

Titre: Développement d'outils d'intelligence artificielle pour la
caractérisation de lames histologiques marquées par nanoparticules
plasmoniques
Title:

Auteur: Julien Fillion
Author:

Date: 2024

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Fillion, J. (2024). Développement d'outils d'intelligence artificielle pour la
caractérisation de lames histologiques marquées par nanoparticules
plasmoniques [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/57356/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/57356/>
PolyPublie URL:

**Directeurs de
recherche:** Michel Meunier
Advisors:

Programme: Génie biomédical
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Développement d'outils d'intelligence artificielle pour la caractérisation de
lames histologiques marquées par nanoparticules plasmoniques**

JULIEN FILLION

Institut de génie biomédical

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie biomédical

Février 2024

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Développement d'outils d'intelligence artificielle pour la caractérisation de lames histologiques marquées par nanoparticules plasmoniques

présenté par **Julien FILLION**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Frédéric LEBLOND, président

Michel MEUNIER, membre et directeur de recherche

Nicolas PICHÉ, membre

REMERCIEMENTS

Je tiens à remercier ma famille proche, Karen-Powell Evans, Robert Fillion, Francis Fillion, Alexia Fillion et Gabrielle Gauthier-Roy pour leur support constant et leur conseil. Je tiens aussi à remercier Michel Meunier et les membres du laboratoire de plasmonique et de procédés laser de Polytechnique Montréal pour le support mental et technique constant. Merci à *Object Research Systems / Comet* de m'avoir accueilli à bras ouverts. Finalement je tiens à remercier par-dessus tout Cécile Darviot, sans qui ce projet n'aurait jamais vu le jour, pour son soutien constant et pour m'avoir fait découvrir la *startup Véga BioImaging*.

Cécile Darviot et Raphaële Fernet-Brochut m'ont permis d'avoir accès à des échantillons cellulaires et tissulaires marqués par nanoparticules plasmoniques et je les en remercie chaleureusement. Je souhaite aussi remercier les groupes de recherches associés, soit les groupes de Pr. Meunier et Pr. Trudel pour leurs travaux précédents sur le marquage de protéines par nanoparticules plasmoniques.

RÉSUMÉ

Introduction : En 2023, près de 29 400 Canadiennes furent diagnostiquées avec un cancer du sein. L'ensemble des patientes a subi une ou plusieurs biopsies servant à faire un test de détection de protéines appelé immunohistochimie (IHC). Le test permet d'établir le plan de traitement optimal pour chaque patiente. Malheureusement, cette méthode est subjective et dépend du pathologiste qui évalue l'échantillon. Le test induit donc un manque de concordance interobservateur important lors de l'évaluation de l'expression d'une protéine d'intérêt.

Le groupe de Pr. Meunier a développé une **solution** de quantification de protéines qui passe par l'utilisation de nanoparticules plasmoniques (NPs). Ces particules qui ciblent les protéines d'intérêt apparaissent ensuite comme des billes brillantes qu'il est possible de compter. Cela donne une mesure quantitative, mais implique qu'il faut compter les NPs attachées à l'échantillon. Cependant, la difficulté de détection des NPs par des algorithmes naïfs limite le potentiel de la méthode.

Objectif : Développer des approches de détection de NPs par intelligence artificielle (IA).

La réalisation de cet objectif permettra de valider l'**hypothèse** selon laquelle l'apprentissage machine est plus propice à la détection de NPs qu'un algorithme non entraîné.

Méthodologie : L'approche choisie fut de d'abord développer une banque de données de lames marquées par NPs en respectant les conditions d'acquisition d'images en pathologie. Ensuite afin de générer une annotation des images, la stratégie employée fut d'identifier le centre des NPs en utilisant une méthode de microscopie à plus haute résolution. Ainsi, différents algorithmes de segmentation sémantique de type *UNet* ont pu être entraînés à détecter les NPs sur les images de la banque de données.

Résultats : Cette approche a permis de déterminer les paramètres d'entraînement adéquat pour ce type de tâche de détection, puis d'effectuer une détection automatique avec une meilleure performance (*FScore* supérieur) que celle des algorithmes non entraînés. La comparaison des performances des différents modèles a permis une réflexion sur la pertinence des mécanismes d'attention dans le contexte de cette application ainsi que sur l'utilisation de ces algorithmes pour identifier des objets de très petite taille dans des images. Cette amélioration de la détection de NPs permettra d'augmenter la plage dynamique de cette méthode de quantification de protéines afin de mieux déterminer les plans de traitements de patients atteints du cancer du sein.

ABSTRACT

Introduction: In 2023, nearly 29 400 Canadian women were diagnosed with breast cancer. Every single one of these patients had a biopsy of their tumor taken to undergo a protein detection test called Immunohistochemistry (IHC). The results of this test were then used to establish the optimal treatment plan for the patient. Sadly, the current method is subjective and extremely reliant on the pathologist's evaluation of the sample. This yields error rates in the evaluation of patients that can go up to 30% for samples in which a protein is weakly expressed.

Pr. Meunier's group at Polytechnique Montréal developed a protein quantification **solution** that uses Plasmonic Nanoparticles (NPs). These particles are used as probes that target the desired proteins. The optical properties of these NPs allow them to appear as bright colorful beads that can then be counted. This enables a relative quantification of protein expression which comes with the task of counting a tremendous amount of NPs on each sample. To achieve this, naïve algorithms have been used, although, their performances are limited due to the inherent variability of the task. The poor performance of these algorithms limits the application potential of this method.

Objective: Develop nanoparticle detection solutions using AI.

The accomplishment of the above objective would allow us to validate the **hypothesis** that AI is more suitable than naïve algorithms for NP detection.

Methodology: The suggested method was to first develop a database of images of Nanoparticle-stained samples. Then, the center of each NP in the database was annotated using a higher resolution microscopy method. Once constructed and annotated, this database was used to train different semantic segmentation algorithms using *UNet* style architectures. These algorithms could then be used to detect NPs in different samples.

Results: The suggested approach allowed us to establish the optimal training parameters for this type of detection task. The resulting algorithms were able to detect nanoparticles in new samples with a higher overall performance than naïve algorithms. The comparison between different models yielded by various training parameters allowed us to discuss the relevance of attention mechanisms in this context. The improvements brought by these algorithms will increase the dynamic range of this protein quantification method, which would allow us to better establish breast cancer patients' treatment plans.

TABLE DES MATIÈRES

REMERCIEMENTS	III
RÉSUMÉ.....	IV
ABSTRACT	V
TABLE DES MATIÈRES	VI
LISTE DES TABLEAUX.....	IX
LISTE DES FIGURES.....	X
LISTE DES SIGLES ET ABRÉVIATIONS	XV
LISTE DES ANNEXES.....	XVII
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 REVUE DE LITTÉRATURE.....	5
2.1 Cancer du sein	5
2.1.1 Bases de l'oncologie et de l'immunologie	5
2.1.2 Traitements ciblés	7
2.1.3 Prédiction de la réponse au traitement	9
2.2 Quantification de protéines par Immunoplasmonique (IP)	16
2.2.1 Théorie des nanoparticules plasmoniques	16
2.2.2 Développement de sondes immunoplasmoniques.....	18
2.2.3 Imagerie et quantification.....	21
CHAPITRE 3 INTELLIGENCE ARTIFICIELLE EN SANTÉ	25
3.1 Concepts fondamentaux de l'intelligence artificielle	25
3.1.1 Introduction à l'apprentissage profond	25
3.1.2 Réseaux Neuronaux Convolutifs.....	29

3.2	Segmentation Sémantique	32
3.2.1	Hyperparamètres en segmentation sémantique	33
3.2.2	Applications récentes et stratégies et détections	35
CHAPITRE 4	MÉTHODOLOGIE.....	38
4.1	Protocole de préparation des échantillons	38
4.2	Préparation des données d'entraînement	39
4.2.1	Acquisition des images.....	39
4.2.2	Génération de masques de vérité 'automatiques'	41
4.2.3	Alignement des images et des masques	42
4.2.4	Ajustements préentraînements et augmentation de données.....	43
4.3	Entraînement des modèles	44
4.3.1	Paramètres Généraux des modèles.....	45
4.3.2	Paramètres d'entraînement.....	46
4.3.3	Résultats d'entraînements	48
4.4	Évaluation des modèles	48
4.5	Obtention du meilleur modèle	52
4.6	Représentation des détections.....	52
CHAPITRE 5	RÉSULTATS.....	54
5.1	Évaluation de l'algorithme naïf actuel – <i>Find Maxima</i>	54
5.2	Résultats d'entraînement	59
5.3	Évaluation préliminaire des modèles.....	62
5.3.1	Comparaison des hyperparamètres.....	66
5.4	Choix final du modèle	75
5.5	Résultats exploratoires.....	83

CHAPITRE 6	DISCUSSION	86
6.1	Discussion sur les hyperparamètres des modèles choisis	86
6.1.1	Effet de l'architecture	86
6.1.2	Effet de la structure d'encodage (<i>backbone</i>)	87
6.1.3	Effet de la taille de parcelles (<i>Patch</i>)	87
6.1.4	Effet de la taille de lot (<i>Batch</i>)	88
6.1.5	Effet de la fonction d'erreur (<i>Loss</i>) et de l'optimiseur	88
6.2	Viabilité d'un modèle d'apprentissage machine pour la détection de NPs	88
6.3	Limites de l'étude	89
CHAPITRE 7	CONCLUSION ET RECOMMANDATIONS	92
7.1	Opportunités futures et applications	92
RÉFÉRENCES		94
ANNEXES		99

LISTE DES TABLEAUX

Tableau 2.1 – Critères d’évaluation de l’expression de HER2 par IHC[15].....	12
Tableau 3.1 – Architectures convolutives	33
Tableau 4.1 – Paramètres d’acquisition des images basse résolution par la caméra PCO.....	40
Tableau 4.2 – Séparation des données pour l’entraînement, la validation et les tests	43
Tableau 4.3 – Paramètres généraux des modèles	46
Tableau 4.4 – Paramètres d’entraînement des modèles	47
Tableau 4.5 – Combinaisons d’hyperparamètres pour l’entraînement des modèles.....	47
Tableau 4.6 – Sommaire des modèles entraînés	48
Tableau 4.7 – Combinaisons d’hyperparamètres comparables pour la comparaison des modèles.....	52
Tableau 5.1 – Comparaison sommaire des FScores en fonction de l’architecture	70
Tableau 5.2 – Comparaison sommaire des hyperparamètres généraux par les moyennes des FScores de leurs modèles correspondants.....	71
Tableau 5.3 – Comparaison des moyennes de FScores des modèles correspondants pour les différents hyperparamètres d’entraînement.....	74
Tableau 5.4 – Sommaire des séparations pour l’entraînement fin des modèles	75
Tableau 5.5 – Comparaison des FScores Moyens des meilleurs modèles.....	78
Tableau 5.6 – Comparaison des performances moyennes entre le modèle choisi et <i>Find Maxima</i>	78
Tableau 5.7 – Comparaison du meilleur modèle à l’algorithme naïf <i>Find Maxima</i>	79
Tableau 5.8 – Comparaison des meilleurs modèles sans l’image test #4	79
Tableau 5.9 – Temps de prédiction	80
Tableau 5.10 – Quantification des détections par <i>Find Maxima</i> et par le modèle entraîné <i>Attention</i> <i>UNet - F</i>	82

LISTE DES FIGURES

Figure 1.1 – Échantillon cellulaire marqué par nanoparticules. Les nanoparticules observées par microscopie en illumination latérale apparaissent comme des points brillants. L’objectif utilisé pour cette acquisition est un objectif de microscopie à air de magnification 40X.....	2
Figure 2.1 - Tissu FFPE coloré par hématoxyline et éosine (H&E) où les noyaux sont colorés en mauve/bleu par l’hématoxyline et le cytoplasme des cellules est marqué en rose par l’éosine.	10
Figure 2.2 – Schéma de l’IHC.....	11
Figure 2.3 – Schéma de FISH	14
Figure 2.4 – Schéma d’une nanoparticule plasmonique sous l’effet d’un champ électrique externe.	17
Figure 2.5 – Nanoparticules de différentes couleurs pour la biodétection sur lames de microscopie. A) NPs d’alliage Or-Argent 90/10 de 43 nm, B) NPs d’or de 100 nm, C) NPs d’alliage Or-Argent 50/50 de 74 nm et D) <i>Nanorods</i> d’or de 25 x 71 nm. Note : Ces images sont des résultats non publiés fournis par Cécile Darvot.	19
Figure 2.6 – A) Schéma de microscopie par rétrodiffusion. B) Image de nanoparticules sur cellules en rétrodiffusion.	22
Figure 2.7 – A) Schéma du système d’imagerie en champ clair. B) Schéma du système d’imagerie en champ sombre. Échantillon cellulaire FFPE H&E marqué par IP C) acquis en champ clair et D) acquis en illumination latérale.	23
Figure 3.1 – Exemple de classification et de régression	26
Figure 3.2 – Exemple d’évolution de la fonction de décision pour un modèle de classification à 2 variables.	26
Figure 3.3 – Schéma de l’apprentissage profond inspiré de [40].	28
Figure 3.4 – A) Photo de nanoparticules sur cellules en microscopie avec une illumination latérale (40x à air). Agrandissement de nanoparticules situées B) sur une cellule et C) sur le substrat.	30

Figure 3.5 – Fonctions d’activations principales utilisées en segmentation sémantique	34
Figure 4.1 – NPs sur cellules en rétrodiffusion (microscopie 60x à huile) selon différentes couches d’acquisition d’un <i>z scan</i> . La distance en <i>z</i> séparant les plans focaux de ces deux images est inférieure à 1 micron.	41
Figure 4.2 - Images couleurs de nanoparticules sur cellules en illumination latérale, toutes les images provenant d’une acquisition en 40x ont été alignées aux masques (issus d’une microscopie en 60x). Ce faisant, les images ont des dimensions comparables à une image 60x. A) Image de test #1 - Cellules MCF-10A. B) Image de test #2 - Cellules T47D sans illumination bleue. C) Image de test #3 - Cellules MCF-7. D) Image de test #4 - Cellules MDA-231.	49
Figure 4.3 – Image de test #5 - Cellules SKBR-3 sans marquage H&E à haute concentration de NPs	50
Figure 4.4 – Schéma de l’évaluation des détections par rapport au masque de vérité manuel.	51
Figure 5.1 – Limites de détection de l’algorithme <i>Find Maxima</i> de NPs sur cellules SKBR-3 avec un seuil de tolérance au bruit optimisé pour l’image de droite. A) Image à haute densité de NPs. B) Image à faible densité de NPs.....	55
Figure 5.2 – Limites de détection de l’algorithme <i>Find Maxima</i> de NPs sur cellules SKBR-3 avec un seuil de tolérance au bruit optimisé pour l’image de gauche. A) Image à haute densité de NPs. B) Image à faible densité de NPs.....	56
Figure 5.3 – Comparaison de la précision (gauche) et du <i>recall</i> (droite) de <i>Find Maxima</i> sur les images de test #1 à #4 pour un intervalle de valeurs du paramètre de tolérance au bruit de l’algorithme naïf.	57
Figure 5.4 – <i>FScores</i> de l’algorithme naïf <i>Find Maxima</i> selon différents seuils de tolérance au bruit lors de la détection de NPs dans les images de test.....	58
Figure 5.5 – Comparaison de l’entraînement en fonction de l’architecture. Les courbes d’entraînement des modèles individuels sont en couleur. Les modèles utilisés pour cette comparaison sont les modèles dont la fonction d’erreur est la <i>Dice Loss</i>	60

- Figure 5.6 – Comparaison de l’entraînement en fonction de la taille de lot (*batch*). Les courbes d’entraînement des modèles individuels sont en couleur. Les modèles utilisés pour cette comparaison sont les modèles dont la fonction d’erreur est la *Dice Loss*..... 61
- Figure 5.7 – Comparaison de l’entraînement en fonction des structures d’encodages (*backbones*). Les courbes d’entraînement des modèles individuels sont en couleur. Les modèles utilisés pour cette comparaison sont les modèles dont la fonction d’erreur est la *Dice Loss*..... 62
- Figure 5.8 – Précisions moyennes (sur les quatre images de test) des 9 combinaisons d’hyperparamètres pour chacune des trois architectures principales (27 modèles évalués sur 4 images). 63
- Figure 5.9 - *Recalls* moyens (sur les quatre images de test) des 9 combinaisons d’hyperparamètres pour chacune des trois architectures principales (27 modèles évalués sur 4 images). 64
- Figure 5.10 – *FScores* moyens (sur les quatre images de test) des 9 combinaisons d’hyperparamètres pour chacune des trois architectures principales (27 modèles évalués sur 4 images). 64
- Figure 5.11 – Comparaison des performances des modèles sur l’image de test #2. 66
- Figure 5.12 – Exemple de comparaison de la performance (*FScore*) des modèles sur les images de test en fonction de leur architecture. Dans cet exemple, les performances (*FScores*) des 9 combinaisons d’hyperparamètres sur chacune des 4 images de test sont évaluées. Ceci signifie que pour UNet par exemple, il y a 9 x 4 points représentant les *FScores* obtenus. Au total, dans l’ensemble du graphique, il y a 27 x 4 points représentant des performances de modèles sur les images de test. 67
- Figure 5.13 – Comparaison des *FScores* de tous les modèles sur les 4 images de test en fonction des différents hyperparamètres généraux. De gauche à droite, les hyperparamètres évalués sont : 1- L’architecture, 2 – La structure d’encodage (*backbone*), 3 - La taille de parcelle (*patch size*). 68
- Figure 5.14 – *FScores* moyens de paires de modèles comparables selon les différents hyperparamètres généraux. Comparaison des *FScores* des modèles comparables sur les 4 images de test en fonction des différents hyperparamètres généraux. De gauche à droite, les

- hyperparamètres évalués sont : 1 – La structure d’encodage (*backbone*), 2 - La taille de parcelle (*patch size*)..... 69
- Figure 5.15 – Comparaison des hyperparamètres d’entraînements, de gauche à droite, les hyperparamètres testés sont : 1 – Fonction d’erreur (*Loss*), 2 – Optimiseur (*Optimizer*), 3 – Taille de lot (*Batch size*)..... 72
- Figure 5.16 – Comparaison des hyperparamètres d’entraînement par des paires de modèles comparables. De gauche à droite, les paramètres évalués sont : 1 – Fonction d’erreur (*Loss*), 2 – Optimiseur (*Optimizer*), 3 – Taille de lot (*Batch size*)..... 73
- Figure 5.17 – Comparaison des performances des modèles issus de 5 séparations de données différentes pour chacun des groupes d’hyperparamètres des modèles les plus performants (F – Bleu, H – Orange et I – Vert) sur chacune des 4 images de tests (de gauche à droite : image de test #1, image de test #2, image de test #3 et image de test #4). 76
- Figure 5.18 – Comparaison des modèles obtenus. À gauche, les 5 modèles (1 par séparation de donnée) entraînés avec les hyperparamètres F, H et I sont comparés. À droite, les modèles obtenus pour les 5 séparations de données sont comparés..... 77
- Figure 5.19 – Comparaison de la performance du modèle *Attention UNet F* et de *Find Maxima* pour une image à haute densité de NPs (image de test #5). Les détections par *Find Maxima* à un seuil de tolérance au bruit optimisé de valeur 7 sont en vert et les détections par le modèle entraîné sont en bleu. Ces comparaisons sont observables sur deux sections (A et B) de l’image de test #5..... 81
- Figure 5.20 – Agrandissement de la section de l’image de test #5 A) de la Figure 5.17 . Les cercles bleus représentent les détections effectuées par l’algorithme *Attention UNet – F*, tandis que les cercles verts représentent les détections effectuées par l’algorithme *Find Maxima* à un seuil de tolérance au bruit optimisé de 7..... 82
- Figure 5.21 – Méthode de marquage numérique de l’immunoplasmonique. Échantillon cellulaire marqué par NPs - A) en Illumination latérale, B) en Champ clair et C) en marquage artificiel. Échantillon tissulaire (HER2 positif) - D) en Illumination latérale, E) en champ clair et F) en marquage artificiel..... 84

