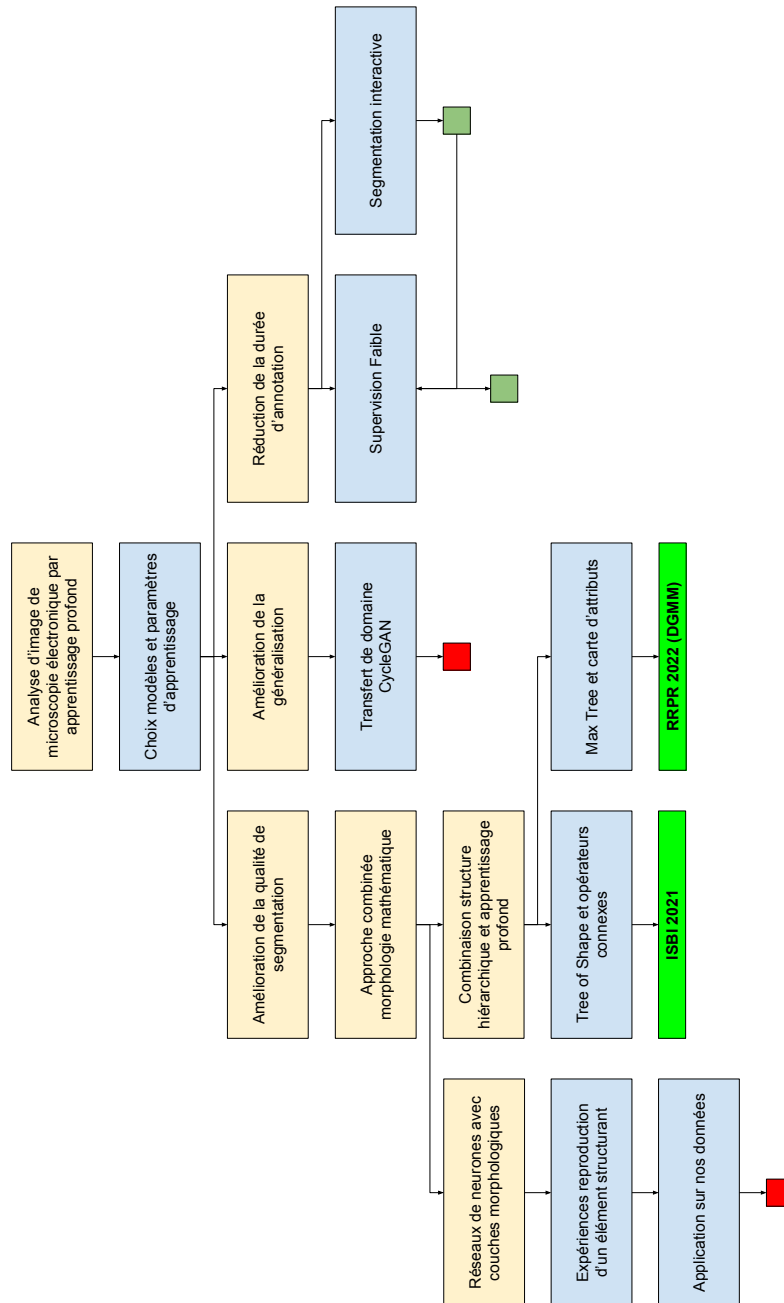


FIGURE A.25 – Représentation schématique de l’exploration scientifique de la thèse.

Bibliographie

- [Abdollahzadeh et al., 2021] Abdollahzadeh, A., Belevich, I., Jokitalo, E., Sierra, A., and Tohka, J. (2021). DeepACSON automated segmentation of white matter in 3D electron microscopy. *Commun Biol*, 4(1) :1–14.
- [Agustsson et al., 2019] Agustsson, E., Uijlings, J. R., and Ferrari, V. (2019). Interactive Full Image Segmentation by Considering All Regions Jointly. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11614–11623.
- [Ahmed et al., 2020] Ahmed, I., Ahmad, M., Khan, F. A., and Asif, M. (2020). Comparison of Deep-Learning-Based Segmentation Models : Using Top View Person Images. *IEEE Access*, 8 :136361–136373.
- [Alom et al., 2019] Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M., and Asari, V. K. (2019). Recurrent residual U-Net for medical image segmentation. *JMI*, 6(1) :014006.
- [Alpy et al., 2013] Alpy, F., Rousseau, A., Schwab, Y., Legueux, F., Stoll, I., Wendling, C., Spiegelhalter, C., Kessler, P., Mathelin, C., Rio, M.-C., Levine, T. P., and Tomasetto, C. (2013). STARD3 or STARD3NL and VAP form a novel molecular tether between late endosomes and the ER. *Journal of Cell Science*, 126(23) :5500–5512.
- [Angulo, 2010] Angulo, J. (2010). Pseudo-morphological Image Diffusion Using the Counter-Harmonic Paradigm. In Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., and Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems*, Lecture Notes in Computer Science, pages 426–437, Berlin, Heidelberg. Springer.
- [Arce et al., 2017] Arce, F., Zamora, E., and Sossa, H. (2017). Dendrite Ellipsoidal Neuron. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 795–802.
- [Arganda-Carreras et al., 2015] Arganda-Carreras, I., Turaga, S. C., Berger, D. R., Cireşan, D., Giusti, A., Gambardella, L. M., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J. M., Liu, T., Seyedhosseini, M., Tasdizen, T., Kamentsky, L., Burget, R., Uher, V., Tan, X., Sun, C., Pham, T. D., Bas, E., Uzunbas, M. G., Cardona, A., Schindelin, J., and Seung, H. S. (2015). Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9.