Figure 5.22 - Marquage artificiel d'un échantillon tissulaire après une détection automatique des NPs par l'algorithme « intelligent » *Attention UNet* - F. Cette figure présente 3 différentes possibilités de représentation de la densité des NPs détectées sur les images en champ clair. Mêmes paramètres de la fonction *kdeplot()* décrits à la section 4.6 de la Méthodologie, en faisant varier la valeur du paramètre de bande passante A) $bw = 0,12$ B) $bw = 0,22$ et C) $bw = 0,32$ 85

LISTE DES SIGLES ET ABRÉVIATIONS

ADN	Acide désoxyribonucléique
AG	<i>Attention gate</i>
ANN	<i>Artificial neural network</i>
ARN	Acide ribonucléique
CAP	<i>College of American Pathologists</i>
CAR-T	Cellules T chimériques réceptrices d'antigène
CEP17	<i>Chromosome enumeration 17</i>
CMA	<i>Cell micro array</i>
CMH	Complexe majeur d'histocompatibilité
CNN	Convolutionnal neural network – Réseau Neuronal Convolutif
CV	Cross validation
DAB	Diaminobenzidine
DDT	Dichlorodiphényltrichloroéthane
DEL	Diode Électroluminescente
EDC	1-éthyl-3-(3-diméthylaminopropyl)carbodiimide
EGF	<i>Epidermal Growth Factor</i>
ER	Récepteur d'oestrogène
FCN	<i>Fully convolutional network</i> – Réseau entièrement convolutif
FDA	<i>U.S. Food and Drug Administration</i>
FFPE	<i>Formalin fixed paraffin embedded tissue</i>
FISH	<i>Fluorescent in situ hybridization</i>
FN	Faux négatif
FP	Faux positif

H&E	Hématoxyline et Éosine
HER2	<i>Human Epidermal Growth Factor Receptor 2</i>
ICI	<i>Immune Checkpoint Inhibitors</i>
IHC	Immunohistochimie
IA	Intelligence artificielle
IF	Immunofluorescence
IOU	<i>Intersection over union</i>
IP	Immunoplasmonique
ISH	<i>In situ hybridation</i>
NGS	<i>Next generation sequencing</i>
NHS	N-Hydroxysuccinimide
NP	Nanoparticule Plasmonique
PD-1	<i>Programed death receptor 1</i>
PDL-1	<i>Programed death receptor ligand 1</i>
PEG	Polyéthylène Glycol
PR	Récepteur de progestérone
QD	<i>Quantum dot</i> – Points quantiques
ReLU	<i>Rectilinear Unit</i>
ResNet	<i>Residual Network</i>
SGD	<i>Stochastic gradient descent</i>
SI	<i>Side Illumination</i> - Illumination latérale
T-Dxd	Trastuzumab Deruxtecan
TNBC	<i>Triple Negative Breast Cancer</i>
TILs	<i>Tumor infiltrating cells</i>
TMA	<i>Tissue micro array</i>
VGG	<i>Visual Geometry Group</i>
VP	Vrai positif

LISTE DES ANNEXES

ANNEXE A DÉFINITIONS INTELLIGENCE ARTIFICIELLE	99
ANNEXE B PROTOCOLE DE PRÉPARATION DES ÉCHANTILLONS CELLULAIRES .	102

CHAPITRE 1 INTRODUCTION

Au Canada, environ 1 femme sur 8 reçoit un diagnostic de cancer du sein au cours de sa vie et 1 femme sur 36 en décède [1]. L'entièreté des patients diagnostiqués subit une biopsie qui sert à déterminer le taux d'expression d'une certaine protéine dans la tumeur. Le résultat de ce test est ensuite utilisé afin d'établir le plan de traitement optimal pour chaque patient. Dans le cas du cancer du sein, 3 protéines (souvent des récepteurs que l'on peut appeler biomarqueurs) sont évaluées : le récepteur d'hormone de croissance de l'épiderme 2 (HER2), le récepteur d'œstrogène (ER) et le récepteur de progestérone (PR) [2].

La présence de ces différentes protéines sur les cellules tumorales du patient permet de déterminer son éligibilité à différents traitements. Ces thérapies dites ciblées seront discutées à la section 2.1.2. Parmi ces options de traitements, on compte les thérapies hormonales (voir section 2.1.2.1) et les immunothérapies (voir section 2.1.2.2).

Ce choix de traitement repose sur l'évaluation de la présence des protéines par une méthode appelée immunohistochimie (IHC) qui sera décrite plus en détail à la section 2.1.3.1. La méthode consiste à colorer l'échantillon en fonction de l'expression de la protéine ciblée. Cette coloration permet à un pathologiste d'évaluer visuellement l'échantillon afin d'estimer l'expression de la protéine. Toutefois, le manque d'aspect quantitatif de cette méthode de marquage est très problématique. En 2021, une étude publiée dans *JAMA Oncology* [3] souligne une concordance de seulement 26% entre les pathologistes lors de l'évaluation de certaines catégories d'expression de la protéine HER2. Ceci signifie qu'une proportion significative de patients ne sont pas évalués correctement lors de cette étape.

Afin de pallier ce problème, le laboratoire de plasmonique et de procédés laser de Polytechnique Montréal dirigé par Pr. Meunier a proposé une stratégie de détection de protéines innovante qui est quantitative et compatible avec les protocoles cliniques. Cette solution est basée sur l'utilisation de nanoparticules plasmoniques (section 2.2) fonctionnalisées qui ciblent les protéines d'intérêts [4]. Ces nanoparticules, grâce à leurs propriétés optiques qui seront discutées à la section 2.2.1, sont visibles en microscopie optique en champ sombre tel que montré à la Figure 1.1. Puisque les nanoparticules apparaissent comme des billes brillantes, il est possible de les compter afin d'obtenir une mesure quantitative de l'expression d'une protéine. De plus, en utilisant des nanoparticules de différentes couleurs, il est possible de détecter plusieurs protéines différentes sur un même

échantillon. Cette nouvelle méthode est donc quantitative, et permet de la détection multiplexée (cibler plusieurs protéines simultanément), tout en étant compatible avec les protocoles utilisés en clinique.

Bien que très performante, on anticipe que cette méthode soit toutefois limitée dans sa capacité à détecter chaque protéine de l'échantillon par la dynamique d'attachement, la gêne stérique entre nanoparticules et la résolution liée à la détection des nanoparticules. Ces limitations, une fois caractérisées et optimisées permettront d'établir la plage dynamique de la méthode. Plus cette plage dynamique est importante, plus la méthode permettra de détecter des différences entre les niveaux d'expressions de protéines des patients atteints du cancer du sein.

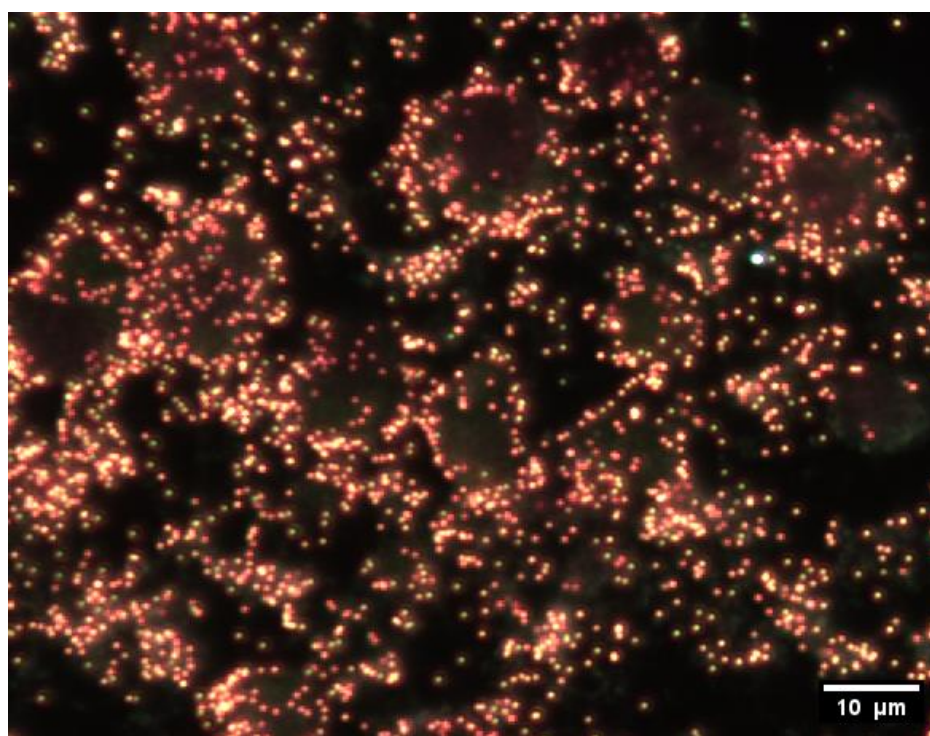


Figure 1.1 – Échantillon cellulaire marqué par nanoparticules. Les nanoparticules observées par microscopie en illumination latérale apparaissent comme des points brillants. L'objectif utilisé pour cette acquisition est un objectif de microscopie à air de magnification 40X.

La technologie offre l'avantage de pouvoir compter les nanoparticules afin d'obtenir une quantification relative de l'expression des protéines. Pour ce faire, le comptage peut être réalisé par l'utilisation d'algorithmes naïfs, toutefois, ce type d'algorithme est peu tolérant aux variations d'épaisseurs, de focalisation et d'intensités qui sont toutes présentes au sein d'un échantillon. Le manque de performance lors de l'étape de détection impose une limite artificielle à la plage

dynamique de la méthode. De plus, un algorithme naïf de ce type doit être ajusté en fonction de la quantité de nanoparticules sur l'échantillon. Cet aspect manuel représente un frein à une utilisation clinique, puisqu'il compliquerait la tâche des pathologistes qui sont déjà surchargés.

Pour pallier ce problème potentiel, il est proposé dans le cadre de cette maîtrise, d'utiliser l'intelligence artificielle pour la détection de nanoparticules sur des échantillons ayant subi le même traitement que des échantillons cliniques. En effet, l'intelligence artificielle a la capacité d'apprendre à reconnaître des structures dans des images, même lorsque les images sont soumises à des variations difficilement modélisables analytiquement (quelques exemples de variations: épaisseur d'échantillon variable, choix de plan focal variable, différentes densités de nanoparticules, etc.). De plus, une technologie de détection par IA qui serait entraînée adéquatement permettrait une détection ne nécessitant pas d'ajustements manuels entre les différents échantillons. Ceci faciliterait une adoption de la technologie en clinique.

Pour ce faire, différentes architectures de réseaux neuronaux convolutifs permettant d'effectuer de la segmentation pixel par pixel (segmentation sémantique) seront étudiées. Les architectures de ce type de réseaux neuronaux seront décrites à la section 3.2, puis les différents mécanismes d'attention permettant potentiellement d'augmenter la performance de ces algorithmes seront discutés à la section 3.2.2.

Au CHAPITRE 4, la méthodologie employée pour la génération d'une banque de données d'images de microscopie et la méthodologie d'entraînement de modèles de type *UNet*, *TransUNet* et *Attention UNet* sera décrite. Cette génération de banque de données passe par la création de masques de vérités (*ground truth*) provenant d'une méthode de microscopie à haute résolution. Ensuite, au CHAPITRE 5, les résultats seront analysés afin de pouvoir tirer des conclusions quant à l'impact des différents hyperparamètres sur les modèles résultants. Notamment, les modèles obtenus seront comparés par la structure d'encodage (le *backbone*), la taille de parcelle (*patch size*), la taille de lot (*batch size*), la fonction d'erreur (*Loss*) et l'optimiseur qui les composent. Ensuite, une comparaison des performances de ces modèles aux performances d'algorithmes naïfs permettra de constater une meilleure performance des modèles « intelligents ». Les impacts des différents hyperparamètres et le potentiel d'application de cette méthode de détection automatique seront discutés au CHAPITRE 6, où nous reviendrons sur les principaux paramètres ayant influencé les algorithmes de détection automatisée. Nous discuterons aussi des avantages et inconvénients de

l'utilisation d'algorithmes comme ceux-ci dans un contexte de détection de nanoparticules. Cette discussion permettra de conclure que les algorithmes « intelligents » sont souhaitables puisqu'ils permettent :

- 1- Une détection de NPs plus performante que les algorithmes naïfs utilisés actuellement.
- 2- Une détection adéquate pour des échantillons à haute densité de NPs, ce qui représente potentiellement une augmentation de la plage dynamique de la méthode de marquage par nanoparticules.
- 3- Une utilisation clinique plus accessible, puisque ces algorithmes ont le potentiel d'effectuer des détections sans être affectés par les variations mineures dans les images acquises. Toutefois, avant d'atteindre ce stade, les algorithmes devront être entraînés sur de plus vastes banques de données.

Au CHAPITRE 7, la conclusion nous permettra de réfléchir sur les prochaines étapes de cette technologie de détection automatique de NPs et sur les utilisations cliniques de l'IA dans le contexte de cette technique de marquage de protéine prometteuse.

CHAPITRE 2 Revue de littérature

2.1 Cancer du sein

Au cours des dernières années, de grands efforts ont été déployés dans la recherche contre le cancer. Les efforts ont permis le développement de plusieurs nouveaux traitements [5]. Malgré cela, la différence entre chaque tumeur au niveau cellulaire est trop importante pour permettre l'utilisation du même traitement pour tous. Les seuls traitements généralisables sont généralement des chimiothérapies qui s'attaquent aux mécanismes mêmes de reproduction de nos cellules et qui résultent en des effets secondaires importants. C'est pourquoi les traitements développés dans les dernières années sont en grande majorité des thérapies ciblées qui sont seulement applicables à des patients dont les cellules cancéreuses expriment des protéines bien précises. Avec l'arrivée de ces traitements très performants, mais auxquels seulement une petite partie de la population est éligible, il est crucial d'avoir accès à des diagnostics de pointe permettant de déterminer l'éligibilité des patients à un traitement plus performant.

2.1.1 Bases de l'oncologie et de l'immunologie

2.1.1.1 Origine d'une tumeur

L'équilibre du corps aussi appelé homéostasie est régulé en partie par les interactions entre les cellules. Les cellules individuelles de notre corps ont la capacité de communiquer entre elles, par différents moyens tels que l'interaction directe, l'utilisation de gradients de concentrations et l'utilisation de chimiokines afin de faire fonctionner les différents organes qui nous composent. Nos cellules possèdent donc de nombreux récepteurs qui leur permettent d'interagir avec les autres cellules. Ces interactions leur permettent de s'autoréguler en augmentant ou en diminuant leur prolifération afin de permettre le bon fonctionnement de nos organes et de notre corps.

Les récepteurs cellulaires sont des protéines. Ils sont synthétisés par certaines composantes de la cellule appelées ribosomes qui traduisent des séquences d'ARN correspondant à des gènes. Les gènes correspondent à des régions de notre ADN qu'il est possible de trouver, bien compacter dans des chromosomes, dans le noyau de chacune de nos cellules.

Dépendamment de leur fonction, les récepteurs dont la recette est encodée dans les gènes de la cellule seront plus ou moins exprimés en fonction des besoins de notre corps. Les gènes associés à des récepteurs qui permettent cette autorégulation peuvent être classés en 3 catégories : 1- les gènes

qui induisent la prolifération cellulaire, 2- les gènes qui inhibent la prolifération cellulaire et 3- les gènes qui régulent la mort cellulaire programmée aussi appelée apoptose [6].

Au cours de nos vies, des interactions avec le monde qui nous entoure peuvent affecter de manière aléatoire l'ADN de nos cellules. On parle ici de carcinogènes qui incluent différents produits chimiques (tels que le formaldéhyde, le DDT et certains pesticides), des agents physiques (tels que l'amiante) et des radiations ionisantes [6]. Ces altérations de notre ADN n'ont souvent aucun impact sur nous, puisque nous possédons des mécanismes de réparation de l'ADN et que certaines parties de notre ADN ont peu ou aucun impact sur nos cellules. Dans de très rares cas, des altérations de l'ADN ne sont pas détectées et ont un impact sur les protéines que produit la cellule. Encore une fois, dans de très rares cas, il arrive que ces protéines modifient réellement le comportement de la cellule. Heureusement, notre système immunitaire peut alors détecter ce mauvais comportement lorsqu'il vient valider l'identité de la cellule par la présentation d'un complexe majeur d'histocompatibilité (CMH) [6].

Les tumeurs naissent lorsque les modifications apportées à notre ADN impliquent l'expression de certaines protéines qui modifient les mécanismes d'autorégulation de nos cellules.

Dans le cadre de ce mémoire, deux de ces récepteurs seront particulièrement pertinents. Le premier, le *Human Epidermal Growth Factor Receptor 2* (HER2) agit sur la prolifération cellulaire et est relié au récepteur d'hormone de croissance de l'épiderme (EGF) [6]. Il peut être surexprimé dans certaines tumeurs, engendrant ainsi une croissance anormale des cellules tumorales. Le second, le *programed death receptor 1* (PD-1) ou son ligand le *programed death receptor ligand 1* (PD-L1) peut aussi être surexprimé pour les cellules tumorales qui lors de l'interaction avec une cellule immunitaire (Cellule T) inhibe la réponse des cellules immunitaires [7]. Dans cette situation, la tumeur prolifère sans que le système immunitaire ne puisse intervenir. Nous traiterons aussi des récepteurs hormonaux d'œstrogène (ER) et de progestérone (PR) qui sont souvent ciblés dans le traitement du cancer du sein puisqu'ils augmentent la prolifération cellulaire [2].

2.1.1.2 Prise en charge du patient

Tout d'abord, il faut comprendre que le diagnostic et le traitement du cancer du sein se divisent en plusieurs étapes. Lorsqu'une tumeur est suspectée pour un patient, on doit procéder à une étape de diagnostic qui implique généralement une mammographie suivie d'une biopsie permettant d'évaluer l'agressivité de la tumeur [8].

Une fois la tumeur confirmée, l'oncologue doit, en fonction du stade d'avancement de la tumeur, établir le plan de traitement. Pour ce qui est des cancers du sein non métastatiques qui doivent être retirés par chirurgie (ou radiothérapie). Les étapes de traitement sont les suivantes [8] :

1 - La première étape est de freiner le développement de la tumeur. On réalise cela par l'administration de chimiothérapie ou de la thérapie hormonale au patient. On parle alors d'un traitement néoadjuvant.

2 - Ensuite, la seconde étape consiste à la médication qui sera donnée suite à l'extraction de la tumeur pour éviter une récurrence. On parle alors d'un traitement adjuvant. Ces traitements peuvent consister de thérapie hormonale, de chimiothérapie et/ou d'immunothérapies.

3 - Finalement, dans les cas où la tumeur se propage dans le reste du corps, on considère alors qu'elle entre en phase métastatique. On peut alors donner d'autres traitements qui freinent la prolifération. Les traitements que l'on peut donner à cette étape sont les immunothérapies, la thérapie hormonale et la chimiothérapie.

2.1.2 Traitements ciblés

2.1.2.1 Thérapie hormonale

La thérapie hormonale est donnée aux patients qui sont considérés positifs aux récepteurs hormonaux (ER et PR) [9] après un test d'IHC. Le traitement administré comme adjuvant aux patients (pré ou post-ménopause) est le Tamoxifène. Pour les patientes post-ménopause, Anastrozol, Letrozol et Exemestan peuvent être utilisés. L'objectif de ces traitements est de bloquer l'interaction entre l'œstrogène et les mécanismes associés qui provoquent une stimulation de la prolifération cellulaire. Près de 50% des cancers positifs aux récepteurs hormonaux sont ou deviennent résistants à la thérapie hormonale [9]. Les patients qui reçoivent le tamoxifène verront diminuer leur risque de récurrence du cancer de 41% et verront diminuer leur risque de mort due au cancer du sein de 34% [9]. Les effets secondaires associés à cette thérapie sont le cancer de l'endomètre, les cancers utérins et un risque de thromboses [9]. Il est ainsi crucial d'établir avec précision l'expression de ces marqueurs chez les patients afin d'augmenter les bénéfices possibles du traitement, tout en s'assurant que des patients non éligibles ne reçoivent pas un traitement pouvant engendrer des effets secondaires graves.

2.1.2.2 Immunothérapies

Les immunothérapies sont dites personnalisées, puisqu'elles ciblent une seule protéine des cellules de la tumeur [5]. Il faut donc que les cellules tumorales expriment cette protéine pour que le traitement soit efficace. Ces traitements ne peuvent pas non plus être administrés à tout le monde sans considération de l'efficacité potentielle du traitement, puisque le traitement peut avoir plus d'impacts négatifs que positifs pour un patient qui n'exprimerait pas la protéine d'intérêt [10].

Les immunothérapies actuellement utilisées pour traiter le cancer du sein sont utilisées comme traitements adjuvants ou comme traitements métastatiques [5]. La cible de l'immunothérapie la plus commune est le récepteur cellulaire HER2. Toutefois, de nouvelles approches plus modernes incluent l'utilisation de thérapies ciblant PD-1 ou son ligand PD-L1 lorsque la tumeur n'exprime pas de récepteurs hormonaux ou de récepteurs HER2 (tumeur triple négative) [5]. En général, même si les thérapies de ce type sont souvent efficaces, elles sont généralement couplées à de la chimiothérapie afin d'augmenter les chances de survie des patients.

Les récepteurs HER2 sont ciblés par différents types de thérapies. Certaines d'entre elles, comme que le *Trastuzumab*, ciblent HER2 en envoyant des anticorps qui se lient à HER2 et qui bloquent ainsi l'activation potentielle du récepteur qui aurait engendré une prolifération cellulaire. Cette approche permet ainsi de limiter la prolifération de la tumeur. Un traitement récemment approuvé pour les tumeurs métastatiques, le *Trastuzumab Deruxtecan* (T-Dxd), représente un autre type de thérapie qui cible aussi HER2 [11]. T-Dxd est composé d'anticorps porteur d'une substance toxique pour la cellule. Ainsi, lorsqu'ils se lient au récepteur HER2, la cellule internalise les anticorps, puis la substance toxique induit l'apoptose de la cellule [12]. Cette thérapie est maintenant approuvée pour tous les patients métastatiques exprimant HER2, que le récepteur soit surexprimé ou exprimé normalement.

Pour le PD-L1 / PD-1, le traitement proposé par les récentes études cliniques est le *Pembrolizumab* [5]. Ce traitement vient bloquer le PD-L1, bloquant ainsi un récepteur qui inhibe l'activation du système immunitaire. La cellule peut donc être détectée et éliminée par le système immunitaire du patient. Les thérapies ciblant PD-L1 sont déjà très utilisées pour le traitement du cancer du poumon, puis font tranquillement leur place dans les thérapies du cancer du sein. En 2020, la combinaison de la chimiothérapie Paclitaxel avec le *Pembrolizumab* a été approuvée par la FDA pour les patients positifs au PD-L1 [5].

En résumé, la présence d'HER2 permet l'éligibilité du patient aux traitements adjuvants et métastatiques qui ciblent HER2, telles que *Trastuzumab* et *Trastuzumab Deruxtecan*. La présence d'ER et PR permet d'établir l'éligibilité du patient à la thérapie hormonale comme traitement néoadjuvant, adjuvant ou métastatique. Dans les cas où la tumeur n'exprime aucun de ces trois marqueurs, on parle d'un cancer du sein triple négatif (TNBC – *triple negative breast cancer*) [8]. Il existe peu de traitements ciblés pour ces patients, mais les immunothérapies de types *immune checkpoint inhibitors* (ICI) tels que *Pembrozimulab* (PD-1/PD-L1) sont de plus en plus utilisées et approuvées par les agences réglementaires comme la FDA [5]. Ils sont utilisés en combinaison avec d'autres traitements tels que la chimiothérapie.

La recherche ne cesse de progresser afin de découvrir de nouvelles approches ayant un impact sur les cancers du sein. Parmi les nouvelles approches, on compte des combinaisons de traitements immunothérapie/chimiothérapies, l'utilisation d'anticorps bispécifiques (BsAbs), l'utilisation de thérapies impliquant des TILs (*tumor infiltrating cells*) ainsi que l'utilisation de cellules T chimériques réceptrices d'antigène (CAR-T). « En 2022, 745 essais cliniques basés sur l'immunothérapie en oncologies étaient en cours. De ce nombre, 450 (60,4%) étaient exclusivement dédiées au cancer du sein » [5]. Cela dit, la prédiction de la réponse à ces traitements est un aspect majeur de la lutte contre le cancer du sein qui doit être à la hauteur des nouvelles thérapies.

2.1.3 Prédiction de la réponse au traitement

Cette panoplie de traitements potentiels qui sont destinés à des groupes de patients très ciblés fait augmenter le besoin pour des analyses plus précises des tumeurs. Les méthodes d'analyses qui permettent de prédire la réponse au traitement d'un patient se concentrent soit sur la détection des protéines ou sur la détection de l'ADN associé à cette protéine. Le standard clinique utilisé par les pathologistes afin d'évaluer les échantillons est l'immunohistochimie (IHC) [8]. Cette méthode permet d'évaluer l'expression de protéines dans les biopsies de tumeurs. Il existe d'autres méthodes utilisées en cliniques telles que le *fluorescent in situ hybridization* (FISH), le *bright field* ISH et le *Next Generation Sequencing* (NGS) qui détectent plutôt le gène associé à la protéine [13]. La plupart de ces méthodes (sauf NGS) ont l'avantage de permettre une visualisation de la morphologie des tissus lors du test. Cette caractéristique est particulièrement importante pour les pathologistes qui prennent fréquemment le contexte tissulaire en compte lorsque vient le temps

d'effectuer une décision quant à l'échantillon. À cette fin, les échantillons sont préparés selon un protocole bien précis de coloration et de fixation [13].

Chaque échantillon est d'abord soumis à une fixation au formol pour sa conservation, suivi d'une déshydratation effectuée à l'aide de solutions alcoolisées. Ensuite l'échantillon est enrobé de paraffine afin de former un bloc de paraffine, on parle alors d'un échantillon fixé au formol et enrobé de paraffine (*Formalin Fixed Paraffin Embedded Tissue* (Tissu FFPE)).

Afin d'effectuer la coloration des cellules, le bloc est ensuite tranché en sections d'environ $4\mu m$, puis déposé sur une lame de microscopie. Il peut ensuite être déparaffiné, puis réhydraté. Finalement, le tissu est coloré par l'hématoxyline qui marque les noyaux cellulaires en bleu et par l'éosine qui marque le cytoplasme et la matrice extracellulaire en rose [14]. L'échantillon est ensuite apprêté pour la visualisation au microscope. La coloration est appelée coloration par Hématoxyline et Éosine (H&E). La Figure 2.1 permet de voir un échantillon de cancer du sein FFPE coloré par H&E.

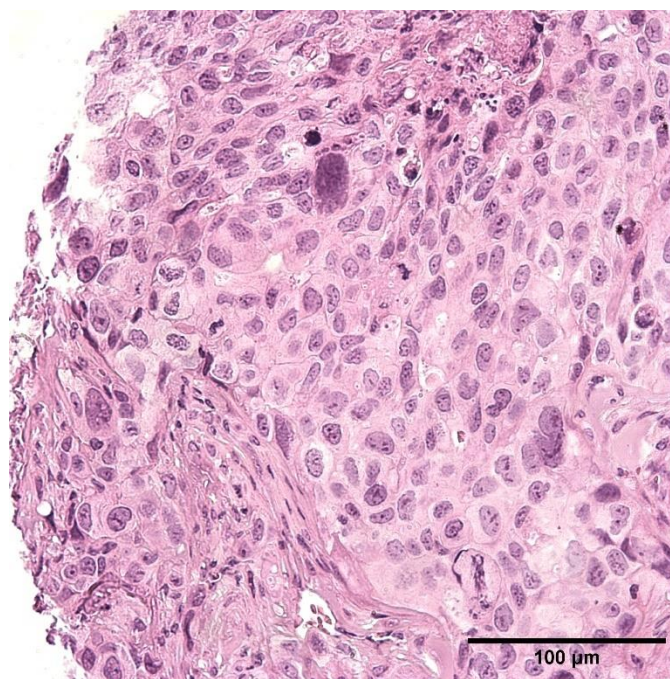


Figure 2.1 – Tissu de cancer du sein FFPE coloré par hématoxyline et éosine (H&E) où les noyaux sont colorés en mauve/bleu par l'hématoxyline et le cytoplasme des cellules est marqué en rose par l'éosine.

2.1.3.1 Immunohistochimie

L'immunohistochimie est une méthode de détection de protéines. Selon les protocoles actuels, les échantillons sont fixés dans de la paraffine, puis découpés en tranches de 4 microns d'épaisseur afin d'être déposés sur une lame de microscopie. Le marquage par IHC est basé sur l'utilisation d'un anticorps primaire qui cible une protéine d'intérêts (ex. : HER2 peut être ciblé par l'anticorps polyclonal A085 de DAKO ou par le CB11 monoclonal de Ventana) [13]. Par la suite, l'excès d'anticorps est lavé, puis un anticorps secondaire cible la chaîne non spécifique (voir Figure 2.2) de l'anticorps primaire. Ce second anticorps est muni d'une chaîne polymère sur laquelle est attaché un agent chimique (*horseradish peroxydase* (HRP)) qui peut interagir par une réaction enzymatique avec un chromogène (ex.: diaminobenzidine (DAB)), ce qui résulte en un précipité brun qui marque les tissus. Plus la protéine est présente dans l'échantillon, plus la coloration sera intense.

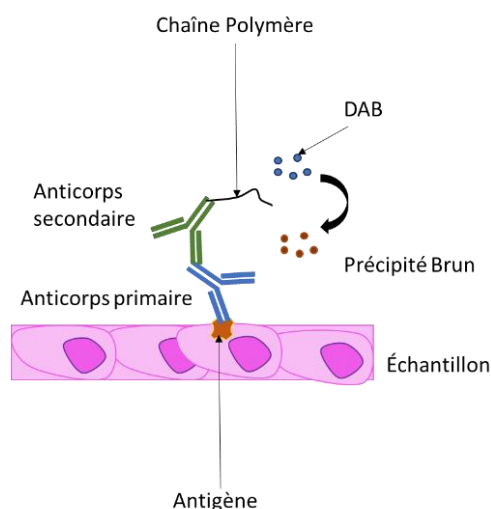


Figure 2.2 – Schéma du fonctionnement de l'IHC

L'intensité de la coloration dépend donc de l'expression de la protéine dans l'échantillon. Les pathologistes doivent ensuite évaluer l'intensité de la coloration par une échelle qualitative à 4 valeurs [0, 1+, 2+, 3+] où 0 est négatif et 3+ correspond à une surexpression évidente de la protéine[15]. Ces catégories provenant des recommandations de *l'American Society of Clinical Oncology – College of American Pathologists* 2023 [15] sont décrites au Tableau 2.1.

Tableau 2.1 – Critères d’évaluation de l’expression de HER2 par IHC[15]

Catégorie	Critère
3+ Positif	Coloration membranaire circonférentielle complète, intense et dans plus de 10% des cellules tumorales
2+ Équivoque	Coloration membranaire complète faible ou modérée dans plus de 10% des cellules tumorales
1+ Négatif	Coloration membranaire incomplète qui est faible ou quasi imperceptible dans plus de 10% des cellules tumorales
0 Négatif	Aucune coloration observée ou Coloration membranaire incomplète qui est faible ou quasi imperceptible dans moins de 10% des cellules tumorales

Pour les traitements comme le *Trastuzumab* qui ciblent une surexpression de la protéine, il est important de procéder à un second test dans les cas équivoques. Ainsi, lorsque l’échantillon est classé dans la catégorie 2+, il doit automatiquement être testé par FISH [15] (méthode de quantification d’ADN décrite à la section 2.1.3.2). Le résultat du test FISH permettra de classer les échantillons entre positifs et négatifs à une surexpression de la protéine. Les patients considérés comme positifs sont éligibles au traitement *Trastuzumab* décrit à la section Immunothérapies 2.1.2.2. Les patients qui sont à un stade métastatique et dont la tumeur exprime faiblement la protéine ne sont pas éligibles au traitement *Trastuzumab*, mais peuvent avoir accès au traitement T-Dxd [11]. Il faut donc être capable de faire la différence entre les niveaux d’expressions de la protéine afin de déterminer si un patient est considéré comme exprimant fortement la protéine (IHC 3+ ou 2+ avec confirmation FISH), comme exprimant faiblement la protéine (IHC 2+ sans

confirmation FISH ou 1+, aussi appelé *HER2-Low*) ou si le patient n'exprime tout simplement pas la protéine (IHC 0).

Quoique cette méthode soit particulièrement pratique à cause de sa facilité d'utilisation, sa stabilité dans le temps et sa facilité d'observation, elle ne permet pas de classer les patients avec précision selon leur expression de la protéine. L'IHC manque d'aspect quantitatif. Dans le cas du cancer du sein, le diagnostic de HER2 est à un point critique. Depuis l'approbation du traitement T-Dxd en 2021, les patients dont l'expression est classée *HER2-Low* (2+ sans confirmation FISH et 1+) doivent être différenciés des patients sans expression de la protéine (IHC 0). Cependant selon des données du *College of American Pathologists* (CAP) analysées par une étude publiée dans *JAMA Oncology* en 2022 [3], 19% des cas généreront des résultats avec 70% ou moins de concordance entre les scores 0 et 1+. Toutefois, en analysant une seconde base de données les chercheurs font état de seulement 26% de concordance lors de la classification entre des échantillons 0 et 1+ [3]. Cette concordance monte seulement à 58% entre les cas 2+ et 3+. Ce manque de concordance est flagrant et démontre le besoin pour différentes approches de détection de protéines.

2.1.3.2 *Fluorescent In Situ Hybridization* (FISH)

Le FISH est considéré comme étant plus précis que l'IHC [16]. Le FISH est aussi utilisé sur des échantillons tranchés, comme l'IHC. La méthode consiste à chauffer l'échantillon pour séparer son ADN, à perméabiliser les cellules et à cibler le gène associé à HER2 avec une sonde fluorescente [13]. Le FISH peut être effectué avec une seule sonde qui cible le gène HER2 (voir Figure 2.3) ou avec deux sondes (*dual probe*). La deuxième sonde cible alors le CEP17 (*chromosome enumeration 17*) qui est le chromosome sur lequel peut se trouver le gène HER2. Cette deuxième sonde sert alors de contrôle. Les tests à deux sondes permettent d'obtenir une quantification relative du nombre de copies de gènes correspondant à HER2 ainsi que le nombre de CEP17. Il est ensuite possible d'établir un ratio. Si ce dernier est supérieur à 2.2 signal/cellule, l'échantillon est classé positif, s'il est inférieur à 1.8 signal/cellule, il est considéré négatif et s'il se trouve entre les deux, il est considéré comme équivoque [15].



Figure 2.3 – Schéma de FISH

2.1.3.3 *Bright-Field* ISH

Le FISH a comme grand défaut de dépendre d'une évaluation de fluorescence, qui peut être laborieuse, puisqu'elle implique de travailler dans un environnement sombre et qu'elle implique le photoblanchiment de l'échantillon lorsqu'il est illuminé. C'est pourquoi d'autres méthodes de détection d'ADN ont vu le jour. Ces techniques consistent à envoyer une sonde d'ADN dotée d'un composé chimique visible en champ clair. La première méthode, le *chromogenic* ISH (CISH), cible les copies de gènes avec un chromophore tel que le DAB qui est aussi utilisé en IHC [13]. La seconde méthode le Silver-enhanced ISH (SISH) utilise de l'argent (Ag) pour marquer l'échantillon [17]. Il est aussi possible de faire du *dual probe*, comme pour le FISH, en utilisant SISH pour cibler le gène HER2 et CISH pour cibler le chromosome CEP17. Le tout peut ensuite être visualisé au microscope en champ clair, ce qui permet d'éviter les problèmes liés à la fluorescence, tout en évitant d'analyser des sections qui ne sont pas reliées à la tumeur (puisque la morphologie des tissus est visible).

2.1.3.4 *Next Generation Sequencing* (NGS)

Le NGS est approuvé depuis quelques années par la FDA [13]. Cette méthode consiste à une extraction de l'ADN des cellules de l'échantillon, à un séquençage, puis à une identification des séquences d'ADN associées au gène HER2. Cette méthode donne des résultats qui concordent avec les résultats de FISH et sont quantitatifs. Toutefois, la méthode semble diminuer en performance lorsque les cellules tumorales représentent moins de 25% des cellules de l'échantillon [18] et la méthode ne permet pas une observation simultanée de la morphologie des cellules de la tumeur.

2.1.3.5 Méthodes non approuvées par la FDA

De nouvelles approches comme l'utilisation de *Quantum Dots* (QD) comme sonde de détection sont apparues dans les dernières années pour tenter de pallier ce problème de détection. Sans rentrer dans les détails de comment fonctionne un QD, c'est un nanocristal semi-conducteur qui émet de la fluorescence [13]. L'avantage des QDs par rapport aux fluorophores que l'on utilise fréquemment passe par le fait que leur taille et leur longueur d'onde d'excitation sont modifiables en adaptant les conditions de synthèse. De plus, les QDs sont beaucoup plus résistants au photoblanchiment, ils sont plus visibles que les fluorophores et permettent aussi d'augmenter la quantité de signal obtenu par rapport à l'auto-fluorescence du tissu [19]

Comme autre solution potentielle, la spectrométrie de masse a aussi été proposée, mais bien qu'elle permette d'obtenir une mesure quantitative de l'expression de la protéine HER2, son utilisation implique la destruction de l'échantillon et il est possible que la préparation de l'échantillon ait un impact sur la quantification [20].

En résumé, les différentes méthodes décrites dans l'entièreté de la section 2.1.3 sur la prédiction de la réponse au traitement évaluent donc par 2 approches potentielles la surexpression d'HER2. L'IHC, les QDs et la spectrométrie de masse passent par l'identification des protéines, tandis que le FISH, le Bright-Field ISH et le NGS passent par l'identification des gènes. Toutefois, les méthodes qui passent par la détection de séquences d'ADN/ARN peuvent engendrer de faux positifs et de faux négatifs. Dans certains cas, la surexpression du gène ne veut pas dire que la protéine sera réellement surexprimée, ce qui fait en sorte que le taux de faux négatifs peut aller jusqu'à 4% et que le taux de faux positifs peut monter jusqu'à 20% [21]. Dans le cas des méthodes qui ciblent la protéine, l'évaluation est subjective et non quantitative pour l'IHC et les QDs, puis la destruction des échantillons engendrée par la spectrométrie de masse est problématique pour une utilisation en clinique.

Les limitations de ces méthodes sont, dans la plupart des cas, inhérentes à la technologie utilisée pour effectuer la détection de l'expression des protéines. Afin d'augmenter la précision de ces tests qui permettent de prédire la réponse au traitement d'un patient, de nouvelles méthodes de quantifications doivent être développées.

2.2 Quantification de protéines par Immunoplasmonique (IP)

Le laboratoire de plasmonique et de procédés laser de Polytechnique Montréal dirigé par Pr. Meunier propose une solution innovante au problème de la détection de protéines pour prédire la réponse au traitement des patients atteints de cancer du sein. Cette solution consiste en un marquage d'échantillons fixés en paraffine – *Formalin-Fixed Paraffin-Embedded* (FFPE) – par immunoplasmonique (IP).