- [Armanious et al., 2019] Armanious, K., Jiang, C., Abdulatif, S., Küstner, T., Gatiadis, S., and Yang, B. (2019). Unsupervised Medical Image Translation Using Cycle-MedGAN. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5.
- [Aydin et al., 2021] Aydin, O. U., Taha, A. A., Hilbert, A., Khalil, A. A., Galinovic, I., Fiebach, J. B., Frey, D., and Madai, V. I. (2021). On the usage of average Hausdorff distance for segmentation performance assessment : Hidden error when used for ranking. *European Radiology Experimental*, 5(1) :4.
- [Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12) :2481–2495.
- [Barmpoutis and Ritter, 2007] Barmpoutis, A. and Ritter, G. X. (2007). Orthonormal Basis Lattice Neural Networks. In Kaburlasos, V. G. and Ritter, G. X., editors, *Computational Intelligence Based on Lattice Theory*, Studies in Computational Intelligence, pages 45–58. Springer, Berlin, Heidelberg.
- [Berger et al., 2007] Berger, Ch., Geraud, Th., Levillain, R., Widynski, N., Baillard, A., and Bertin, E. (2007). Effective Component Tree Computation with Application to Pattern Recognition in Astronomical Imaging. In *2007 IEEE International Conference on Image Processing*, volume 4, pages IV – 41–IV – 44.
- [Boykov and Jolly, 2001] Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 105–112, Vancouver, BC, Canada. IEEE Comput. Soc.
- [Briggman and Bock, 2012] Briggman, K. L. and Bock, D. D. (2012). Volume electron microscopy for neuronal circuit reconstruction. *Current Opinion in Neurobiology*, 22(1) :154–161.
- [Bullen, 2003] Bullen, P. S. (2003). *Handbook of Means and Their Inequalities*. Springer Netherlands, Dordrecht.
- [Cai et al., 2020] Cai, S., Tian, Y., Lui, H., Zeng, H., Wu, Y., and Chen, G. (2020). Dense-UNet : A novel multiphoton in vivo cellular image segmentation model based on a convolutional neural network. *Quant Imaging Med Surg*, 10(6) :1275–1285.
- [Cardona et al., 2010] Cardona, A., Saalfeld, S., Preibisch, S., Schmid, B., Cheng, A., Pulokas, J., Tomancak, P., and Hartenstein, V. (2010). An Integrated Micro- and Macroarchitectural Analysis of the Drosophila Brain by Computer-Assisted Serial Section Electron Microscopy. *PLOS Biology*, 8(10) :e1000502.
- [Carlinet et al., 2018] Carlinet, E., Crozet, S., and Géraud, T. (2018). The Tree of Shapes Turned into a Max-Tree : A Simple and Efficient Linear Algorithm. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1488–1492.
- [Carlinet and Géraud, 2013] Carlinet, E. and Géraud, T. (2013). A Comparison of Many Max-tree Computation Algorithms. In Hendriks, C. L. L., Borgefors, G., and Strand,

- R., editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 73–85, Berlin, Heidelberg. Springer.
- [Carlinet and Géraud, 2015] Carlinet, E. and Géraud, T. (2015). MToS : A Tree of Shapes for Multivariate Images. *IEEE Transactions on Image Processing*, 24(12) :5330–5342.
- [Casser et al., 2020] Casser, V., Kang, K., Pfister, H., and Haehn, D. (2020). Fast Mitochondria Detection for Connectomics. *arXiv :1812.06024 [cs]*.
- [Charisopoulos and Maragos, 2017] Charisopoulos, V. and Maragos, P. (2017). Morphological Perceptrons : Geometry and Training Algorithms. In Angulo, J., Velasco-Forero, S., and Meyer, F., editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 3–15, Cham. Springer International Publishing.
- [Chaurasia and Culurciello, 2017] Chaurasia, A. and Culurciello, E. (2017). LinkNet : Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.
- [Chen et al., 2018a] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018a). DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4) :834–848.
- [Chen et al., 2018b] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 833–851, Cham. Springer International Publishing.
- [Chen and Pan, 2018] Chen, X. and Pan, L. (2018). A Survey of Graph Cuts/Graph Search Based Medical Image Segmentation. *IEEE Reviews in Biomedical Engineering*, 11 :112–124.
- [Çiçek et al., 2016] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3D U-Net : Learning Dense Volumetric Segmentation from Sparse Annotation. In Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G., and Wells, W., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, Lecture Notes in Computer Science, pages 424–432, Cham. Springer International Publishing.
- [Conrad and Narayan, 2021] Conrad, R. and Narayan, K. (2021). CEM500K, a large-scale heterogeneous unlabeled cellular electron microscopy image dataset for deep learning. *eLife*, 10 :e65894.
- [Conrad and Narayan, 2023] Conrad, R. and Narayan, K. (2023). Instance segmentation of mitochondria in electron microscopy images with a generalist deep learning model trained on a diverse dataset. *Cell Systems*, 14(1) :58–71.e5.
- [Cordonnier et al., 2020] Cordonnier, J.-B., Loukas, A., and Jaggi, M. (2020). On the Relationship between Self-Attention and Convolutional Layers. In *International Conference on Learning Representations*.

- [Danielsson, 1980] Danielsson, P.-E. (1980). Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3) :227–248.
- [Davidson and Hummer, 1993] Davidson, J. L. and Hummer, F. (1993). Morphology neural networks : An introduction with applications. *Circuits Systems and Signal Process*, 12(2) :177–210.
- [Davidson and Ritter, 1990] Davidson, J. L. and Ritter, G. X. (1990). Theory of morphological neural networks. In *Digital Optical Computing II*, volume 1215, pages 378–388. SPIE.
- [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet : A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- [Denk and Horstmann, 2004] Denk, W. and Horstmann, H. (2004). Serial Block-Face Scanning Electron Microscopy to Reconstruct Three-Dimensional Tissue Nanostructure. *PLOS Biology*, 2(11) :e329.
- [Dosovitskiy et al., 2023] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2023). An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.
- [Dumoulin and Visin, 2016] Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*.
- [Falk et al., 2019] Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T. L., Prinz, M., Palme, K., Simons, M., Diester, I., Brox, T., and Ronneberger, O. (2019). U-Net : Deep learning for cell counting, detection, and morphometry. *Nat Methods*, 16(1) :67–70.
- [Farfan Cabrera et al., 2021a] Farfan Cabrera, D., Grossiord, É., Gogin, N., Papathanassiou, D., and Passat, N. (2021a). Analysis Of Lymph Node Tumor Features In Pet/Ct For Segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 588–592.
- [Farfan Cabrera et al., 2021b] Farfan Cabrera, D. L., Gogin, N., Morland, D., Naegel, B., Papathanassiou, D., and Passat, N. (2021b). Segmentation of Axillary and Supraclavicular Tumoral Lymph Nodes in PET/CT : A Hybrid CNN/Component-Tree Approach. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6672–6679.
- [Follain et al., 2020] Follain, G., Herrmann, D., Harlepp, S., Hyenne, V., Osmani, N., Warren, S. C., Timpson, P., and Goetz, J. G. (2020). Fluids and their mechanics in tumour transit : Shaping metastasis. *Nat Rev Cancer*, 20(2) :107–124.
- [Follain et al., 2018] Follain, G., Osmani, N., Azevedo, A. S., Allio, G., Mercier, L., Karremann, M. A., Solecki, G., Leôn, M. J. G., Lefebvre, O., Fekonja, N., Hille, C., Chabannes, V., Dollé, G., Metivet, T., Hovsepian, F. D., Prudhomme, C., Pichot, A., Paul, N., Carapito, R., Bahram, S., Ruthensteiner, B., Kemmling, A., Siemonsen, S.,