L'IP [22] est une méthode de marquage qui s'apparente à des méthodes comme L'IHC ou l'Immunofluorescence (IF) qui ciblent une protéine à l'aide d'anticorps et qui ajoutent un marqueur visible à l'échantillon. Toutefois, cette nouvelle approche s'affranchit du défaut principal des méthodes classiques, soit leur manque d'aspect quantitatif. En effet, l'IP consiste à utiliser des nanoparticules plasmoniques (NPs) sur lesquelles ont été attachés des anticorps qui ciblent la protéine d'intérêt. Les NPs peuvent ainsi s'attacher aux protéines dans l'échantillon et, grâce à leurs propriétés optiques exceptionnelles, peuvent être visualisées en microscopie en champ sombre [23]. Une fois les NPs comptées, la méthode permet d'obtenir une mesure quantitative de l'expression de la protéine évaluée afin de mieux prédire la réponse des patients aux traitements ciblés.

2.2.1 Théorie des nanoparticules plasmoniques

Les nanoparticules plasmoniques (NP) sont étudiées depuis plusieurs années pour leur interaction particulière avec la lumière. Ces NPs composées de métaux nobles sont principalement faites d'Or (Au), d'argent (Ag) ou de platine (Pt) [23]. Les électrons contenus dans ces particules sont libres de se déplacer puisque les métaux en question sont conducteurs. Les électrons se déplacent ainsi librement selon les champs électriques externes qui sont appliqués à la NP, tels que les ondes électromagnétiques d'une source lumineuse. Lorsqu'une onde électromagnétique provenant d'une source lumineuse est envoyée sur la NP, le champ électrique appliqué à la particule varie de manière oscillatoire. Ceci provoque un mouvement des électrons dans la NP qui fait en sorte que l'énergie des photons incidents est transmise en grande partie à la NP (voir Figure 2.4).

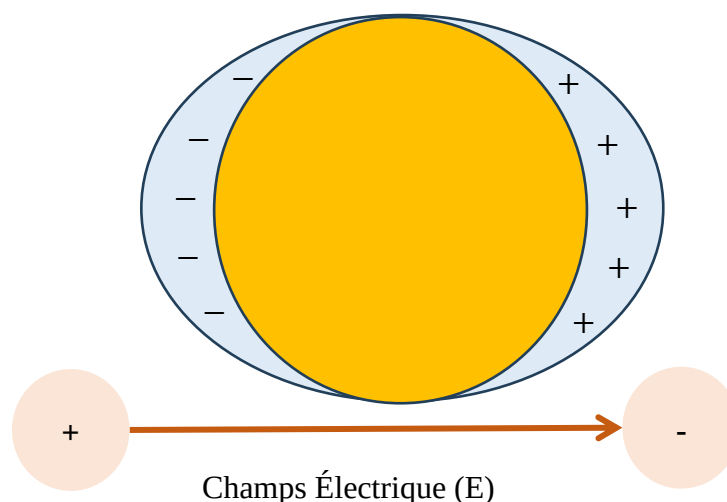


Figure 2.4 – Schéma d'une nanoparticule plasmonique sous l'effet d'un champ électrique externe.

La mesure utilisée pour décrire cette transmission d'énergie entre la lumière incidente et l'objet est appelée « section efficace d'extinction » (σ_{ext}) et permet de quantifier la surface de lumière qui – une fois multipliée par l'irradiance des ondes électromagnétiques incidentes – donne le flux de radiance absorbé et diffusé par la nanoparticule [24]. Cette transmission d'énergie entre la lumière incidente et la nanoparticule est caractérisée par deux phénomènes. Premièrement, l'oscillation simultanée des électrons (excitation / désexcitation) génère des ondes électromagnétiques (lumière diffusée) moins énergétiques que les ondes incidentes. Deuxièmement, la résistance du matériau au déplacement des électrons génère de l'énergie thermique. Ainsi, la surface d'extinction associée à une nanoparticule est donnée par la somme de l'absorption et de la diffusion de la nanoparticule : $\sigma_{ext} = \sigma_{diffusion} + \sigma_{absorption}$. Si une NP est irradiée par une source lumineuse dont la longueur d'onde correspond à la longueur d'onde de résonance de la nanoparticule le ou les dipôles générés par le déplacement des électrons (le nombre de dipôles dépend de la taille et la forme des nanoparticules) deviendront constants spatialement, mais varieront dans le temps, au rythme des ondes incidentes. Ce phénomène fait en sorte qu'un pic plasmonique (*Localized Surface Plasmon Resonance Peak*) est atteint et que les propriétés d'extinction de lumière de la nanoparticule sont à leur maximum [23].

La proportion de lumière absorbée ou diffusée et la longueur d'onde du pic plasmonique dépendent d'un grand nombre de facteurs tels que la composition de la nanoparticule, sa taille, sa forme et le

milieu qui l'entoure [4]. En ajustant ces différents paramètres, les nanoparticules peuvent servir à différentes applications qui bénéficient d'une maximisation de l'absorption, telle que les applications d'hyperthermie, et d'autres qui bénéficient d'une maximisation de la diffusion, comme les applications de biodétection [23]. Le groupe de Pr. Meunier a développé un algorithme qui utilise la théorie de Mie afin de prédire la réponse des nanoparticules à une lumière incidente à l'aide des paramètres de composition, d'environnement et de forme de celle-ci. Ce programme appelé *NFMie* [25], [26] permet de mieux choisir les nanoparticules pour une application donnée.

Afin de détecter des protéines telles qu'HER2, il est important de choisir des nanoparticules qui diffusent dans le spectre du visible (longueur d'onde de diffusion entre 400 et 700nm) [27] avec une intensité suffisamment élevée pour être visualisée en microscopie optique. Une fois la particule choisie et synthétisée, il faut par la suite la fonctionnaliser afin qu'elle puisse cibler la protéine voulue.

2.2.2 Développement de sondes immunoplasmoniques

Le développement de sondes IP passe par 2 grandes étapes. Il faut d'abord synthétiser les nanoparticules en choisissant leur taille, leur forme ou leur composition selon les besoins de notre application. Ensuite, il faut effectuer la fonctionnalisation de celles-ci afin que les nanoparticules puissent cibler la protéine d'intérêt.

2.2.2.1 Synthèse de nanoparticules

Les nanoparticules métalliques peuvent être synthétisées par différentes méthodes. Parmi les méthodes proposées dans la littérature, les approches peuvent être chimiques, thermiques, électrochimiques et sonochimiques [28]. Ces différentes méthodes ont toutes leurs avantages et leurs inconvénients respectifs. Dans le cas de l'application IP, il est particulièrement important d'avoir un contrôle précis sur la taille et la composition des NPs afin de pouvoir assurer une diffusion suffisante et afin de prédire les couleurs des NPs présentes dans l'échantillon. La méthode Turkevich qui a d'abord été proposée en 1951 [29], puis améliorée par Frens en 1973 [30] est une approche de synthèse chimique qui permet justement de contrôler avec une certaine précision la taille des nanoparticules synthétisées. La méthode Turkevich-Frens est une méthode en deux étapes qui implique d'abord la synthèse de petites nanoparticules d'environ 20nm de diamètre appelées noyaux, suivie d'une étape de croissance qui permet – simplement en modifiant les concentrations

des réactifs – de faire croître les différentes nanoparticules jusqu’à la taille désirée. C’est donc une méthode idéale pour des applications de bio détection.

Le groupe de Pr. Meunier a développé puis publié en 2015 [31] une méthode de synthèse de nanoparticules d’alliages or-argent (Au-Ag). Cette méthode s’inspire de la méthode de *seeded-growth* de Turkevich-Frens combinée à une méthode de co-réduction. La méthode consiste d’abord en la génération de noyaux d’or par la méthode Turkevich, puis de deux ou trois phases de croissances où des composés d’or – HAuCl_4 – et d’argent – AgNO_3 – sont présents en fonction du ratio Au-Ag désiré dans la nanoparticule finale. Ceci permet le développement de nanoparticules de couleurs précises simplement en changeant les concentrations de réactifs lors de la synthèse. Il est donc possible d’augmenter grandement le potentiel de détection par l’utilisation de différentes couleurs de nanoparticules Au-Ag.

En effet, tel que mentionné à la section 2.1.1.2, il est parfois pertinent pour les pathologistes de cibler plusieurs protéines (ex. : HER2, ER, PR et/ou PD-L1). En utilisant différentes couleurs de nanoparticules ayant différentes cibles, il est donc possible de quantifier différentes protéines d’un coup. On peut donc parler d’une possibilité de multiplexage. L’article de Marcoux-Valiquette, et al. [22] qui propose le ciblage de 4 types de protéines simultanément utilise 4 nanoparticules différentes ayant des spectres de diffusion suffisamment espacés. Elles comprennent des nanoparticules d’Or de 100nm qui diffusent dans le jaune à 607nm, des nanoparticules d’Or cylindrique (*Nanorods*) de 25 x 71 nm qui diffusent dans le rouge à 669nm, des nanoparticules composées d’un alliage Or-Arget 50/50 de 74 nm qui diffusent dans le vert à 529 nm et des nanoparticules composées d’un alliage Or-Arget 10/90 de 43nm qui diffusent dans le bleu à 457nm. Les différentes couleurs de nanoparticules peuvent être visualisées à la Figure 2.5.

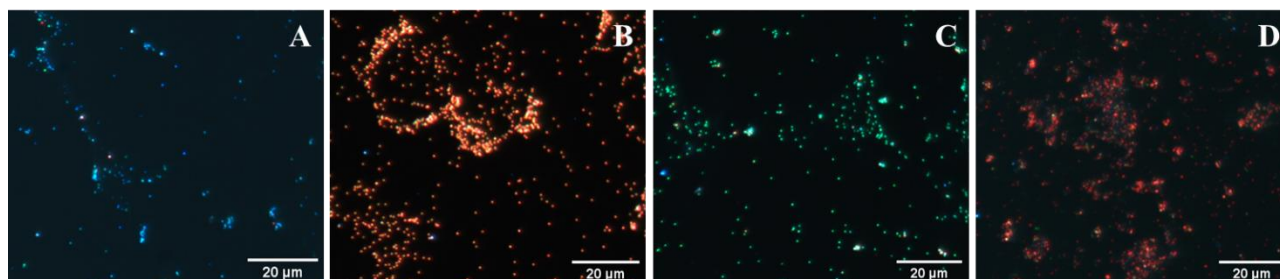


Figure 2.5 – Nanoparticules de différentes couleurs pour la biodétection sur lames de microscopie. A) NPs d’alliage Or-Arget 90/10 de 43 nm, B) NPs d’or de 100 nm, C) NPs

d'alliage Or-Argent 50/50 de 74 nm et D) *Nanorods* d'or de 25 x 71 nm. Note : Ces images sont des résultats non publiés fournis par Cécile Darviot.

Une fois synthétisées, les différentes nanoparticules peuvent être fonctionnalisées afin de devenir des sondes de détection qui sont capable de cibler des protéines et de s'attacher à celles-ci.

2.2.2.2 Fonctionnalisation de nanoparticules

Au fil des ans, différentes stratégies de fonctionnalisations ont fait leur apparition. La plupart des approches proposées utilisent soit le phénomène d'adsorption, le phénomène de liaisons covalentes ou des molécules adaptatrices (*adapter molecules*) (ex. : complexe chimique biotine-avidine) [32]. Le choix de la méthode est important, puisqu'il fera en sorte que les anticorps ajoutés seront plus ou moins stables sur la nanoparticule. Il aura aussi un impact sur la quantité d'anticorps maximale pouvant être ajoutés à la NP, sur l'orientation des anticorps et sur la stabilité des NPs dans le milieu.

La première stratégie, l'adsorption, profite du fait que les nanoparticules synthétisées possèdent une charge négative à leur surface. La stratégie consiste donc à ajouter des composés qui forment des liaisons ioniques avec la NP dont la surface est chargée négativement. Toutefois, ces méthodes sont moins stables que les méthodes covalentes [32].

La seconde stratégie qui utilise les liaisons covalentes est plus complexe que la stratégie d'adsorption ou de liaisons ioniques. Elle requiert généralement une activation qui permet le lien entre la nanoparticule et un composé chimique. Ceci permettra l'ajout d'un anticorps par après qui sera plus stable et mieux orienté. Il existe différentes approches de fonctionnalisation par liaisons covalentes telles que la chimie carbodiimide, la chimie maleimide et la click-chemistry [32].

Parmi ces stratégies de fonctionnalisations, celle qui sera appliquée aux nanoparticules utilisées dans le cadre de ce mémoire est une méthode de chimie carbodiimide appelée EDC-sulfoNHS [32]. Cette méthode implique d'abord l'ajout de SH-PEG (thiol-polyéthylène glycol) à la nanoparticule [33]. Une fois le PEG ajouté, les nanoparticules gagnent une stabilité dans des milieux salins s'apparentant aux milieux biologiques du corps humain. Le PEG utilisé possède un groupe COOH à son extrémité, qui permet les prochaines étapes chimiques. Après l'ajout des composés chimiques EDC (1-ethyl-3-(3-dimethylaminopropyl) carbodiimide) et NHS (N-hydroxysuccinimide) ou sulfo-NHS (N-hydroxysulfoxuccinimide), le EDC s'attache au PEG en réagissant avec le COOH, puis est remplacé sur le PEG par le NHS. Il est ensuite possible d'ajouter des anticorps qui réagissent avec le NHS afin de remplacer le NHS à l'extrémité du PEG. Le résultat de cette réaction

est une nanoparticule fonctionnalisée couverte de PEG et d'anticorps. Cette méthode permet d'avoir des anticorps qui sont orientés vers l'extérieur, mais ne garantit pas que le fragment de détection (Fab) de l'anticorps sera orienté vers l'extérieur [32].

La troisième stratégie, qui passe par l'utilisation de molécules adaptatrices (*adapter molecules*) peut permettre une orientation optimale des anticorps à la surface des nanoparticules afin que leur région Fab soit orientée vers l'extérieur. La méthode la plus courante utilisant ce type de chimie est le système biotine-avidine [32]. D'autres méthodes, tel que l'utilisation de protéines A/G pour lier des anticorps aux NPs peuvent aussi être employées. Dans le cadre de ce mémoire, les échantillons fournis par les étudiantes Cécile Darviot et Raphaële Fernet-Brochu contiennent des nanoparticules ayant été fonctionnalisées par la méthode EDC-sulfoNHS.

2.2.3 Imagerie et quantification

2.2.3.1 Microscopie optique

Les propriétés optiques des nanoparticules font en sorte que malgré leur petite taille, la lumière qu'elles diffusent est visible en microscopie optique. La microscopie en champ clair ne permet pas leur visualisation, alors que la microscopie en champ sombre le permet. Parmi les méthodes d'observation en champ sombre, nous discuterons de la microscopie en rétrodiffusion (*backscattering*) et de la microscopie par illumination latérale (*Side Illumination* (SI)) qui peuvent toutes les deux être utilisées pour visualiser des nanoparticules plasmoniques.

La microscopie en rétrodiffusion consiste à illuminer l'échantillon avec un angle, afin que seulement la diffusion de l'échantillon soit renvoyée vers l'objectif. La Figure 2.6 A) montre de manière schématique le chemin employé par la lumière en microscopie par rétrodiffusion sur un microscope inversé. Cette méthode d'observation permet de seulement voir la lumière diffusée par l'échantillon, donc lorsque l'on observe un échantillon contenant des nanoparticules qui diffusent la lumière, elles apparaissent comme de petits points brillants voir Figure 2.6 B).

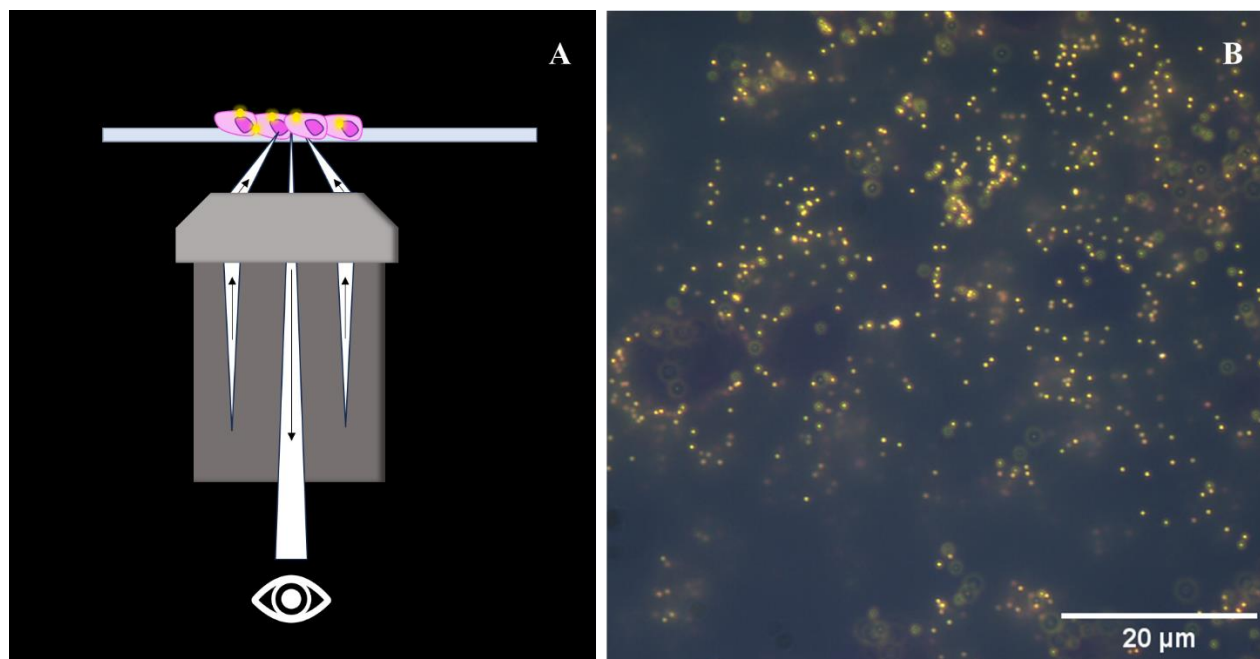


Figure 2.6 – A) Schéma de microscopie par rétrodiffusion. B) Image de nanoparticules sur cellules en rétrodiffusion.

Il est aussi possible de visualiser la diffusion des nanoparticules par illumination latérale. Le principe est relativement similaire, mais l'illumination est effectuée directement vers la lame sur laquelle se trouvent les cellules de l'échantillon et les nanoparticules. En plus d'être compatible avec un montage de microscopie classique, cette méthode permet de visualiser facilement la diffusion des nanoparticules, tout en permettant de visualiser les membranes cellulaires [4].

L'illumination latérale consiste en un dispositif simple qui peut être ajouté à la plupart des microscopes standards. Ce dispositif comprend des DELs rouges, vertes et bleues qui sont orientées vers le côté de la lame de microscopie [34]. La lumière émise par ces DELs permet de visualiser la diffusion des nanoparticules ainsi que certaines membranes cellulaires dans l'objectif principal.

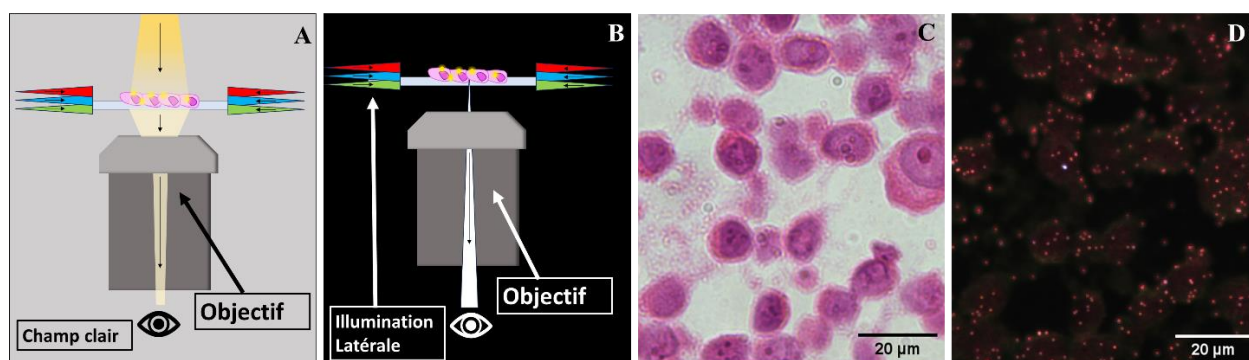


Figure 2.7 – A) Schéma du système d'imagerie en champ clair. B) Schéma du système d'imagerie en champ sombre. Échantillon cellulaire FFPE H&E marqué par IP C) acquis en champ clair et D) acquis en illumination latérale.