- Schneider, T., Fiehler, J., Glatzel, M., Winkler, F., Schwab, Y., Pantel, K., Harlepp, S., and Goetz, J. G. (2018). Hemodynamic Forces Tune the Arrest, Adhesion, and Extravasation of Circulating Tumor Cells. *Developmental Cell*, 45(1) :33–52.e12.
- [Franchi and Santucci, 2013] Franchi, A. and Santucci, M. (2013). The Contribution of Electron Microscopy to the Characterization of Soft Tissue Fibrosarcomas. *Ultrastructural Pathology*, 37(1) :9–14.
- [Franchi et al., 2020] Franchi, G., Fehri, A., and Yao, A. (2020). Deep morphological networks. *Pattern Recognition*, 102 :107246.
- [Fukushima, 1975] Fukushima, K. (1975). Cognitron : A self-organizing multilayered neural network. *Biol. Cybernetics*, 20(3) :121–136.
- [Géraud et al., 2013] Géraud, T., Carlinet, E., Crozet, S., and Najman, L. (2013). A Quasi-linear Algorithm to Compute the Tree of Shapes of nD Images. In Hendriks, C. L. L., Borgefors, G., and Strand, R., editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 98–110, Berlin, Heidelberg. Springer.
- [Ghoroghi et al., 2021] Ghoroghi, S., Mary, B., Larnicol, A., Asokan, N., Klein, A., Osmanni, N., Busnelli, I., Delalande, F., Paul, N., Halary, S., Gros, F., Fouillen, L., Haerberle, A.-M., Royer, C., Spiegelhalter, C., André-Grégoire, G., Mittelheisser, V., Detappe, A., Murphy, K., Timpson, P., Carapito, R., Blot-Chabaud, M., Gavard, J., Carapito, C., Vitale, N., Lefebvre, O., Goetz, J. G., and Hyenne, V. (2021). Ral GT-Pases promote breast cancer metastasis by controlling biogenesis and organ targeting of exosomes. *eLife*, 10 :e61539.
- [Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.
- [Goetz et al., 2007] Goetz, J. G., Genty, H., St-Pierre, P., Dang, T., Joshi, B., Sauvé, R., Vogl, W., and Nabi, I. R. (2007). Reversible interactions between smooth domains of the endoplasmic reticulum and mitochondria are regulated by physiological cytosolic Ca²⁺ levels. *Journal of Cell Science*, 120(20) :3553–3564.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.
- [Goodfellow et al., 2013] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13*, pages III–1319–III–1327, Atlanta, GA, USA. JMLR.org.
- [Grady, 2006] Grady, L. (2006). Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11) :1768–1783.
- [Grossiord et al., 2017] Grossiord, É., Talbot, H., Passat, N., Meignan, M., and Najman, L. (2017). Automated 3D lymphoma lesion segmentation from PET/CT characteristics. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 174–178.

- [Grossiord et al., 2015] Grossiord, É., Talbot, H., Passat, N., Meignan, M., Tervé, P., and Najman, L. (2015). Hierarchies and shape-space for pet image segmentation. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 1118–1121.
- [Guay et al., 2021a] Guay, M. D., Emam, Z. A. S., Anderson, A. B., Aronova, M. A., Pokrovskaya, I. D., Storrie, B., and Leapman, R. D. (2021a). Dense cellular segmentation for EM using 2D–3D neural network ensembles. *Sci Rep*, 11(1) :2561.
- [Guay et al., 2021b] Guay, M. D., Emam, Z. A. S., Anderson, A. B., Aronova, M. A., Pokrovskaya, I. D., Storrie, B., and Leapman, R. D. (2021b). Dense cellular segmentation for EM using 2D–3D neural network ensembles. *Sci Rep*, 11(1) :2561.
- [Haberl et al., 2018] Haberl, M. G., Churas, C., Tindall, L., Boassa, D., Phan, S., Bushong, E. A., Madany, M., Akay, R., Deerinck, T. J., Peltier, S. T., and Ellisman, M. H. (2018). CDeep3M—Plug-and-Play cloud-based deep learning for image segmentation. *Nat Methods*, 15(9) :677–680.
- [Harris et al., 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825) :357–362.
- [Harris et al., 2006] Harris, K. M., Perry, E., Bourne, J., Feinberg, M., Ostroff, L., and Hurlburt, J. (2006). Uniform Serial Sectioning for Transmission Electron Microscopy. *J. Neurosci.*, 26(47) :12101–12103.
- [Hayworth et al., 2006] Hayworth, KJ., Kasthuri, N., Schalek, R., and Lichtman, JW. (2006). Automating the Collection of Ultrathin Serial Sections for Large Volume TEM Reconstructions. *Microscopy and Microanalysis*, 12(S02) :86–87.
- [He et al., 2020] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2) :386–397.
- [He et al., 2016a] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA. IEEE.
- [He et al., 2016b] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity Mappings in Deep Residual Networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 630–645, Cham. Springer International Publishing.
- [Heinrich et al., 2021] Heinrich, L., Bennett, D., Ackerman, D., Park, W., Bogovic, J., Eckstein, N., Petruncio, A., Clements, J., Pang, S., Xu, C. S., Funke, J., Korff, W., Hess, H. F., Lippincott-Schwartz, J., Saalfeld, S., and Weigel, A. V. (2021). Whole-cell organelle segmentation in volume electron microscopy. *Nature*, pages 1–6.
- [Heinrich et al., 2020] Heinrich, L., Bennett, D., Ackerman, D., Park, W., Bogovic, J., Eckstein, N., Petruncio, A., Clements, J., Xu, C. S., Funke, J., Korff, W., Hess, H. F., Lippincott-Schwartz, J., Saalfeld, S., Weigel, A. V., and Team, C. P. (2020). Automatic

- whole cell organelle segmentation in volumetric electron microscopy. *bioRxiv*, page 2020.11.14.382143.
- [Heinrich et al., 2018] Heinrich, L., Funke, J., Pape, C., Nunez-Iglesias, J., and Saalfeld, S. (2018). Synaptic Cleft Segmentation in Non-isotropic Volume Electron Microscopy of the Complete *Drosophila* Brain. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, pages 317–325, Cham. Springer International Publishing.
- [Hermay et al., 2022] Hermay, R., Tochon, G., Puybareau, É., Kirszenberg, A., and Angulo, J. (2022). Learning Grayscale Mathematical Morphology with Smooth Morphological Layers. *J Math Imaging Vis*, 64(7) :736–753.
- [Hernández et al., 2017] Hernández, G., Zamora, E., and Sossa, H. (2017). Comparing Deep and Dendrite Neural Networks : A Case Study. In Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., and Olvera-López, J. A., editors, *Pattern Recognition*, Lecture Notes in Computer Science, pages 32–41, Cham. Springer International Publishing.
- [Heymann et al., 2006] Heymann, J. A. W., Hayles, M., Gestmann, I., Giannuzzi, L. A., Lich, B., and Subramaniam, S. (2006). Site-specific 3D imaging of cells and tissues with a dual beam microscope. *Journal of Structural Biology*, 155(1) :63–73.
- [Holschneider et al., 1990] Holschneider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, Ph. (1990). A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform. In Combes, J.-M., Grossmann, A., and Tchamitchian, P., editors, *Wavelets*, Inverse Problems and Theoretical Imaging, pages 286–297, Berlin, Heidelberg. Springer.
- [Hu et al., 2018] Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.
- [Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.
- [Hyenne et al., 2015] Hyenne, V., Apaydin, A., Rodriguez, D., Spiegelhalter, C., Hoff-Yoessle, S., Diem, M., Tak, S., Lefebvre, O., Schwab, Y., Goetz, J. G., and Labouesse, M. (2015). RAL-1 controls multivesicular body biogenesis and exosome secretion. *Journal of Cell Biology*, 211(1) :27–37.
- [Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR.
- [Jadon, 2020] Jadon, S. (2020). A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7.

- [Jang and Kim, 2019] Jang, W.-D. and Kim, C.-S. (2019). Interactive Image Segmentation via Backpropagating Refinement Scheme. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5292–5301.
- [Karreman et al., 2016] Karreman, M. A., Hyenne, V., Schwab, Y., and Goetz, J. G. (2016). Intravital Correlative Microscopy : Imaging Life at the Nanoscale. *Trends in Cell Biology*, 26(11) :848–863.
- [Kasthuri et al., 2015] Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., Lee, D., Vázquez-Reina, A., Kaynig, V., Jones, T. R., Roberts, M., Morgan, J. L., Tapia, J. C., Seung, H. S., Roncal, W. G., Vogelstein, J. T., Burns, R., Sussman, D. L., Priebe, C. E., Pfister, H., and Lichtman, J. W. (2015). Saturated Reconstruction of a Volume of Neocortex. *Cell*, 162(3) :648–661.
- [Keskar et al., 2016] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On Large-Batch Training for Deep Learning : Generalization Gap and Sharp Minima. In *International Conference on Learning Representations*.
- [Khan et al., 2020] Khan, Z., Yahya, N., Alsaih, K., Ali, S. S. A., and Meriaudeau, F. (2020). Evaluation of Deep Neural Networks for Semantic Segmentation of Prostate in T2W MRI. *Sensors*, 20(11) :3183.
- [Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam : A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- [Kirillov et al., 2023] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment Anything.
- [Kirszenberg et al., 2021] Kirszenberg, A., Tochon, G., Puybureau, É., and Angulo, J. (2021). Going Beyond p-convolutions to Learn Grayscale Morphological Operators. In Lindblad, J., Malmberg, F., and Sladoje, N., editors, *Discrete Geometry and Mathematical Morphology*, Lecture Notes in Computer Science, pages 470–482, Cham. Springer International Publishing.
- [Knott et al., 2008] Knott, G., Marchman, H., Wall, D., and Lich, B. (2008). Serial Section Scanning Electron Microscopy of Adult Brain Tissue Using Focused Ion Beam Milling. *J. Neurosci.*, 28(12) :2959–2964.
- [Krizhevsky et al., 2012a] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). ImageNet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.
- [Krizhevsky et al., 2012b] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.
- [Kugelman et al., 2022] Kugelman, J., Allman, J., Read, S. A., Vincent, S. J., Tong, J., Kalloniatis, M., Chen, F. K., Collins, M. J., and Alonso-Caneiro, D. (2022). A