2.2.3.2 Quantification

Une fois les images de microscopie enregistrées à l'aide d'une caméra, il est par la suite possible de procéder à une quantification des nanoparticules afin de donner une mesure de l'expression d'une protéine. La méthode de quantification adoptée par le groupe de Pr. Meunier implique l'utilisation d'un outil nommé *Find Maxima* [35], [36] inspiré de la fonction disponible dans le logiciel d'analyse d'images Open Source (Accès Gratuit pour la recherche) *Fiji ImageJ* [37]. L'algorithme *Find Maxima* de ce logiciel d'analyse d'image permet l'extraction des maxima locaux dans des images au-dessus d'un seuil de tolérance au bruit.

L'algorithme applique une série d'opérations à l'image afin d'extraire les maximums locaux significatifs. Tout d'abord, l'image est soumise à un filtre de maximum local avec un noyau de 3x3 (*neighborhood maxima finder*). Toutefois, les maximums obtenus par cette méthode sont beaucoup trop nombreux pour être significatifs. Il faut donc appliquer un algorithme d'affinage des maximums qui requiert un seul paramètre : le seuil de tolérance au bruit, qui correspond à une valeur d'intensité de pixels. L'algorithme d'affinage de maximums se décrit comme suit [35]:

- 1- Analyser la région entourant chaque maximum afin de vérifier s'il y a des maximums plus élevés. S'il y en a, sauter ce maximum.
- 2- En partant du maximum le plus intense, si d'autres pixels se trouvent à une intensité qui est supérieure à la valeur du maximum (intensité du pixel) moins le seuil de tolérance au bruit, ajouter ces pixels à une liste de pixels ('scannés'). Cette étape de l'algorithme se propage dans l'image selon un algorithme *floodfill* jusqu'à ce que la zone soit complétée.

- 3- Si les pixels de la région sont égaux au maximum de départ, ce maximum est identifié comme 'égal'. Si les pixels sont inférieurs au maximum mais supérieur au maximum de départ moins le seuil de tolérance au bruit, ils sont marqués comme 'listés'.
- 4- Tous les pixels évalués sont ensuite identifiés comme '*processed*'.
- 5- Ensuite, parmi les régions contenant un maximum, le meilleur pixel pour être considéré le maximum parmi les pixels égaux est identifié en se basant sur la distance minimale entre ces pixels (on établit le maximum au centre d'un groupe de pixels égaux).

En résumé, le marquage d'échantillons par IP à plusieurs avantages. Premièrement, la méthode est compatible avec le marquage H&E utilisé en clinique qui permet l'évaluation de la morphologie des cellules de l'échantillon. Deuxièmement, puisque les nanoparticules apparaissent comme de petits points brillants, il est possible de les compter afin d'obtenir une mesure quantitative de l'expression de la protéine. Troisièmement, puisque la couleur des particules dépend de la synthèse, des NPs de différentes couleurs qui ciblent différents marqueurs peuvent être utilisés. Il est donc possible de faire du multiplexage (cibler différentes protéines simultanément). Quatrièmement, puisque la visualisation des nanoparticules peut être réalisée sur un microscope simple auquel est ajouté un adaptateur d'illumination latérale, la méthode est très facilement adaptable au milieu clinique.

Toutefois, cette technologie possède des désavantages liés à la quantification des nanoparticules qui est limitée par les méthodes d'imageries disponibles en clinique de pathologie et par l'algorithme de détection naïf (non « intelligent ») actuel. La méthode de quantification actuelle repose sur une valeur fixée (le seuil de tolérance au bruit) qui doit être ajustée manuellement entre les différents échantillons évalués, puisque l'algorithme a très peu de tolérance aux variations d'intensités dans les images. L'utilisation d'un algorithme d'apprentissage machine permettrait théoriquement de détecter les NPs dans les images sans avoir besoin d'effectuer des changements manuels entre chaque évaluation d'échantillon. De plus, un algorithme de ce type permettrait potentiellement de mieux détecter les NPs qu'un algorithme naïf. C'est pourquoi ce mémoire propose l'utilisation de l'intelligence artificielle (IA) afin d'effectuer une meilleure détection des nanoparticules.

CHAPITRE 3 Intelligence artificielle en santé

3.1 Concepts fondamentaux de l'intelligence artificielle

3.1.1 Introduction à l'apprentissage profond

Les définitions des termes liés à l'intelligence artificielle mis en **gras** peuvent être trouvées à l'Annexe A.

Le terme **intelligence artificielle** est un terme général pour décrire toutes les tâches dites intelligentes qui peuvent être réalisées par un ordinateur. Lorsqu'un algorithme possède la capacité d'apprendre, on parle alors d'**apprentissage machine**. Ces algorithmes peuvent être entraînés de manière **supervisée ou non**. Dans le cadre de ce mémoire, nous nous pencherons sur l'apprentissage supervisé. Ce type d'apprentissage peut servir à réaliser des tâches de **classification ou de régression** suite à un entraînement qui implique de multiples **itérations**. À chaque itération, les résultats en sortie de l'algorithme sont comparés à une **vérité de terrain** (*ground truth*) par une **fonction d'erreur** (*Loss function*) et les paramètres internes de l'algorithme sont mis à jour selon un **taux d'apprentissage** préétabli afin de minimiser la fonction d'erreur. Les paramètres internes qui permettent d'effectuer une décision à partir de données d'entrées peuvent être appelés **poids** (*weights*). La structure et le mode d'apprentissage de l'algorithme sont décrits par des **hyperparamètres** qui sont ajustés manuellement par l'expérimentateur responsable de l'entraînement du modèle. Ces algorithmes ont pour objectif d'effectuer des prédictions à partir d'informations.

Par exemple, un algorithme qui a pour but de prédire si un élève performera à l'école ou non en se fiant aux performances et aux habitudes de vies d'anciens élèves pourrait se baser sur les heures d'études et le temps de sommeil afin d'effectuer sa prédiction. Il serait alors possible d'effectuer soit une tâche de régression afin de déterminer les habitudes de sommeil et d'étude optimales ou une tâche de classification entre les élèves performants ou non (Figure 3.1). En prenant l'exemple de classification, à l'aide du processus d'entraînement (Figure 3.2), l'algorithme ajustera des poids qui correspondent aux paramètres d'une fonction de décision qui permettra de séparer les élèves en fonction des informations associées à chacun d'entre eux. Par exemple, pour une fonction de décision qui serait une régression linéaire on aurait : $y = w_0 + x_1w_1 + x_2w_2 + \dots + x_nw_n$, où w correspond aux poids du modèle. Une fois que cette fonction de décision est jugée acceptable,

l'entraînement est terminé et il est maintenant possible de prédire la performance d'un nouvel élève en fonction de ses habitudes de vies.

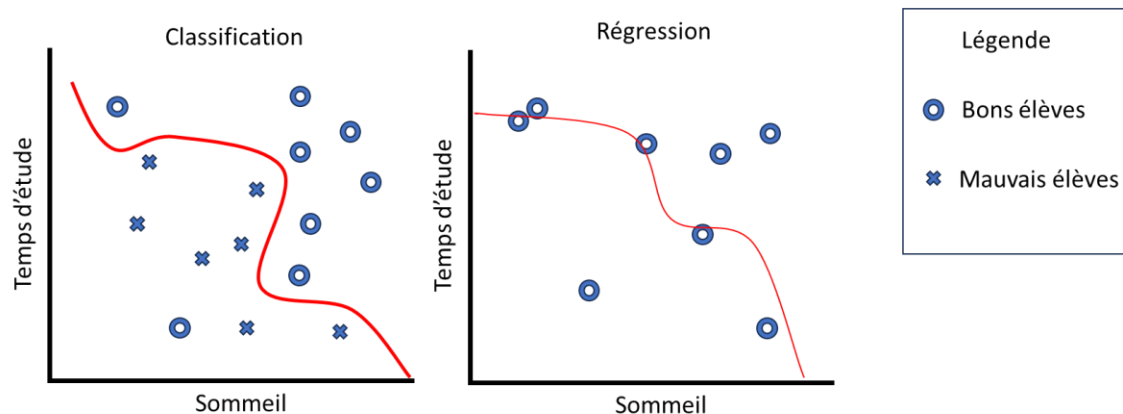


Figure 3.1 – Exemple de classification (gauche) et de régression (droite)

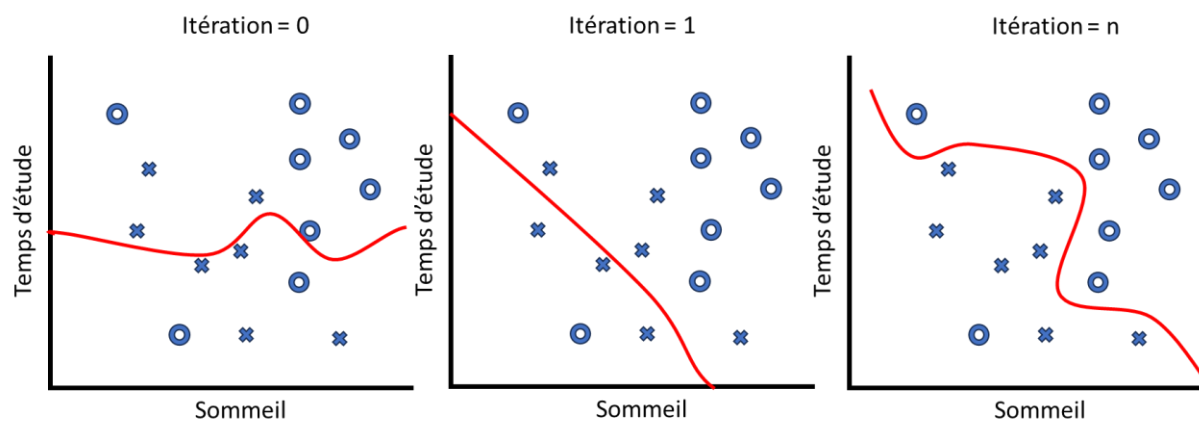


Figure 3.2 – Exemple d'évolution de la fonction de décision pour un modèle de classification à 2 variables.

Afin d'effectuer un entraînement comme celui de l'exemple précédent, la première étape sera d'initialiser les paramètres internes du modèle ou poids à des valeurs aléatoires. Par la suite, nous pouvons évaluer résultat du modèle en le comparant aux résultats attendus (vérité de terrain) par une fonction d'erreur (*Loss*). La fonction d'erreur sera ensuite dérivée par rapport à chaque poids afin d'obtenir la direction requise pour minimiser la fonction. Ce gradient sera ensuite multiplié

par le taux d'apprentissage et soustrait au poids correspondant afin d'effectuer une minimisation de la fonction d'erreur.

Ex. : $w_{f1} = w_{i1} - \frac{dLoss}{dw_{i1}} * Taux\ d'apprentissage$ où w_{f1} est le poids après l'itération et w_{i1} est le poids avant l'itération. La fonction modélisant la prédiction pourra donc évoluer grâce à cette modification des poids.

Ces étapes seront répétées pour un nombre d'itérations déterminé par l'expérimentateur, jusqu'à l'atteinte d'une performance raisonnable du modèle ou à l'atteinte d'un plateau d'apprentissage. En général, lorsque l'entraînement atteint un plateau, c'est souvent dû à un minimum local de la fonction d'erreur (*Loss*). C'est pourquoi la méthode d'optimisation qui est dictée par le choix d'optimiseur est très importante. L'ajout de *momentum* décrit à la section 3.2.1 peut par exemple permettre au modèle de sortir d'un minimum local pour aller vers un minimum plus significatif, tout en accélérant la convergence du modèle [38].

Lors de l'entraînement du modèle, il est possible de surentraîner un modèle au point qu'il ne soit plus capable d'évaluer des données autres que ses données d'entraînement. On parle alors de **surajustement** ou d'*overfitting*. Afin de limiter l'*overfitting* et d'augmenter la généralisation du modèle, la performance de l'algorithme sur des données de validation est évaluée à chaque itération afin de savoir quand le modèle a atteint le maximum de sa capacité d'entraînement sans *overfitting*. Après l'entraînement, il est pertinent d'évaluer le modèle avec de nouvelles données appelées données de test afin de savoir à quel point notre modèle performe sur des données qu'il n'a jamais vues.

Pour des bases de données entre 100 et 1 000 000 d'instances, les données d'entraînement, de validation et de test devraient respectivement représenter 60%, 20% et 20% des données [39].

Depuis quelques années, l'apprentissage machine a évolué vers **l'apprentissage profond**, décrit en détail par Y. Benjio et al. dans l'article *Deep Learning* [40]. L'apprentissage profond permet d'augmenter la capacité d'un modèle à comprendre les représentations de données avec plusieurs niveaux d'abstraction [40]. Cette approche a considérablement amélioré la performance des algorithmes de détection d'objets, de reconnaissance vocale et de reconnaissance visuelle. Elle permet de trouver de meilleurs paramètres internes ou poids pour le modèle. En apprentissage profond, les poids du modèle sont d'abord initialisés aléatoirement, puis interagissent avec les

données en entrée du modèle. Ensuite, le résultat de l'interaction entre la donnée d'entrée et le poids rejoint un **nœud ou neurone** qui se trouve dans une couche de neurones. Les résultats de ces interactions avec différents poids initialisés aléatoirement sont sommés, puis une **fonction d'activation** est appliquée. Celle-ci permet de garder seulement les valeurs de poids pertinentes pour l'évolution du modèle, tout en permettant d'introduire des non-linéarités qui permettent au modèle de devenir plus performant [40]. Les données passent à travers le modèle dans un **forward pass** afin que le modèle puisse s'autoévaluer en se comparant à la vérité de terrain. La fonction d'erreur est alors évaluée et les gradients des différents poids sont calculés dans une étape de **backpropagation** (voir Figure 3.3). C'est grâce à cette étape de *backpropagation*, que les poids du modèle pourront être mis à jour en suivant le gradient de la fonction d'erreur.

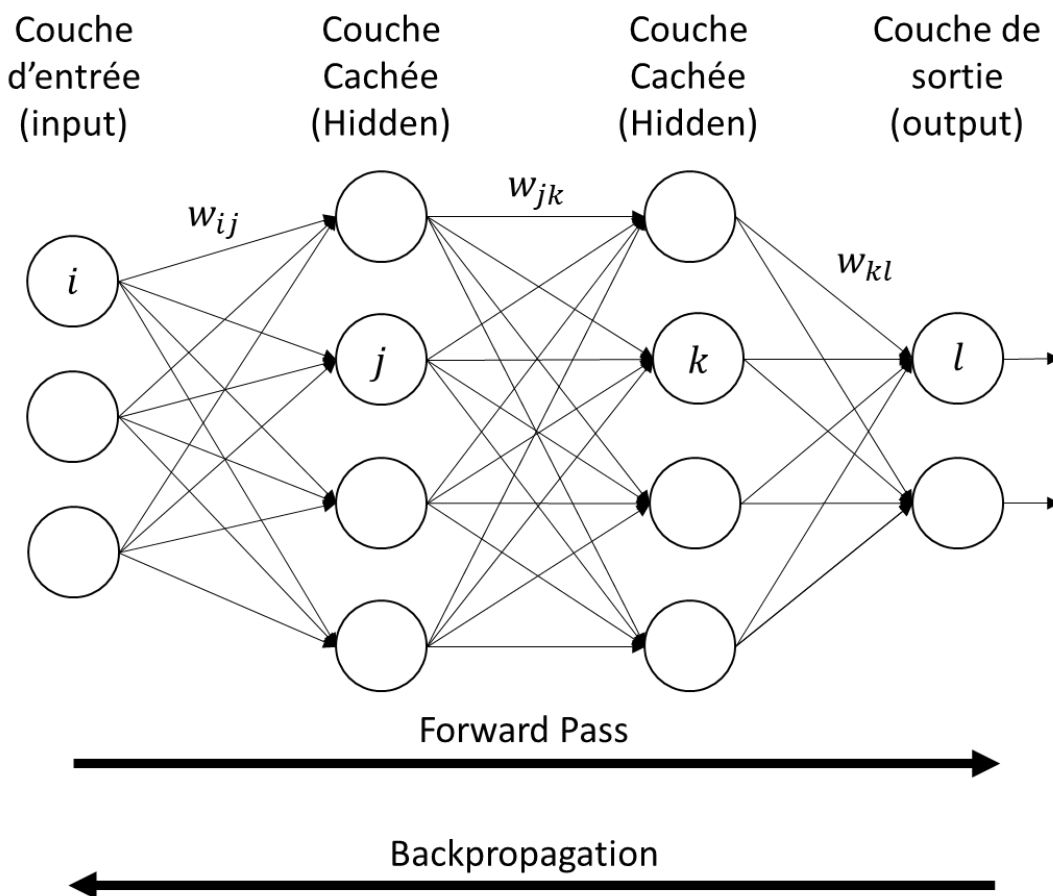


Figure 3.3 – Schéma de l'apprentissage profond inspiré de [40].

3.1.2 Réseaux Neuronaux Convolutifs

Une des grandes difficultés de l'apprentissage machine est de fournir à l'ordinateur des données dans un format compréhensible. Par exemple, lorsque l'on parle d'images, certaines relations spatiales entre les pixels ou entre les régions de l'image sont particulièrement importantes pour effectuer des décisions sur ces données. Il devient relativement difficile d'obtenir des valeurs numériques pour ce type de relations entre les informations. Initialement, les chercheurs ont tenté de déterminer le meilleur moyen d'exprimer les données en utilisant des approches logiques, telles que l'ajout de filtres qui permettent à l'algorithme de comprendre l'orientation des objets dans des images ou l'extraction de propriétés liées à des propriétés statistiques entre les pixels. Ceci a donné lieu au domaine de l'ingénierie de propriétés (*feature engineering*), qui impliquait que les informations extraites des images étaient conçues manuellement [41].

Toutefois, l'information permettant de comprendre le sens de l'image ne se retrouve pas souvent dans des pixels individuels. Il faut souvent comprendre le contexte entier de l'image [40]. Par exemple, dans le contexte d'application de ce mémoire, nous pouvons penser à une nanoparticule qui apparaît différemment sur une cellule ou sur le substrat. Dans ce type de cas, l'algorithme a besoin de comprendre le contexte de l'image afin de savoir qu'une nanoparticule est correctement identifiée et qu'une organelle cellulaire n'est pas considérée à tort comme une nanoparticule. La différence entre des nanoparticules sur le substrat et sur des cellules est montrée à la Figure 3.4.