- comparison of deep learning U-Net architectures for posterior segment OCT retinal layer segmentation. *Sci Rep*, 12(1) :14888.
- [LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4) :541–551.
- [Li et al., 2018a] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018a). Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, pages 6391–6401, Red Hook, NY, USA. Curran Associates Inc.
- [Li et al., 2018b] Li, Z., Chen, Q., and Koltun, V. (2018b). Interactive Image Segmentation with Latent Diversity. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 577–585.
- [Liew et al., 2017] Liew, J., Wei, Y., Xiong, W., Ong, S.-H., and Feng, J. (2017). Regional Interactive Image Segmentation Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2746–2754.
- [Lin et al., 2016] Lin, D., Dai, J., Jia, J., He, K., and Sun, J. (2016). ScribbleSup : Scribble-Supervised Convolutional Networks for Semantic Segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3159–3167, Las Vegas, NV, USA. IEEE.
- [Lin et al., 2017] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944.
- [Liu et al., 2010] Liu, D., Xiong, Y., Shapiro, L., and Pulli, K. (2010). Robust interactive image segmentation with automatic boundary refinement. In *2010 IEEE International Conference on Image Processing*, pages 225–228.
- [Liu et al., 2020] Liu, J., Li, L., Yang, Y., Hong, B., Chen, X., Xie, Q., and Han, H. (2020). Automatic Reconstruction of Mitochondria and Endoplasmic Reticulum in Electron Microscopy Volumes by Deep Learning. *Front. Neurosci.*, 14.
- [Liu et al., 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin Transformer : Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, Montreal, QC, Canada. IEEE.
- [Liu et al., 2022] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A ConvNet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976.
- [Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.
- [Lucchi et al., 2012] Lucchi, A., Smith, K., Achanta, R., Knott, G., and Fua, P. (2012). Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks With Learned Shape Features. *IEEE Transactions on Medical Imaging*, 31(2) :474–486.

- [Maclagan and Sturmfels, 2021] Maclagan, D. and Sturmfels, B. (2021). *Introduction to Tropical Geometry*. Graduate Studies in Mathematics. American Mathematical Society.
- [Maninis et al., 2018] Maninis, K.-K., Caelles, S., Pont-Tuset, J., and Van Gool, L. (2018). Deep Extreme Cut : From Extreme Points to Object Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 616–625.
- [Masci et al., 2013] Masci, J., Angulo, J., and Schmidhuber, J. (2013). A Learning Framework for Morphological Operators Using Counter-Harmonic Mean. In Hendriks, C. L. L., Borgefors, G., and Strand, R., editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 329–340, Berlin, Heidelberg. Springer.
- [Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10) :761–767.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4) :115–133.
- [Meyer et al., 2023] Meyer, C., Baudrier, É., Schultz, P., and Naegel, B. (2023). Combining Max-Tree and CNN for Segmentation of Cellular FIB-SEM Images. In Kerautret, B., Colom, M., Krähenbühl, A., Lopresti, D., Monasse, P., and Perret, B., editors, *Reproducible Research in Pattern Recognition*, Lecture Notes in Computer Science, pages 77–90, Cham. Springer Nature Switzerland.
- [Meyer et al., 2021] Meyer, C., Mallouh, V., Spehner, D., Baudrier, É., Schultz, P., and Naegel, B. (2021). Automatic Multi Class Organelle Segmentation For Cellular Fib-Sem Images. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 668–672.
- [Minaee et al., 2022] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2022). Image Segmentation Using Deep Learning : A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7) :3523–3542.
- [Mo et al., 2022] Mo, Y., Wu, Y., Yang, X., Liu, F., and Liao, Y. (2022). Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493 :626–646.
- [Moen et al., 2019] Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., and Van Valen, D. (2019). Deep learning for cellular image analysis. *Nat Methods*, 16(12) :1233–1246.
- [Monasse, 2000] Monasse, P. (2000). *Représentation Morphologique d’images Numériques et Application Au Recalage*. These de doctorat, Paris 9.
- [Monasse and Guichard, 1999] Monasse, P. and Guichard, F. (1999). Scale-Space from a Level Lines Tree. In Nielsen, M., Johansen, P., Olsen, O. F., and Weickert, J., editors, *Scale-Space Theories in Computer Vision*, Lecture Notes in Computer Science, pages 175–186, Berlin, Heidelberg. Springer.

- [Mondal et al., 2020] Mondal, R., Dey, M. S., and Chanda, B. (2020). Image Restoration by Learning Morphological Opening-Closing Network. *Mathematical Morphology - Theory and Applications*, 4(1) :87–107.
- [Mondal et al., 2019] Mondal, R., Purkait, P., Santra, S., and Chanda, B. (2019). Morphological Networks for Image De-raining. In Couprie, M., Cousty, J., Kenmochi, Y., and Mustafa, N., editors, *Discrete Geometry for Computer Imagery*, Lecture Notes in Computer Science, pages 262–275, Cham. Springer International Publishing.
- [Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML’10, pages 807–814, Madison, WI, USA. Omnipress.
- [Najman and Couprie, 2006] Najman, L. and Couprie, M. (2006). Building the Component Tree in Quasi-Linear Time. *IEEE Transactions on Image Processing*, 15(11) :3531–3539.
- [Najman and Talbot, 2008] Najman, L. and Talbot, H. (2008). *Morphologie Mathématique 1 : Approches Déterministes*. Hermès/Lavoisier.
- [Najman and Talbot, 2013] Najman, L. and Talbot, H. (2013). *Mathematical Morphology : From Theory to Applications*. John Wiley & Sons.
- [Nasr-Esfahani et al., 2016] Nasr-Esfahani, E., Samavi, S., Karimi, N., Sorousmehr, S., Ward, K., Jafari, M., Felfeliyan, B., Nallamotheu, B., and Najarian, K. (2016). Vessel extraction in X-ray angiograms using deep learning. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 643–646.
- [Nguyen et al., 2021] Nguyen, T., Pham, Q.-H., Le, T., Pham, T., Ho, N., and Hua, B.-S. (2021). Point-set Distances for Learning Representations of 3D Point Clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10458–10467.
- [Nguyen et al., 2012] Nguyen, T. N. A., Cai, J., Zhang, J., and Zheng, J. (2012). Robust Interactive Image Segmentation Using Convex Active Contours. *IEEE Transactions on Image Processing*, 21(8) :3734–3743.
- [Ning et al., 2010] Ning, J., Zhang, L., Zhang, D., and Wu, C. (2010). Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition*, 43(2) :445–456.
- [Nogueira et al., 2021] Nogueira, K., Chanussot, J., Mura, M. D., and Santos, J. A. D. (2021). An Introduction to Deep Morphological Networks. *IEEE Access*, 9 :114308–114324.
- [Noh et al., 2015] Noh, H., Hong, S., and Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1528.
- [Nur-A-Alam et al., 2021] Nur-A-Alam, Ahsan, M., Based, M. A., Haider, J., and Kowalski, M. (2021). An intelligent system for automatic fingerprint identification using

- feature fusion by Gabor filter and deep learning. *Computers and Electrical Engineering*, 95 :107387.
- [Oktay et al., 2018] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D. (2018). Attention U-Net : Learning Where to Look for the Pancreas.
- [Papadopoulos et al., 2017] Papadopoulos, D. P., Uijlings, J. R. R., Keller, F., and Ferrari, V. (2017). Extreme Clicking for Efficient Object Annotation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4940–4949.
- [Perret et al., 2019] Perret, B., Chierchia, G., Cousty, J., F. Guimarães, S. J., Kenmochi, Y., and Najman, L. (2019). Higua : Hierarchical Graph Analysis. *SoftwareX*, 10 :100335.
- [Pessoa and Maragos, 1996] Pessoa, L. and Maragos, P. (1996). Morphological/rank neural networks and their adaptive optimal design for image processing. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 6, pages 3398–3401 vol. 6.
- [Pessoa and Maragos, 1998] Pessoa, L. and Maragos, P. (1998). MRL-filters : A general class of nonlinear systems and their optimal design for image processing. *IEEE Transactions on Image Processing*, 7(7) :966–978.
- [Pessoa and Maragos, 2000] Pessoa, L. F. C. and Maragos, P. (2000). Neural networks with hybrid morphological/rank/linear nodes : A unifying framework with applications to handwritten character recognition. *Pattern Recognition*, 33(6) :945–960.
- [Puccio et al., 2016] Puccio, B., Pooley, J. P., Pellman, J. S., Taverna, E. C., and Craddock, R. C. (2016). The preprocessed connectomes project repository of manually corrected skull-stripped T1-weighted anatomical MRI data. *GigaScience*, 5(1) :s13742–016–0150–5.
- [Ramadan et al., 2020] Ramadan, H., Lachqar, C., and Tairi, H. (2020). A survey of recent interactive image segmentation methods. *Comp. Visual Media*, 6(4) :355–384.
- [Ritter et al., 2003] Ritter, G., Iancu, L., and Urcid, G. (2003). Morphological perceptrons with dendritic structure. In *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03.*, volume 2, pages 1296–1301 vol.2.
- [Ritter and Sussner, 1996] Ritter, G. and Sussner, P. (1996). An introduction to morphological neural networks. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 4, pages 709–717 vol.4.
- [Ritter and Urcid, 2003] Ritter, G. and Urcid, G. (2003). Lattice algebra approach to single-neuron computation. *IEEE Transactions on Neural Networks*, 14(2) :282–295.
- [Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net : Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham. Springer International Publishing.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408.

- [Roy et al., 2018] Roy, A. G., Navab, N., and Wachinger, C. (2018). Concurrent Spatial and Channel ‘Squeeze & Excitation’ in Fully Convolutional Networks. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, pages 421–429, Cham. Springer International Publishing.
- [Roy et al., 2019] Roy, A. G., Navab, N., and Wachinger, C. (2019). Recalibrating Fully Convolutional Networks With Spatial and Channel “Squeeze and Excitation” Blocks. *IEEE Transactions on Medical Imaging*, 38(2) :540–549.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088) :533–536.
- [Salembier et al., 1998] Salembier, P., Oliveras, A., and Garrido, L. (1998). Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4) :555–570.
- [Salembier and Wilkinson, 2009] Salembier, P. and Wilkinson, M. H. (2009). Connected operators. *IEEE Signal Processing Magazine*, 26(6) :136–157.
- [Santucci and Franchi, 2008] Santucci, M. and Franchi, A. (2008). Recognizing Hidden Phenotypes in Sarcomas Through the Electron Microscope. *Ultrastructural Pathology*, 32(2) :51–61.
- [Schalek et al., 2016] Schalek, R., Lee, D., Kasthuri, N., Peleg, A., Jones, T., Kaynig, V., Haehn, D., Pfister, H., Cox, D., and Lichtman, JW. (2016). Imaging a 1 mm³ Volume of Rat Cortex Using a MultiBeam SEM. *Microscopy and Microanalysis*, 22(S3) :582–583.
- [Schalek et al., 2012] Schalek, R., Wilson, A., Lichtman, J., Josh, M., Kasthuri, N., Berger, D., Seung, S., Anger, P., Hayworth, K., and Aderhold, D. (2012). ATUM-based SEM for High-Speed Large-Volume Biological Reconstructions. *Microscopy and Microanalysis*, 18(S2) :572–573.
- [Selmi et al., 2017] Selmi, Z., Ben Halima, M., and Alimi, A. M. (2017). Deep Learning System for Automatic License Plate Detection and Recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1132–1138.
- [Shapson-Coe et al., 2021] Shapson-Coe, A., Januszewski, M., Berger, D. R., Pope, A., Wu, Y., Blakely, T., Schalek, R. L., Li, P., Wang, S., Maitin-Shepard, J., Karlupia, N., Dorkenwald, S., Sjostedt, E., Leavitt, L., Lee, D., Bailey, L., Fitzmaurice, A., Kar, R., Field, B., Wu, H., Wagner-Carena, J., Aley, D., Lau, J., Lin, Z., Wei, D., Pfister, H., Peleg, A., Jain, V., and Lichtman, J. W. (2021). A connectomic study of a petascale fragment of human cerebral cortex.
- [Shen et al., 2022] Shen, Y., Shih, F. Y., Zhong, X., and Chang, I.-C. (2022). Deep Morphological Neural Networks. *Int. J. Patt. Recogn. Artif. Intell.*, 36(12) :2252023.
- [Shensa, 1992] Shensa, M. (1992). The discrete wavelet transform : Wedding the a trous and Mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10) :2464–2482.