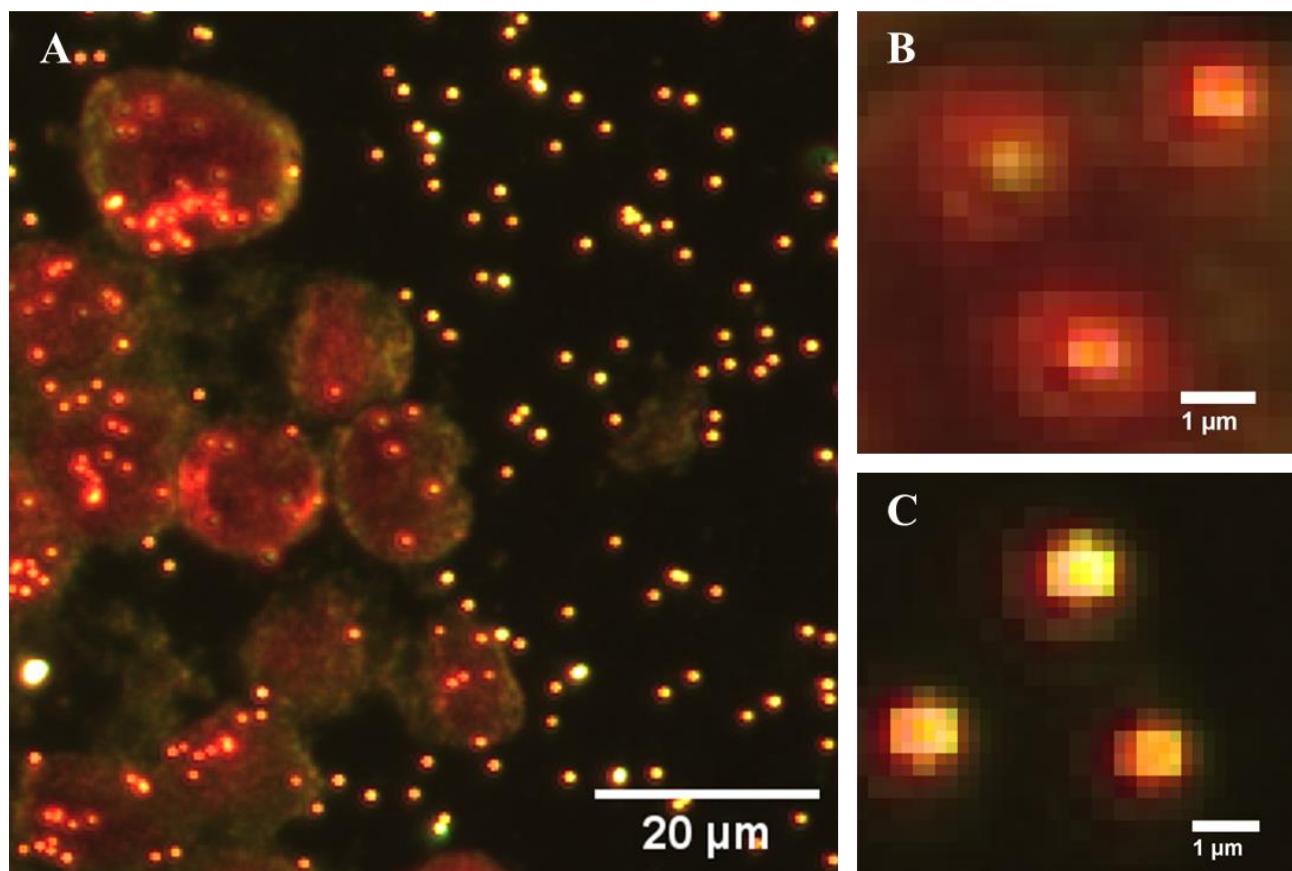


Figure 3.4 – A) Photo de nanoparticules sur cellules en microscopie en illumination latérale (40x à air). Agrandissement de nanoparticules situées B) sur une cellule et C) sur le substrat.

Afin d'extraire des informations pertinentes sur l'image, Lecun, Y., Benjio, Y., et al [41] ont publié en 1998 un modèle multicouche de type *artificial neural network* (ANN) appelé *LeNet-5* afin de classer des lettres écrites à la main. Ce type d'architecture ensuite appelé Réseaux Neuronaux Convolutifs - *Convolutional Neural Networks* (CNN) - peut apprendre la meilleure séquence de **convolution** à appliquer à une image afin d'extraire des **filtres** (*feature maps*) contenant de l'information pertinente sur l'image. Avec le temps, les CNNs ont été améliorés par de nouvelles séquences de convolutions telles qu'*AlexNet* [42], *ZFNet* [43], *VGGNet* [44], *GoogLeNet* [45], *ResNet* [46] et bien d'autres. Au fil du temps, les modèles développés sont devenus de plus en plus performants sur les grandes bases de données publiques telles que COCO [47] ou ImageNet [42] qui comprennent respectivement plus de 300 000 et plus de 1 300 000 images. Cette augmentation en performance vient souvent avec une augmentation de la profondeur des réseaux. Cette profondeur permet de mieux modéliser les images en informations pertinentes, mais une trop grande profondeur augmente le nombre de paramètres du modèle ainsi que le risque d'overfitting.

Les CNNs utilisent le concept de profondeur apporté par l'apprentissage profond afin de déterminer les valeurs numériques composant les noyaux convolutifs du CNN. En effet, une fois que les **noyaux** sont convolués à l'image originale, les résultats des convolutions (filtres) sont additionnés et soumis à des fonctions d'activations telles que *ReLU* (voir section 3.2.1) afin de garder seulement les informations pertinentes. Lors de l'apprentissage, les CNN sont entraînés en suivant l'approche de l'apprentissage profond, mais les poids qui sont modifiés à chaque itération servent plutôt à déterminer les valeurs numériques présentes dans les noyaux de convolution afin d'obtenir les filtres les plus descriptifs possibles de l'image. L'apprentissage de ces valeurs se fait ainsi en deux étapes. Une étape de ***feed forward*** ou ***forward pass*** lors de laquelle l'image est convoluée par toutes les couches de convolutions et est chaque fois soumise à une fonction d'activation jusqu'à la sortie du modèle. Puis une étape de ***backpropagation*** qui survient suite à la comparaison de la sortie avec le résultat attendu par la fonction d'erreur. Lors de cette étape, les poids du modèle sont modifiés en suivant le gradient de la fonction d'erreur afin de minimiser l'erreur du modèle et d'améliorer sa performance.

Lors de l'entraînement d'algorithmes de types CNN, il est fréquent de diviser les images en plusieurs sections afin de former des parcelles (*patches*), ceci permet à l'algorithme de concentrer son entraînement sur une région précise à la fois, tout en permettant de limiter la quantité de mémoire requise pour chaque étape d'entraînement. La taille de ces sections est appelée dans ce mémoire **taille de parcelle** (*patch size*). De plus, lors de l'entraînement, chaque itération est séparée en étapes, lors desquelles, une portion du nombre de parcelles totales est utilisée pour l'entraînement du modèle. Ce nombre de parcelles utilisées à chaque étape d'entraînement peut être appelé la **taille de lot** (*batch size*). Une fois que toutes les étapes sont complétées, ceci signifie que le modèle a été entraîné au moins une fois sur chaque parcelle et qu'une itération est complétée.

Après chaque étape de convolution, la résolution en largeur et en hauteur des filtres diminue. De cette façon, les efforts de calculs sont limités et les informations globales peuvent être extraites. Toutefois, lorsque l'on parle de détection de nanoparticules, l'objectif est de détecter de très petits objets qui peuvent mesurer quelques pixels seulement. Il est donc important de choisir une architecture de noyaux et de filtres qui permette d'effectuer des prédictions pour chaque pixel de l'image originale.

3.2 Segmentation Sémantique

Le fait de classer une image pixel par pixel s'appelle de la segmentation sémantique. Au fil des ans, différentes approches ont été proposées, telles que les réseaux entièrement convolués (*Fully Convolutional Networks* (FCN)) [48] qui effectuent des déconvolutions des différentes couches de convolutions après l'application des convolutions afin de retrouver une résolution similaire à l'image initiale. Les déconvolutions des couches de différentes profondeurs sont ensuite fusionnées avant d'appliquer la dernière fonction d'activation à cette couche et d'obtenir une réponse en sortie du modèle. Ceci permet d'avoir une prédiction pour chaque pixel de l'image. Un sous-type de FCN, le *UNet* [49] a permis de nettement améliorer les performances de segmentation sémantique. Cette architecture est rapidement devenue omniprésente dans le monde de la segmentation sémantique appliquée au domaine médical.

Le UNet est composé de deux parties principales :

La phase d'encodage (*contracting path*) [49] qui applique successivement des convolutions de noyaux 3x3, suivies d'une fonction d'activation de type ReLU et d'une étape de *max pooling* de 2x2 qui consiste à sous-échantillonner l'image en prenant seulement la valeur la plus importante dans une zone de 2x2 pixels. Le nombre d'étapes convolutives dépend de la séquence de convolution choisie, c'est-à-dire de la **structure d'encodage** (*backbone*) du modèle.

La phase de décodage appelée (*expensive path*) [49] implique plusieurs niveaux de suréchantillonnage (*Up-Sampling*). Les filtres sont d'abord suréchantillonnés, puis une opération appelée up-convolution ou déconvolution permet ensuite de diminuer le nombre de filtres de moitié. Ensuite, le résultat est concaténé avec les filtres de taille correspondante dans la phase d'encodage. Finalement, une convolution 3x3 suivi d'une fonction d'activation ReLU termine chaque bloc de décodage.

Une fois que la carte propriété est définie et reconstruite de manière à avoir des dimensions en hauteur et largeur similaires à l'image en entrée, il ne suffit plus qu'à appliquer une convolution 1x1 à l'image suivie d'une fonction d'activation permettant soit d'associer chaque pixel à une classe grâce à l'activation *Softmax* ou d'associer une valeur continue entre 0 et 1 à chaque pixel pour effectuer une régression à l'aide de la fonction d'activation *Sigmoïde*.

3.2.1 Hyperparamètres en segmentation sémantique

Il est possible de modifier différents paramètres de l'architecture ou de la méthode d'apprentissage des algorithmes de segmentation sémantique. Les aspects principaux du modèle sont : la séquence de convolution appliquée, la taille des images ou parcelles utilisées ainsi que la fonction d'activation utilisée après chaque séquence de convolution. Pour ce qui est de l'entraînement, les paramètres principaux sont la fonction d'erreur, le nombre d'images utilisées par étape d'entraînement (taille de lot) , l'optimiseur et le taux d'apprentissage.

La séquence de convolution est déterminée par la structure d'encodage du modèle. Il existe différentes structures d'encodages plus ou moins profondes qui sont devenues des standards dans les applications de détection d'objets et de segmentation sémantique telles que *VGG16/VGG19* [44] et *ResNet50/ResNet101* [46] (Tableau 3.1). Ces structures déterminent la séquence de convolution appliquée à l'image et la profondeur du modèle. Le nombre de paramètres de ces différentes architectures est décrit au Tableau 3.1. Ce tableau permet de constater que les modèles *VGG* ont plus de paramètres internes, mais que les modèles *ResNet* ont plus de couches convolutives. Ceci fait en sorte que l'entraînement et l'espace numérique occupé par des modèles *VGG* seront plus importants. Les modèles de type *ResNet* ont plus de profondeur, ce qui peut permettre une meilleure compréhension des données complexes, mais augmente le risque d'**overfitting** [50].

Tableau 3.1 – Architectures convolutives

Structure d'encodage	Nombre de couches convolutives	Nombre de Paramètres	Référence
VGG16	16	~138 millions	[44]
VGG19	19	~144 millions	[44]
ResNet50	50	~26 millions	[46]
ResNet101	101	~44 millions	[46]

Les fonctions d'activations sont utilisées à la sortie de chaque neurone ou séquence de convolution afin d'introduire des non-linéarités dans les prédictions et d'éviter une réduction trop importante

des valeurs de gradients à l'étape de backpropagation. Ces fonctions peuvent prendre différentes formes telles que *ReLU*, *LeakyReLU* [51] ou *GeLU* [52] montrées à la Figure 3.5. Ces activations peuvent être utilisées à tous les neurones du modèle. *ReLU* permet de seulement garder les valeurs de poids pertinentes (supérieures à 0), tandis que *LeakyReLU* et *GeLU* accordent une petite importance aux valeurs qui ne semblent pas pertinentes au cas où elles auraient tout de même une pertinence quelconque pour l'apprentissage.

Il existe aussi d'autres activations telles que *Sigmoïde* et *Softmax* qui sont utiles en sorties de modèle afin de pouvoir faire une régression ou une classification binaire dans le cas de la sigmoïde et afin d'effectuer une classification multi-classe à l'aide de la *Softmax*.

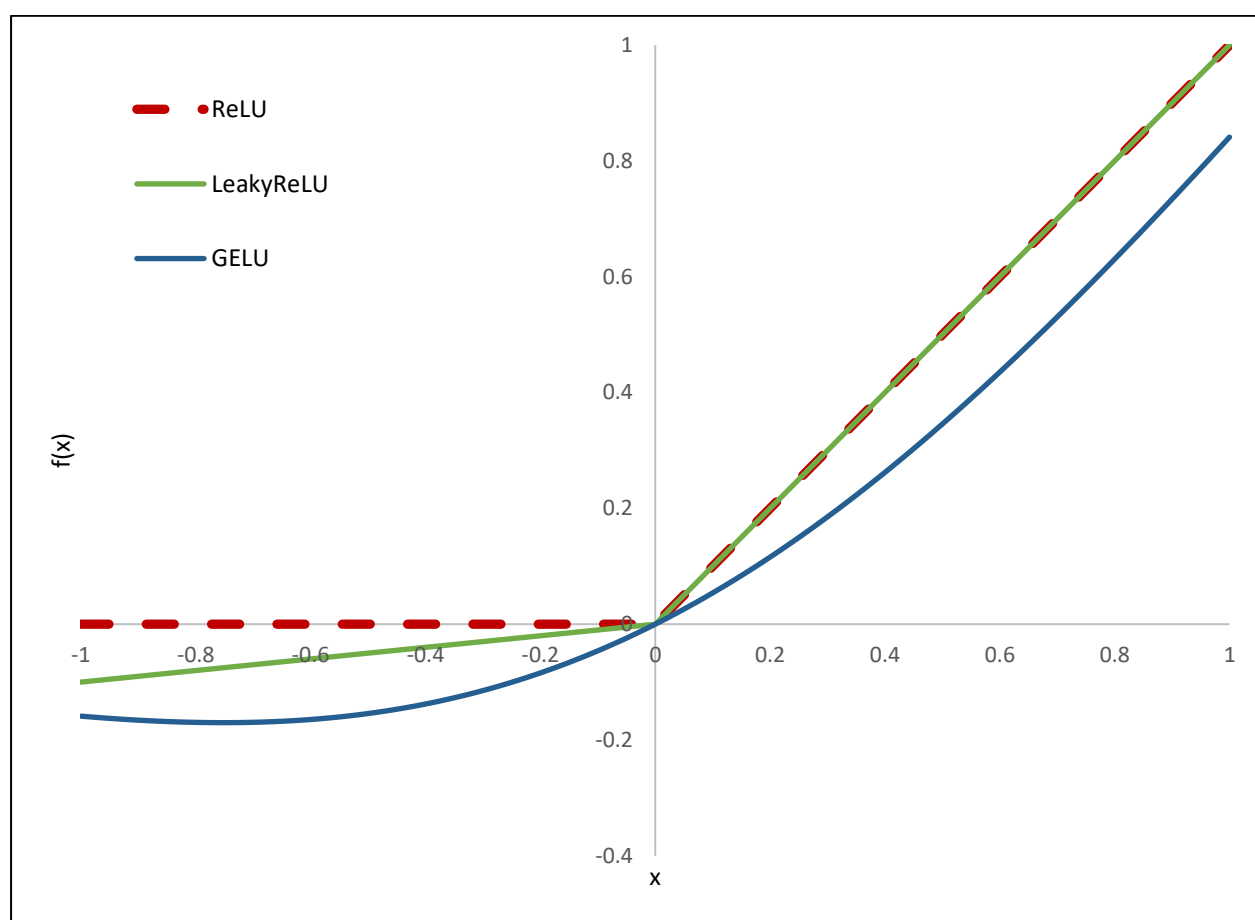


Figure 3.5 – Fonctions d'activations principales utilisées en segmentation sémantique

Les fonctions d'erreur ont aussi un grand impact sur la performance du modèle. Pour effectuer une classification des pixels, les fonctions de *categorical cross entropy* ou *binary cross entropy*

(dépendamment du nombre de classes) sont les approches classiques. Toutefois, de nouvelles fonctions qui utilisent les coefficients *Dice* [53] et *Intersection Over Union* (IOU) [54] ont été implémentées comme fonctions d'erreurs dans les applications de segmentation sémantique au cours des dernières années. Les fonctions d'erreur *Cross Entropy*, IOU et *Dice* peuvent être implémentées de la façon suivante :

***Cross Entropy* (CE) Loss:**

$$CE = - \sum_i^C t_i \log(s_i)$$

où t_i est la valeur de vérité associée à la classe, s_i est la valeur prédite et C est le nombre de classes

IOU:

$$Loss_{IOU_seg} = 1 - IOU = 1 - \frac{Aire\ Intersection}{Aire\ Union}$$

***Dice* :**

$$Loss_{Dice} = 1 - Dice_{coef} = 1 - \frac{2 * Vrai\ Positifs + \beta}{2 * Vrai\ Positifs + Faux\ Positifs + Faux\ Négatifs}$$

Où β est une constante de stabilité (exemple : $1 * 10^{-7}$).

Un autre paramètre important est la fonction d'optimisation ou optimiseur qui détermine comment la descente de gradient sera effectuée. L'optimiseur classique est la descente de gradients stochastique (SGD). Il est possible d'ajouter un terme de *momentum* [38] à cet optimiseur afin de mieux converger. Toutefois, l'optimiseur le plus utilisé au cours des dernières années est l'optimiseur *Adam* [55] qui permet une convergence rapide de l'algorithme par l'adaptation du taux d'apprentissage en fonction de la rapidité à laquelle converge l'algorithme.

3.2.2 Applications récentes et stratégies et détections

Le *UNet* a été particulièrement adopté dans le domaine médical, puisque la segmentation sémantique est très utile pour détecter différents organes, tumeurs ou autres objets anatomiques avec précision dans des images parfois très complexes à analyser. L'innovation dans le domaine a donc explosé, ce qui a permis le développement de différentes approches pour améliorer l'architecture et une application des *UNets* à des tâches médicales variées.

Les processus d'attentions dans des architectures comme le *UNet* permettent d'augmenter l'importance de régions d'intérêts dans les images analysées. À ma connaissance, la première approche d'attention impliquant un *UNet* est l'implémentation d'*attention gates* (AG) par Oktay, et al. en 2018 [56]. Cette approche a permis d'augmenter la performance du *UNet* lors de la segmentation du pancréas, de la rate et des reins dans des images CT provenant de la base de données *CT-150*. Les portes d'attention sont ajoutées à la fusion des filtres plus profonds et des filtres plus détaillés dans la phase de décodage. À la place de simplement suréchantillonner les informations moins détaillées, elles sont utilisées afin générer un coefficient d'importance pour chaque pixel qui est ensuite multiplié (pixel par pixel) aux filtres détaillés provenant de la phase d'encodage.

Une autre approche développée concernant les mécanismes d'attentions est le concept de *self-attention*, qui consiste à établir les régions les plus importantes dans les filtres les plus profonds du *UNet*, permet au *UNet* d'avoir une hiérarchisation des régions de l'image. Ceci permet au *UNet* de déterminer quelles régions globales de l'image sont intéressantes. L'article par Guo, et al. [57] ayant publié cette approche proposait une méthode de détection cellulaire innovante. À partir d'annotations du centre de cellules, ils ont entraîné leur *Self-Attention UNet* à détecter le centre de cellules lorsque leur nombre (confluence) était très élevé. Ceci leur a permis de détecter des cellules de différentes bases de données acquises par une variété de modalités d'imagerie. Cette approche qui permet de discerner deux objets très rapprochés a été une inspiration pour la méthodologie de détection qui sera présentée dans la section méthodologie de ce mémoire.

En 2021, le *TransUNet* [58] a été proposé. Cette approche combine l'utilisation d'une architecture *UNet* pour extraire des informations pertinentes de l'image lors de la phase d'encodage, mais soumet les derniers filtres à un *Transformer*. Ceci permet à l'algorithme d'avoir un mécanisme d'attention supplémentaire. Le *Transformer* est une approche d'encodage et de décodage de l'information pour l'apprentissage machine maintenant utilisé dans les modèles génératifs de langage tels que ChatGPT. L'approche a été adaptée pour des images dans l'article *An Image is worth 16 x 16 words* [59] et permet d'abord de séparer l'image en régions afin que l'algorithme apprenne à encoder la relation entre chaque région. Cette méthode s'affranchit de l'utilisation de convolutions, qui sont des opérations très localisées. Ceci lui permet d'avoir une meilleure compréhension de l'image en entier. Quoique l'utilisation d'architectures d'encodage et de décodage seulement par *Transformers* est maintenant utilisée (voir le Swin *UNet* [60]), cette

approche demande un plus grand nombre de données d'entraînement et est moins performante pour extraire de l'information localisée sur des images d'après Chen et al. [58] .