- [Shih et al., 2019] Shih, F. Y., Shen, Y., and Zhong, X. (2019). Development of Deep Learning Framework for Mathematical Morphology. *Int. J. Patt. Recogn. Artif. Intell.*, 33(06) :1954024.
- [Simard et al., 2003] Simard, P., Steinkraus, D., and Platt, J. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963.
- [Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition.
- [Sofiuk et al., 2020] Sofiuk, K., Petrov, I., Barinova, O., and Konushin, A. (2020). F-BRS : Rethinking Backpropagating Refinement for Interactive Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8620–8629.
- [Sofiuk et al., 2022] Sofiuk, K., Petrov, I. A., and Konushin, A. (2022). Reviving Iterative Training with Mask Guidance for Interactive Segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145.
- [Soille, 2004] Soille, P. (2004). *Morphological Image Analysis : Principles and Applications*. Springer-Verlag, Berlin Heidelberg, 2 edition.
- [Sossa and Guevara, 2014] Sossa, H. and Guevara, E. (2014). Efficient training for dendrite morphological neural networks. *Neurocomputing*, 131 :132–142.
- [Spiegelhalter et al., 2010] Spiegelhalter, C., Tosch, V., Hentsch, D., Koch, M., Kessler, P., Schwab, Y., and Laporte, J. (2010). From Dynamic Live Cell Imaging to 3D Ultrastructure : Novel Integrated Methods for High Pressure Freezing and Correlative Light-Electron Microscopy. *PLOS ONE*, 5(2) :e9014.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56) :1929–1958.
- [Stenger et al., 2023] Stenger, A., Vedrenne, L., Schultz, P., Faisan, S., Baudrier, É., and Naegel, B. (2023). Fast and interpretable unsupervised domain adaptation for FIB-SEM cell segmentation. In *International Symposium on Biomedical Imaging (ISBI), Cartagena, Colombia, Avril 2023*.
- [Su et al., 2021] Su, R., Zhang, D., Liu, J., and Cheng, C. (2021). MSU-Net : Multi-Scale U-Net for 2D Medical Image Segmentation. *Frontiers in Genetics*, 12.
- [Sudre et al., 2017] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. (2017). Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In Cardoso, M. J., Arbel, T., Carneiro, G., Syeda-Mahmood, T., Tavares, J. M. R., Moradi, M., Bradley, A., Greenspan, H., Papa, J. P., Madabhushi, A., Nascimento, J. C., Cardoso, J. S., Belagiannis, V., and Lu, Z., editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Lecture Notes in Computer Science, pages 240–248, Cham. Springer International Publishing.

- [Sussner, 1998] Sussner, P. (1998). Morphological perceptron learning. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) Held Jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, pages 477–482.
- [Sussner and Esmi, 2011] Sussner, P. and Esmi, E. L. (2011). Morphological perceptrons with competitive learning : Lattice-theoretical framework and constructive learning algorithm. *Information Sciences*, 181(10) :1929–1950.
- [Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9.
- [Talbot and Najman, 2010] Talbot, H. and Najman, L. (2010). *Morphologie Mathématique 2 : Estimation, Choix et Mise En Œuvre*. Hermès/Lavoisier.
- [Tan and Le, 2019] Tan, M. and Le, Q. (2019). EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR.
- [Urbach and Wilkinson, 2002] Urbach, ER. and Wilkinson, MHF. (2002). Shape-only granulometries and gray-scale shape filters : 6th International Symposium on Mathematical Morphology. *MATHEMATICAL MORPHOLOGY, PROCEEDINGS*, pages 305–314.
- [Van der Walt et al., 2014] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. (2014). Scikit-image : Image processing in Python. *PeerJ*, 2 :e453.
- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- [Velasco-Forero et al., 2022a] Velasco-Forero, S., Pagès, R., and Angulo, J. (2022a). Learnable Empirical Mode Decomposition based on Mathematical Morphology. *SIAM J. Imaging Sci.*, 15(1) :23–44.
- [Velasco-Forero et al., 2022b] Velasco-Forero, S., Rhim, A., and Angulo, J. (2022b). Fixed point layers for geodesic morphological operations. In *British Machine Vision Conference (BMVC)*.
- [Wang et al., 2018a] Wang, G., Li, W., Zuluaga, M. A., Pratt, R., Patel, P. A., Aertsen, M., Doel, T., David, A. L., Deprest, J., Ourselin, S., and Vercauteren, T. (2018a). Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning. *IEEE Transactions on Medical Imaging*, 37(7) :1562–1573.
- [Wang et al., 2019] Wang, G., Zuluaga, M. A., Li, W., Pratt, R., Patel, P. A., Aertsen, M., Doel, T., David, A. L., Deprest, J., Ourselin, S., and Vercauteren, T. (2019). DeepIGeoS : A Deep Interactive Geodesic Framework for Medical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7) :1559–1572.