CHAPITRE 4 Méthodologie

Afin de réaliser l'objectif de quantification précise de nanoparticules sur des échantillons cellulaires, la stratégie choisie impliquait 5 composantes principales:

- 1) Échantillons marqués par NPs plasmoniques
- 2) Banque de données d'images de ces échantillons marqués par Immunoplasmonique
- 3) Méthode pour générer une vérité de terrain fiable
- 4) Architectures de modèles capables de réaliser la tâche de détection de NPs
- 5) Méthodes de représentations potentielles des résultats de la détection

Le développement de chacune de ces composantes a été réalisé ayant comme objectif final une utilisation de l'algorithme dans un contexte clinique. Ceci impliquait certaines **contraintes** :

- 1) Les échantillons doivent être marqués par H&E pour que la morphologie des cellules soit visible en champ clair par les pathologistes. Ils doivent aussi provenir d'échantillons mis en blocs de paraffine et tranchés à une épaisseur d'environ 4 microns.
- 2) L'objectif utilisé pour la microscopie doit avoir une résolution similaire à celle des scanners de lames automatisés utilisés en clinique. La détection doit donc être réalisée avec une résolution maximale correspondant à celle d'un objectif 40x à air.
- 3) La détection des NPs doit être rapide (moins d'une minute par image) pour ne pas ralentir la routine des pathologistes.
- 4) La détection doit être au minimum plus fiable que la détection par un algorithme naïf tel que l'algorithme *Find Maxima* du logiciel *ImageJ*.

4.1 Protocole de préparation des échantillons

Les échantillons utilisés pour la génération des images furent préparés par deux étudiantes du groupe de Pr. Meunier et de Dr. Dominique Trudel du CRCHUM, soit Cécile Darvot et Raphaële Fernet-Brochu. Un CMA (*Cell Micro Array*) avec différentes lignées cellulaires du cancer du sein exprimant à des niveaux différents la protéine HER2 leur a été fourni par le Pr. Mader de l'Université de Montréal. Les échantillons FFPE marqués par H&E, furent marqués par Immunoplasmonique (IP). Les sondes IP utilisées sont des nanoparticules d'or (*NanoComposix, Inc.*, de 100nm ($99 \pm 10nm$) – lot : JLC0071 – *Product Number* : AUCN100) recouvertes de PEG

avec une terminaison en acide carboxylique et d'un anticorps anti-HER2 (222482 d'Abcam anti-HER2).

Les principaux protocoles impliqués lors du marquage par IP sont présentés à l'ANNEXE B. Ces étapes de préparations ont été réalisées par les équipes de Pr. Trudel et Pr. Meunier; plus particulièrement par les étudiantes Cécile Darviot et Raphaële Fernet-Brochu.

De ces expériences, trois CMAs ont été obtenus. De plus, un TMA (Tissue Micro Array) a aussi été réalisé avec les mêmes étapes de préparation que les CMAs. Des trois CMAs, seulement deux ont maintenu une coloration H&E pour la durée du projet, rendant ainsi l'utilisation du troisième CMA moins propice à l'entraînement de l'algorithme.

4.2 Préparation des données d'entraînement

La stratégie de détection de nanoparticules proposée est l'utilisation d'une microscopie à plus haute résolution que les outils disponibles en clinique afin de générer un masque de vérité. Ce dernier permet ensuite d'entraîner un modèle de segmentation sémantique de type *UNet* à détecter des nanoparticules dans des images acquises selon des conditions respectant les contraintes d'application clinique de la technologie.

4.2.1 Acquisition des images

La première étape de la génération de masques qui serviront de vérité de terrain pour l'algorithme passe par l'acquisition d'images hautes résolutions des NPs. L'objectif utilisé est un objectif 60x à huile (Nikon Plan Fluor 60X Oil | 0,50-1,25 MRH02601) ayant une ouverture numérique de (0,50-1,25). L'huile utilisée entre l'objectif et la lame est une huile d'immersion à faible autofluorescence (Olympus IMMOIL-F30CC | *Immersion Oil Type F for Microscopy* | $ne = 1,518$; $ve = 41$; 23C | *Batch No. :200101*). Les images sont acquises par une caméra monochrome (Mono) QIClick (01-QIClick-F-M-12 Mono 12bit | SN : Q30018 | pixel size (h x v) : 6,45um x 6,45 um).

Les images à basse résolution ont été acquises à partir du même microscope que les images à haute résolution. Cependant, l'objectif utilisé est un objectif 40x à air (Nikon Plan Fluor 40X | 0,6 MRH08430) ayant une ouverture numérique de (0,60). Les images ont été acquises en couleur (RGB) à l'aide de la caméra PCO (pco.panda. 4.2 C SN :14400608, taille de pixel (h x v): 6,5um x 6,5 um). Les paramètres d'acquisition d'images avec NPs ont été fixés dans le logiciel d'accompagnement de la caméra selon le Tableau 4.1.

Tableau 4.1 – Paramètres d’acquisition des images basse résolution par la caméra PCO

Saturation	4%	Contraste	0%
Vibrance	-10%	Gamma	1,25
Col. Temp.	4472	Range	0 à 46617
Teinte	-65%	Temps d’acquisition	300 ms

La résolution lors de l’acquisition est limitée par la longueur d’onde et l’ouverture numérique de l’objectif. Elle suit la formule suivante : $Résolution = \frac{1.22 * \lambda}{NA}$. En prenant une longueur d’onde correspondant au centre des longueurs d’ondes du visible (400 à 700nm), il est possible d’estimer théoriquement la résolution associée à l’ouverture numérique de l’objectif. Ainsi, $Résolution_{microscopie40x} = 1,22 * \frac{550nm}{0,6} = 1,118\mu m$ et $Résolution_{microscopie60x} = 1,22 * \frac{550nm}{[0,75; 1,25]} = [0,894; 0,536]\mu m$.

Note : Les ouvertures numériques pour l’objectif à haute résolution étaient ajustées entre 0,75 et 1,25 lors de l’acquisition, c’est pourquoi cet intervalle est utilisé pour le calcul précédent.

Il est aussi possible d’estimer la résolution associée à l’acquisition de la caméra en divisant la taille des pixels par le grossissement tel que : $Résolution_{camRGB} = \frac{6,5\mu m}{40x} = 0,16525\mu m$ et $Résolution_{camMono} = \frac{6,45\mu m}{60x} = 0,1075\mu m$.

Ces estimations permettent d’établir la résolution limitante comme étant celle associée à l’objectif. L’objectif à huile est donc très pertinent afin d’obtenir une meilleure résolution lors de la détection des NPs.

Puisque la résolution de l’objectif à huile permet une meilleure identification des nanoparticules, elle fait aussi en sorte que la résolution en z est meilleure. Ce faisant, il n’est donc pas possible de visualiser toutes les nanoparticules d’un échantillon de largeur de 4 microns en un seul plan d’acquisition (voir Figure 4.1). Il faut donc effectuer un *scan* en z afin d’identifier les nanoparticules qui se trouvent dans différents plans. Afin d’assurer une couverture complète de l’échantillon (épaisseur d’environ 4 microns), 30 images sont acquises avec un intervalle en z de

0,5 microns entre chaque image. La hauteur couverte par ce scan est donc d'approximativement 15 microns. Il est possible qu'un glissement se produise lors de l'acquisition, c'est pourquoi la zone couverte est plus importante que la hauteur attendue de l'échantillon.

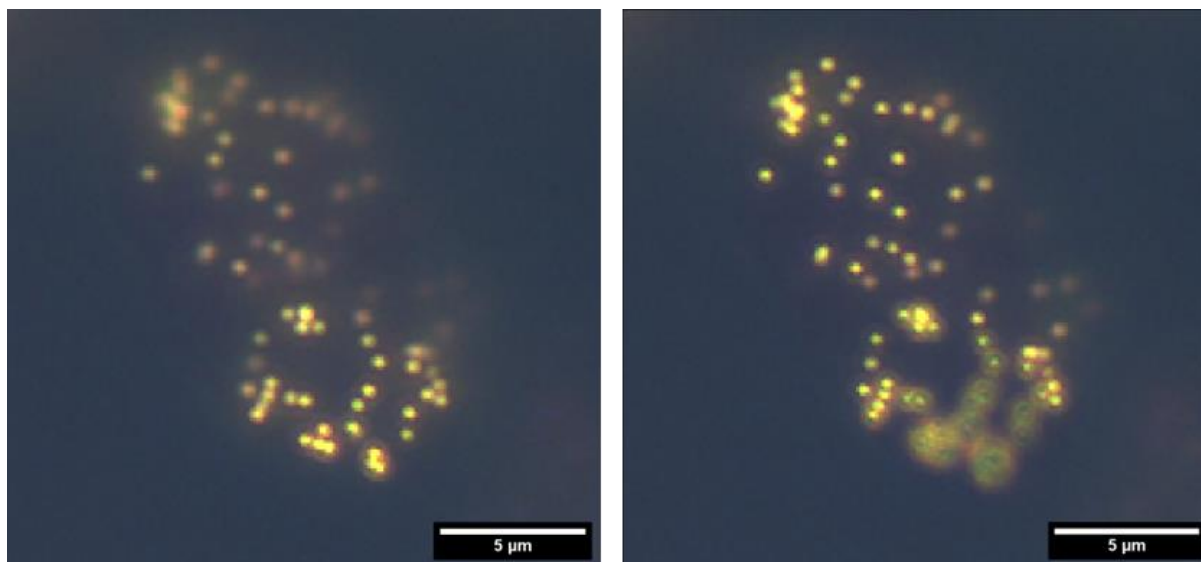


Figure 4.1 – NPs sur cellules en rétrodiffusion (microscopie 60x à huile) selon différentes couches d'acquisition d'un *z scan*. La distance en *z* séparant les plans focaux de ces deux images est inférieure à 1 micron.

4.2.2 Génération de masques de vérité 'automatiques'

L'acquisition des images à haute résolution implique donc une acquisition en *z* de l'échantillon. Ceci permet de détecter chaque nanoparticule et de créer une vérité de terrain à partir de leur position. Afin d'effectuer la détection des nanoparticules dans chaque plan, l'algorithme naïf *Find Maxima*[36] du logiciel d'imagerie *ImageJ* [37] (décrit à la section 2.2.3.2) est utilisé. Les différentes étapes de détection des NPs dans les différentes couches en *z* sont les suivantes:

- A) La fonction *Find Maxima* est appliquée à chaque image du *stack*. Cette fonction prend comme paramètre d'entrée une valeur de tolérance au bruit qui correspond à une intensité de pixel. Cette valeur est déterminée par essai-erreur pour chaque couche en *z* afin d'obtenir une détection raisonnable des nanoparticules dans les différentes images du *stack*. En général, la valeur de tolérance au bruit optimale permettant de détecter adéquatement les NPs varie entre 5 et 40 pour des valeurs de maximums encodées en 8 bits (valeurs de 0 à 255 d'intensité de pixels).

- B) Une fois les détections effectuées, elles sont soumises à une fonction qui élimine les détections doubles de nanoparticules. Cette fonction prend en entrée une distance radiale minimale ($\sqrt{x^2 + y^2}$) et une distance verticale (z) minimale. Pour chaque NP, si une autre NP se trouve à une distance radiale de moins de 10 pixels et une distance verticale (couches du scan) de moins de 4 couches, les intensités des pixels correspondants aux deux NPs sont comparées et seulement la nanoparticule ayant la plus forte intensité moyenne est conservée.
- C) Ensuite, puisque seulement les coordonnées dans le plan xy sont considérées pour le masque de vérité, les détections sont projetées dans ce plan afin d'obtenir une vérité de terrain en 2 dimensions.
- D) Une fonction permet ensuite de générer un masque où chaque nanoparticule apparaît comme une gaussienne dont les paramètres sont : $\sigma = 0,5$; $\mu = 0$ dans un noyau de 8×8 pixels. Cette fonction prend en entrée les paramètres de la gaussienne désirée et les coordonnées des nanoparticules, puis ajoute une gaussienne à une image vide pour chaque nanoparticule. L'ajout d'une gaussienne est un choix qui a été effectué dans l'optique de pouvoir réaliser des tâches de régression ou de classification. Ce faisant, puisque les modèles utilisés sont des modèles de classification, les masques sont seuillés, de manière à être composés de seulement deux valeurs de pixels (0 ou 1).
- E) Le masque est ensuite enregistré.

4.2.3 Alignement des images et des masques

La méthode utilisée afin de s'assurer que la disposition spatiale soit la même entre les images issues de l'acquisition au 40x à air et au 60x à huile est une méthode de déformation à 3 points. Pour ce faire, 3 coordonnées correspondantes sont sélectionnées dans l'image 40x et dans l'image à 60x. Ensuite, la fonction *cv2.getAffineTransform()* est utilisée pour obtenir les coefficients de la transformation affine entre les points, puis la fonction *cv2.warpAffine()* est utilisée pour aligner l'image basse résolution à l'image haute résolution. Ces deux fonctions sont disponibles dans la librairie *Python/C++ OpenCV* [61].

Afin de maintenir l'intégrité des masques de vérités, l'image ajustée est l'image couleur (basse résolution). Ceci signifie que lors de l'entraînement et lors de la détection, l'image utilisée sera une

version déformée de l'image originale. Pour pallier à cela, lorsqu'une image non déformée est évaluée, elle est suréchantillonnée de manière à ce que sa longueur et sa largeur équivalent à 1,5 fois leur valeur réelle.

4.2.4 Ajustements préentraînements et augmentation de données

Une fois les images et les masques préparés, une étape de séparation de la banque de données est effectuée. Les séparations pour de petites banques de données doivent accorder une proportion importante à l'entraînement afin que le modèle soit exposé au plus grand nombre de variations possible. Il faut toutefois s'assurer de maintenir une certaine quantité de données pour la validation et pour l'évaluation. Les séparations sont montrées au Tableau 4.2. Les images utilisées pour l'évaluation finale (images de test) proviennent de différentes sessions d'acquisitions afin de garantir une performance de l'algorithme dans des contextes relativement variables.

Tableau 4.2 – Séparation des données pour l'entraînement, la validation et les tests

Séparations	Images (1024 x 1024 px)	Patchs (256 x 256 px)	Patchs (128 x 128 px)	Proportion
Entraînement	27	346	1382	70%
Validation	-	86	346	17%
Test	4	64	256	13%
Total	31	496	1984	100%

Toutes les images sont normalisées selon les trois canaux de couleurs (R, G et B) avant l'entraînement et avant d'effectuer des prédictions sur les images de tests. Une augmentation de données est effectuée à l'aide de couches préconçues de la librairie *Python Tensorflow|Keras* [62]. Cette augmentation de données a permis aux modèles d'être entraînés à visualiser les NPs après différentes transformations. Les transformations appliquées aux images sont :

- 1- Transformation de type réflexion horizontale et verticale par la couche : *tf.keras.layers.RandomFlip()*
- 2- Transformation de type Rotation par la fonction *tf.keras.layers.RandomRotation()*
- 3- Transformation de type Zoom par la fonction *tf.keras.layers.RandomZoom()*

- 4- Transformation de type variation de contraste par la fonction `tf.keras.layers.RandomContrast()`

Ces transformations sont appliquées aléatoirement selon un nombre d'itérations prédéfini. En augmentant les données d'entraînement 5 fois, on atteint 1730 parcelles ou *patches* de taille (256 x 256) ou 9920 parcelles de taille (128 x 128).

4.3 Entraînement des modèles

Avant de procéder à l'entraînement de modèles, il faut s'assurer de bien déterminer le type de modèles que l'on désire entraîner et les hyperparamètres que l'on souhaite faire varier.

Afin d'effectuer une détection des nanoparticules qui soit précise en termes de position de pixels. Il est important de choisir une architecture de réseau qui permette d'obtenir une prédiction ayant une résolution équivalente à l'image d'entrée. C'est pourquoi l'approche du *UNet*, une architecture de réseau très reconnue dans la littérature a été priorisée. Le *UNet* qui, tel qu'exprimé précédemment à la section 3.2, permet une extraction d'informations globales et locales des pixels de l'image, permet aussi d'obtenir une prédiction de résolution similaire à celle de l'image d'entrée.

Parmi les différentes architectures de *UNet*, trois furent choisies, soit le *UNet*, l'*Attention UNet* et le *TransUNet*.

La tâche classification de pixels permettant une détection de NPs passe par un entraînement supervisé de différents modèles. Lors de l'entraînement de l'algorithme, ce dernier va comparer sa prédiction avec le masque de vérité par une fonction d'erreur. L'objectif final est ainsi de déterminer quels pixels sont associés au centre des NPs afin d'obtenir une position et une détection pour chaque NP. Les différents hyperparamètres des modèles peuvent avoir des impacts importants sur les modèles finaux, c'est pourquoi différents hyperparamètres doivent être évalués. Les différents hyperparamètres généraux utilisés sont présentés à la section 4.3.1 et les différents hyperparamètres d'entraînements utilisés sont présentés à la section 4.3.2.

Les modèles proviennent de la librairie python *Keras UNet Collection* [63] qui reprend les architectures (*UNet*, *Attention UNet* et *TransUNet*) à partir de leur article d'origine. Les hyperparamètres de ces modèles peuvent être modifiés directement lors de l'initialisation des différents modèles. Parmi ces hyperparamètres on compte : La taille de l'image en entrée, le nombre de filtres (par étape d'encodage ou de décodage), le nombre de classes, la fonction

d'activation, la taille des batchs, le nombre de couches d'encodage et de décodage, le type de *Pooling* (pas discuté dans ce mémoire), l'interpolation utilisée pour le décodage (pas discutée dans ce mémoire). Pour le module *Transformer* du *TransUNet*, différents hyperparamètres peuvent être modifiés tels que : le nombre de blocs de *Transformer*, le nombre de têtes d'attentions, le nombre de nœuds par *Transformer*, le nombre de dimensions d'encodage (l'encodage du *Transformer* n'est pas convolutif) et une fonction d'activation pour la section *Transformer*.

Parmi ces hyperparamètres, les valeurs d'initialisations en excluant celles qui seront discutées aux sections 4.3.1 et 4.3.2 sont les suivantes : nombre de classes : 2, nombre de filtres : (64, 128, 512, 256, 1024, 2048), activation : ReLU, activation en sortie : *sigmoïde*, normalisation de *batch* : Vrai, méthode de décodage (unpooling) : le plus proche (*nearest*), poids initiaux: *ImageNet*.

L'utilisation de modèles préentraînés sur *ImageNet* permet aux modèles de converger plus rapidement, ce qui est très bien pour notre application où la taille de la banque de données est restreinte.

4.3.1 Paramètres généraux des modèles

Afin de développer le meilleur modèle de détection de NPs, différentes architectures de modèles, différentes structures d'encodage (*backbones*), différentes tailles de parcelles (*patch size*) et différentes tailles de lots (*batch size*) ont été utilisées lors des entraînements.

Les architectures choisies, *UNet*, *Attention UNet* et *TransUNet* permettent d'évaluer l'importance d'ajouter des mécanismes d'attentions au *UNet*. Cette attention peut être utilisée lors de la reconstruction des prédictions (partie décodage) en utilisant un *Attention UNet* ou dans les couches les plus profondes du *UNet* afin d'obtenir une conception globale de l'image en utilisant le *TransUNet*.

Les séquences de convolutions choisies aussi appelées structures (*backbones*) sont *VGG16*, *VGG19* et *ResNet50*. Les structures d'encodage *VGG* sont très reconnus pour leur performance sur des images médicales et l'ajout de *ResNet50* permet de comparer l'influence de la profondeur sur la performance des algorithmes. *ResNet50* possède beaucoup plus de couches convolutives que *VGG16* ou *VGG19*, mais possède moins de paramètres au total. L'augmentation de la profondeur pourrait soit augmenter la performance de l'algorithme, soit augmenter *l'overfitting* de ce dernier dépendamment de la complexité de la tâche.