- [Wang et al., 2018b] Wang, J., Lu, J., Qin, G., Shen, L., Sun, Y., Ying, H., Zhang, Z., and Hu, W. (2018b). Technical Note : A deep learning-based autosegmentation of rectal tumors in MR images. *Medical Physics*, 45(6) :2560–2564.
- [Wei et al., 2020] Wei, D., Lin, Z., Franco-Barranco, D., Wendt, N., Liu, X., Yin, W., Huang, X., Gupta, A., Jang, W.-D., Wang, X., Arganda-Carreras, I., Lichtman, J. W., and Pfister, H. (2020). MitoEM Dataset : Large-Scale 3D Mitochondria Instance Segmentation from EM Images. In Martel, A. L., Abolmaesumi, P., Stoyanov, D., Mateus, D., Zuluaga, M. A., Zhou, S. K., Racoceanu, D., and Joskowicz, L., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, Lecture Notes in Computer Science, pages 66–76, Cham. Springer International Publishing.
- [Wilhelm et al., 2017] Wilhelm, L. P., Wendling, C., Védie, B., Kobayashi, T., Chenard, M.-P., Tomasetto, C., Drin, G., and Alpy, F. (2017). STARD3 mediates endoplasmic reticulum-to-endosome cholesterol transport at membrane contact sites. *The EMBO Journal*, 36(10) :1412–1433.
- [Wilson, 1989] Wilson, S. S. (1989). Morphological Networks. In *Visual Communications and Image Processing IV*, volume 1199, pages 483–495. SPIE.
- [Xiao et al., 2018] Xiao, C., Chen, X., Li, W., Li, L., Wang, L., Xie, Q., and Han, H. (2018). Automatic Mitochondria Segmentation for EM Data Using a 3D Supervised Convolutional Network. *Front. Neuroanat.*, 12.
- [Xu et al., 2021] Xu, C. S., Pang, S., Shtengel, G., Müller, A., Ritter, A. T., Hoffman, H. K., Takemura, S.-y., Lu, Z., Pasoli, H. A., Iyer, N., Chung, J., Bennett, D., Weigel, A. V., Freeman, M., van Engelenburg, S. B., Walther, T. C., Farese, R. V., Lippincott-Schwartz, J., Mellman, I., Solimena, M., and Hess, H. F. (2021). An open-access volume electron microscopy atlas of whole cells and tissues. *Nature*, pages 1–5.
- [Xu et al., 2020] Xu, C. S., Pang, S., Shtengel, G., Müller, A., Ritter, A. T., Hoffman, H. K., Takemura, S.-y., Lu, Z., Pasoli, H. A., Iyer, N., Chung, J., Bennett, D., Weigel, A. V., Walther, T. C., Farese, R. V., van Engelenburg, S. B., Mellman, I., Solimena, M., and Hess, H. F. (2020). Isotropic 3D electron microscopy reference library of whole cells and tissues. *bioRxiv*, page 2020.11.13.382457.
- [Xu et al., 2017] Xu, N., Price, B., Cohen, S., Yang, J., and Huang, T. (2017). Deep GrabCut for Object Selection. In *Proceedings of the British Machine Vision Conference 2017*, page 182, London, UK. British Machine Vision Association.
- [Xu et al., 2016] Xu, N., Price, B., Cohen, S., Yang, J., and Huang, T. S. (2016). Deep Interactive Object Selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381.
- [Yakubovskiy, 2019] Yakubovskiy, P. (2019). Segmentation models.
- [Yeghiazaryan and Voiculescu, 2018] Yeghiazaryan, V. and Voiculescu, I. (2018). Family of boundary overlap metrics for the evaluation of medical image segmentation. *J Med Imaging (Bellingham)*, 5(1) :015006.
- [Zamora and Sossa, 2016] Zamora, E. and Sossa, H. (2016). Dendrite morphological neurons trained by stochastic gradient descent. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8.

- [Zeng et al., 2017] Zeng, T., Wu, B., and Ji, S. (2017). DeepEM3D : Approaching human-level performance on 3D anisotropic EM image segmentation. *Bioinformatics*, 33(16) :2555–2562.
- [Zhang et al., 2019] Zhang, Y., Blusseau, S., Velasco-Forero, S., Bloch, I., and Angulo, J. (2019). Max-Plus Operators Applied to Filter Selection and Model Pruning in Neural Networks. In Burgeth, B., Kleefeld, A., Naegel, B., Passat, N., and Perret, B., editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, Lecture Notes in Computer Science, pages 310–322, Cham. Springer International Publishing.
- [Zhao et al., 2017] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid Scene Parsing Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239.
- [Zheng et al., 2015] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. S. (2015). Conditional Random Fields as Recurrent Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537.
- [Zhou et al., 2016] Zhou, C., Wu, D., Qin, W., and Liu, C. (2016). An efficient two-stage region merging method for interactive image segmentation. *Computers & Electrical Engineering*, 54 :220–229.
- [Zhou et al., 2018] Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., and Liang, J. (2018). UNet++ : A Nested U-Net Architecture for Medical Image Segmentation. In Stoyanov, D., Taylor, Z., Carneiro, G., Syeda-Mahmood, T., Martel, A., Maier-Hein, L., Tavares, J. M. R., Bradley, A., Papa, J. P., Belagiannis, V., Nascimento, J. C., Lu, Z., Conjeti, S., Moradi, M., Greenspan, H., and Madabhushi, A., editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Lecture Notes in Computer Science, pages 3–11, Cham. Springer International Publishing.
- [Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232.

Segmentation d'images de microscopie électronique par apprentissage profond pour l'analyse quantitative de l'ultrastructure cellulaire

Résumé

L'imagerie FIB-SEM permet d'obtenir des images 3D de résolution nanométrique en microscopie électronique cellulaire. Une segmentation sémantique de ces images est nécessaire pour permettre une analyse quantitative et comparative de l'ultrastructure cellulaire. Les méthodes actuelles basées sur les réseaux de neurones convolutifs rencontrent des défis liés à la rareté des données annotées, au bruit et aux variations d'acquisition.

L'objectif de la thèse est de développer des méthodes de segmentation robustes. La première partie de nos travaux se concentre sur l'optimisation des hyperparamètres pour la segmentation d'image par apprentissage profond. Une seconde partie s'intéresse aux méthodes interactives et utilisant les annotations faibles afin de réduire le temps d'annotation. La dernière partie concerne les méthodes combinant apprentissage profond et morphologie mathématique dans le but d'améliorer la qualité de la segmentation.

Mots-clés : apprentissage profond, segmentation sémantique, morphologie mathématique, segmentation interactive, annotations faibles, microscopie électronique, imagerie cellulaire.

Résumé en anglais

FIB-SEM imaging provides 3D images with nanometric resolution in cellular electron microscopy. Semantic segmentation of these images is required to enable quantitative and comparative analysis of cell ultrastructure. Current methods based on convolutional neural networks face challenges related to the scarcity of annotated data, noise, and acquisition variations.

The thesis aims to develop robust segmentation methods. The first part of our work focuses on hyperparameter optimization for image segmentation using deep learning. A second part explores interactive methods and sparse annotations to reduce annotation time. The last part concerns methods combining deep learning and mathematical morphology to improve segmentation quality.

Keywords: deep learning, semantic segmentation, mathematical morphology, interactive segmentation, sparse annotations, electron microscopy, cellular imaging.