L'impact des fonctions d'activations *ReLU* et *leakyReLU* n'ont pas été évalués, puisque des tests préliminaires démontraient une influence négligeable de ce changement de paramètres. La fonction d'activation *ReLU* a donc été utilisée pour tous les modèles entraînés.

Les autres paramètres généraux, ont aussi pu être comparés. Les tailles de parcelles (*patch*) utilisées furent de 128 x 128 pixels et de 256 x 256 pixels.

Tableau 4.3 – Paramètres généraux des modèles

Architecture	Structure (<i>Backbone</i>)	Fonction d'activation	Taille de parcelles (<i>Patch Size</i>)
<i>UNet</i>	<i>VGG16</i>	<i>ReLU</i>	(128 x 128) pixels
<i>Attention UNet</i>	<i>VGG19</i>		(256 x 256) pixels
<i>TransUNet</i>	<i>ResNet50</i>		

4.3.2 Paramètres d'entraînement

Il était ensuite important d'établir les paramètres d'entraînements optimaux pour le modèle. Pour la fonction d'erreur, les fonctions *Dice* et *iou_seg* furent utilisées. Il est possible de s'attendre à une performance raisonnable de ces fonctions, puisque leur valeur est relative à l'aire de pixels correctement classés, ce qui devrait améliorer la performance des modèles dans la détection de données déséquilibrées (entre le nombre de pixels associés au *background* et le nombre de pixels associés aux NPs). Les optimiseurs évalués furent *Adam* et *Stochastic Gradient Descent (SGD)* avec un *momentum* de 0,1. Normalement, *Adam* devrait converger plus rapidement que *SGD*, mais l'ajout de *momentum* à *SGD* devrait lui permettre d'obtenir des performances comparables à *Adam*. Le taux d'apprentissage (*learning rate*) était initialisé à 0,001, puis diminuait d'un facteur 10 après 15 itérations sans amélioration, jusqu'à un minimum de 0,00001. Dans tous les cas, si le modèle parcourait 20 itérations sans améliorations de la fonction d'erreur de validation, l'entraînement était arrêté.

L'augmentation de données était de $N = 5$ pour les modèles comparés. Les tailles de lots (*batch size*) utilisées furent de 3 et 5 parcelles par étape d'entraînement. Une plus grande taille de lot (*batch*) permet un entraînement plus rapide, mais peut limiter la capacité d'apprentissage du modèle.

Tableau 4.4 – Paramètres d'entraînement des modèles

Fonction d'erreur (<i>Loss</i>)	Optimiseur (<i>Optimizer</i>)	Augmentation de données	Taille de lot (<i>Batch Size</i>)
Dice	Adam	N = 5	3
Iou_seg	SGD		5

Afin d'obtenir une bonne comparaison de l'impact des hyperparamètres sur les modèles résultants, 9 combinaisons d'hyperparamètres sont créées : A, B, C, D, E, F, G, H et I (voir Tableau 4.5).

Tableau 4.5 – Combinaisons d'hyperparamètres pour l'entraînement des modèles.

Combinaisons d'hyperparamètres	<i>Backbone</i>	<i>Patch Size</i>	<i>Batch Size</i>	<i>Loss</i>	<i>Optimizer</i>
A	VGG16	256	3	Dice	Adam
B	VGG19	256	3	Dice	Adam
C	VGG19	256	5	Dice	Adam
D	VGG16	256	5	Dice	Adam
E	VGG19	256	5	iou_seg	Adam
F	VGG19	256	3	iou_seg	Adam
G	VGG19	256	3	iou_seg	SGD
H	VGG19	128	5	Dice	Adam
I	ResNet50	256	5	Dice	Adam

Ces combinaisons de paramètres sont utilisées pour chacune des 3 architectures principales (*UNet*, *Attention UNet* et *TransUNet*) (voir Tableau 4.6). Ce faisant, un total de 27 modèles sont obtenus.

Tableau 4.6 – Sommaire des modèles entraînés

Architectures	Combinaisons d’hyperparamètres								
<i>UNet</i>	A	B	C	D	E	F	G	H	I
<i>Attention UNet</i>	A	B	C	D	E	F	G	H	I
<i>TransUNet</i>	A	B	C	D	E	F	G	H	I

4.3.3 Résultats d’entraînements

La prédiction par les modèles permet l’obtention d’un masque dans lequel des points sont présents pour chaque NP détectée. À l’aide de la librairie Python *OpenCV* [61], nous pouvons retrouver la position associée au centre de chacun de ces points. Pour ce faire, la fonction *cv2.findContours()* est appliquée au masque afin d’obtenir les contours des objets présents dans le masque. Ensuite, pour chaque contour (correspondant à une NP), la fonction *cv2.moments()* permet d’obtenir les moments de chaque contour. À partir de ceux-ci, il est possible de retrouver le centre des NPs. Les coordonnées de chaque nanoparticule détectée peuvent ainsi être enregistrées.

4.4 Évaluation des modèles

Il faut ensuite évaluer la performance de chaque modèle afin de les comparer. Pour ce faire, il serait possible de se fier aux métriques utilisées dans les fonctions d’erreur du modèle quantifiant les ratios entre pixels correctement classés et mal classés (coefficient *Dice* ou *IOU*). Cependant, l’objectif réel est de correctement identifier les NPs et leur position et ce peu importe le nombre de pixels associés à chaque NP. C’est pourquoi l’évaluation finale des modèles est effectuée en comparant chaque nanoparticule détectée à la nanoparticule la plus rapprochée du masque de vérité. Les masques de vérité générés automatiquement par la méthode proposée à la section 4.2.2 sont suffisamment précis pour permettre un entraînement adéquat du modèle. Toutefois, leur conception dépend de l’algorithme naïf *Find Maxima* sur les images haute résolution. Les lacunes de cet

algorithm font en sorte que certaines NPs ne sont pas correctement identifiées dans les masques automatiques. C'est pourquoi une vérité de terrain manuelle a aussi été conçue. Cette dernière implique une annotation manuelle du *stack* en z entier de 4 images de 1024 x 1024 pixels. Les quatre images sélectionnées et annotées manuellement sont présentées à la Figure 4.2. Il est possible de remarquer que l'image B de la figure 4.2 possède des paramètres d'acquisition différents. En effet, lors de cette séquence d'acquisition, le système d'illumination latérale avait des DELs dysfonctionnelles. Les images de cette séance d'acquisition ont été utilisées dans les données d'entraînements, de validation et de test afin de pouvoir visualiser la capacité d'adaptation des modèles à une image présentant des différences de couleurs.

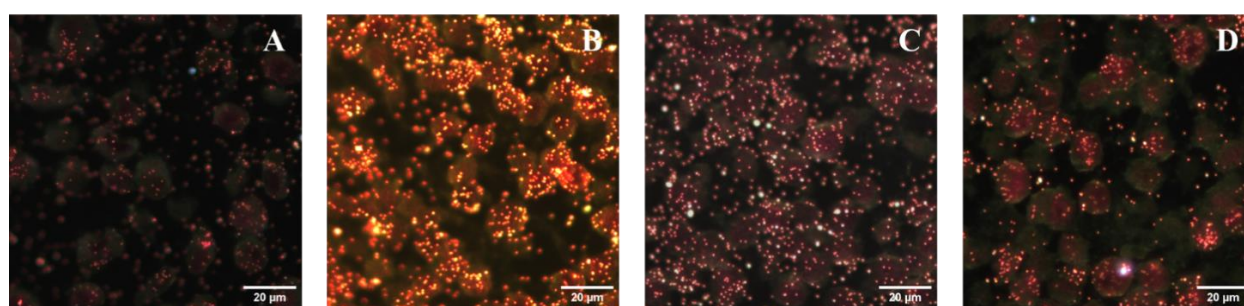


Figure 4.2 - Images couleurs de nanoparticules sur cellules en illumination latérale, toutes les images provenant d'une acquisition en 40x ont été alignées aux masques (issus d'une microscopie en 60x). Ce faisant, les images ont des dimensions comparables à une image 60x. A) **Image de test #1** - Cellules MCF-10A. B) **Image de test #2** - Cellules T47D sans illumination bleue. C) **Image de test #3** - Cellules MCF-7. D) **Image de test #4** - Cellules MDA-231.

Une autre image présentée à la Figure 4.3 est utilisée pour évaluer les modèles sur des concentrations très élevées de NPs. Toutefois, cette image provient d'un échantillon non marqué par H&E et ne représente donc pas adéquatement la tâche que le modèle tente de résoudre. Cette

image permet toutefois d'évaluer le modèle final sur une image à haute densité de NPs. Elle sera alors appelée 'Image de Test #5'.

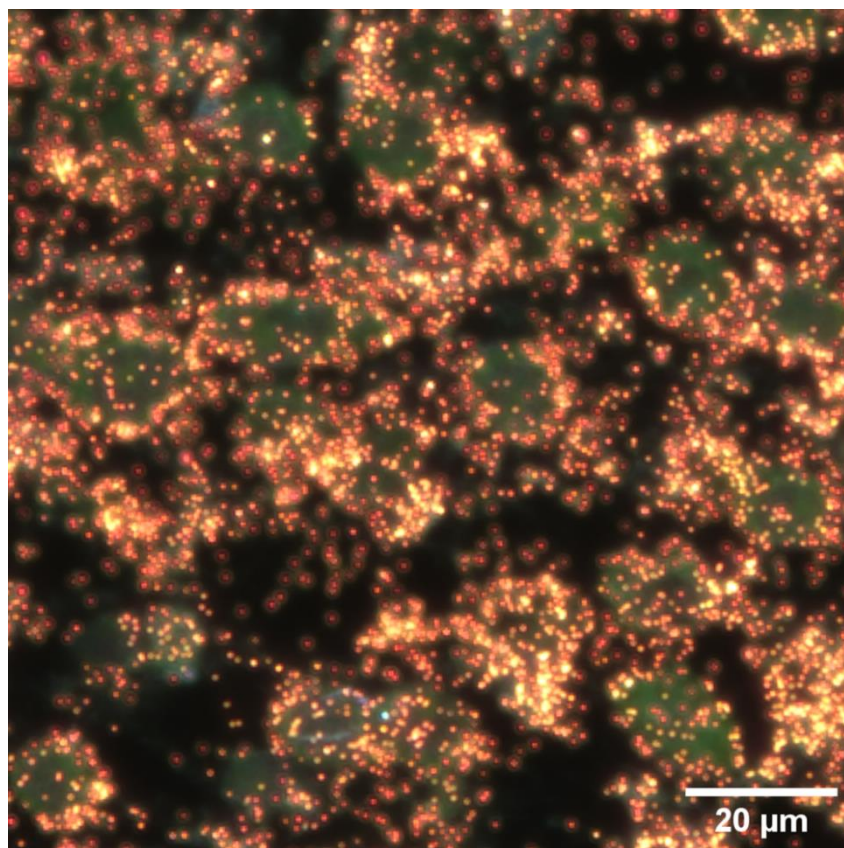


Figure 4.3 – **Image de test #5** - Cellules SKBR-3 sans marquage H&E à haute concentration de NPs

Une fois cette vérité de terrain manuelle générée, il est possible d'évaluer les masques obtenus par les différents modèles. L'approche employée pour l'évaluation est une comparaison de la position des nanoparticules réelles provenant de la vérité manuelle à la position des NPs prédites provenant des prédictions de modèles.

Pour ce faire, il faut comparer chaque NP détectée par le modèle à la NP correspondante dans le masque de vérité afin d'obtenir le nombre de détections qui correspondent à des faux positifs (FP), des faux négatifs (FN) ou des vrais positifs (VP). Un schéma de cette approche est présenté à la Figure 4.4.

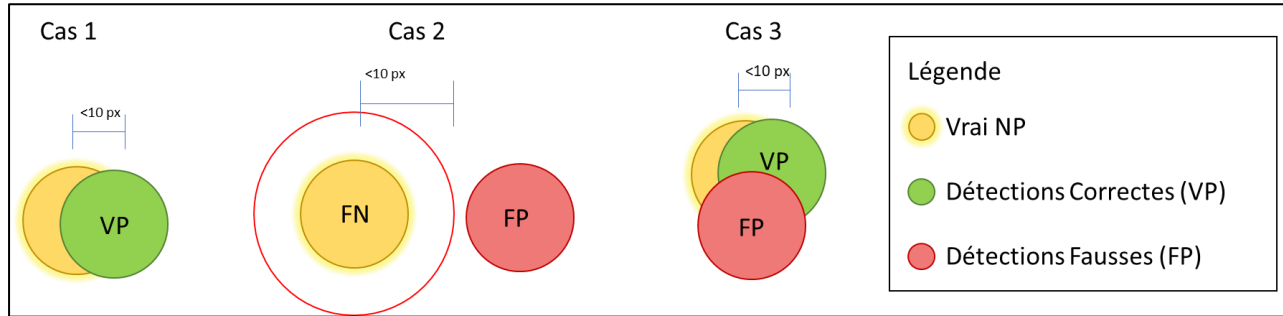


Figure 4.4 – Schéma de l'évaluation des détections par rapport au masque de vérité manuel.

L'obtention des VP, FP et FN permet d'évaluer la précision (aussi appelé valeur prédictive positive en statistique) et le *recall* (aussi appelé sensibilité) qui permettent de caractériser la performance de chaque modèle par une métrique combinée : le *FScore* (aussi appelé moyenne harmonique). La précision est décrite par l'Équation 4.1 et elle permet d'évaluer à quel point un modèle est correct lorsqu'il identifie les pixels comme étant associé à une NP. Le *recall* est décrit par l'Équation 4.2 et permet d'évaluer à quel point le modèle détecte un nombre de NPs correctement élevé par rapport au nombre de NPs réel. Finalement, le *FScore*, décrit par l'Équation 4.3, est une mesure combinant la précision et le *recall*.

$$\text{Précision} = \frac{VP}{VP + FP} \quad (\text{Équation 4.1})$$

$$\text{Recall} = \frac{VP}{VP + FN} \quad (\text{Équation 4.2})$$

$$F - \text{Score} = 2 \times \frac{\text{Précision} \times \text{Recall}}{\text{Précision} + \text{Recall}} \quad (\text{Équation 4.3})$$

Les 27 modèles peuvent ensuite être comparés par leur *FScore*. Afin de permettre une évaluation de l'impact de différents hyperparamètres, certaines paires de modèles comparables dont seulement 1 paramètre diffère peuvent être utilisées. Pour chacun des hyperparamètres testés, les modèles

comparables sont inscrits au Tableau 4.7. À noter que pour les architectures, les 9 modèles sont tous comparables les uns avec les autres (seulement l'architecture varie).

Tableau 4.7 – Combinaisons d'hyperparamètres comparables pour la comparaison des modèles

Structure (<i>Backbone</i>)	Taille de Parcelles (<i>Patches</i>)	Taille de lots (<i>Batches</i>)	Fonction d'erreur (<i>Loss</i>)	Optimiseur
[A, B]	[C, H]	[A, D]	[B, F]	[F, G]
[D, C]		[B, C]	[C, E]	
		[E, F]		

Les performances des différents modèles comparables sont évaluées en utilisant des graphiques de type boîte à moustache (*boxplots*). Ces graphiques permettent d'obtenir l'emplacement des extrémités de chaque groupe de données ainsi que l'emplacement des différents quartiles (25%, 50% - médiane, 75%) décrivant la distribution du groupe de donnée. La boîte sera formée entre le 1^{er} et le 3^e quartile. La comparaison des hyperparamètres sera donc faite en comparant la position des boîtes pour chaque groupe de modèles.

4.5 Obtention du meilleur modèle

Une fois tous les modèles entraînés et évalués, il est possible de sélectionner les modèles ayant performé le mieux sur les différentes images de test. Les paramètres ayant permis d'obtenir les 3 meilleurs modèles (correspondant à environ 10% des modèles) peuvent ensuite être réentraînés avec différentes séparations des données (toujours aléatoires). On parle alors de *Cross-Validation(CV)*. Une fois la *cross-validation* ($N = 5$) effectuée, le modèle obtenant les meilleurs résultats sur les données de test parmi tous les modèles testés est considéré comme le plus performant. La performance du meilleur modèle est ensuite comparée à l'algorithme *Find Maxima*.

4.6 Représentation des détections

Une fois que le meilleur modèle est déterminé, il est possible d'évaluer des échantillons tissulaires en illumination latérale afin de détecter les NPs, puis de les représenter dans une image de microscopie en champ clair de l'échantillon. Dans un contexte d'utilisation clinique, ceci permet une représentation des données dans un mode d'observation familier pour les pathologistes qui

évaluent des échantillons. La méthode employée pour représenter la distribution des NPs sur l'échantillon est de réaliser un graphique de densité des NPs avec une certaine transparence et de le superposer à l'échantillon en champ clair. La méthode de génération de ce graphique de densité de NPs passe par l'utilisation de la librairie Python *Seaborn* qui permet de générer des graphiques de *kernel density estimation (KDE)* par la fonction *seaborn.kdeplot()*. Les paramètres de cette fonction, déterminés par essai erreur, permettant de représenter adéquatement le marquage par NPs sont les suivants : $\alpha = 0,4$ – permet d'ajuster la transparence de la distribution KDE; $\text{thresh} = 0,2$ – permet de déterminer à partir de quel seuil de densité de NPs on désire afficher la carte de densité; $\text{levels} = 10$ – nombre de niveaux de densité représentés; $\text{bw_adjust} = 0,12$ – bande passante de la méthode KDE (une plus basse valeur pour ce paramètre permet d'obtenir des distributions plus serrées autour des détections).

CHAPITRE 5 RÉSULTATS

Cette section de résultats débute par une évaluation rapide de l'algorithme de détection actuellement utilisé en laboratoire pour effectuer le décompte des NPs sur des échantillons. Ceci est effectué en comparant la performance de l'algorithme *Find Maxima* sur les images de test en fonction de son seuil de tolérance au bruit (valeur d'intensité de pixel permettant de sélectionner les maxima importants).

Ensuite, les performances des 27 modèles sur les quatre images de test seront d'abord comparées de manière sommaire par la moyenne de leur précision, de leur *recall* et de leur *FScore*.

Les différents modèles seront ensuite comparés en fonction de leurs hyperparamètres. Cette section comprendra les hyperparamètres généraux (architecture, structure d'encodage (*backbone*) et taille de parcelle (*patch size*)), puis les hyperparamètres d'entraînement (taille de lot (*batch size*), optimiseur et fonction d'erreur (*Loss*)). Tous ces hyperparamètres seront comparés en trois temps :

- 1- Comparaison générale (le *FScore* de chaque modèle sur chacune des 4 images de test (27 modèles x 4 images)) de tous les modèles qui comprennent l'hyperparamètre évalué.
- 2- Comparaison des modèles comparables (idem qu'en 1, mais seulement les modèles dont l'unique paramètre variable est le paramètre évalué sont montrés).
- 3- Comparaison des moyennes des *FScores* (moyennes sur les quatre images, puis sur les modèles s'il y a plus d'un modèle) pour les modèles comparables.

Par la suite, les meilleurs modèles seront utilisés pour effectuer un entraînement fin par validation croisée. Le meilleur modèle est ensuite comparé à *Find Maxima* sur les images de test. Ce modèle est aussi comparé à *Find Maxima* sur des échantillons de l'image de test #5 afin de voir la performance de détection de l'algorithme à haute densité de NPs. Finalement, des résultats exploratoires de visualisation du marquage IP sont présentés.

5.1 Évaluation de l'algorithme naïf actuel – *Find Maxima*

Il est possible d'évaluer la capacité de *Find Maxima* à tolérer des variations dans les images en comparant des images contenant différentes concentrations de NPs. Afin d'évaluer cela, nous pouvons fixer le seuil de tolérance au bruit de l'algorithme afin qu'il soit optimisé pour une image, puis comparer la performance de l'algorithme sur une nouvelle image ayant une densité de NPs

différente. Ceci est effectué par les figures (Figure 5.1 et Figure 5.2) qui montrent les lacunes de ce type d'algorithme naïf .

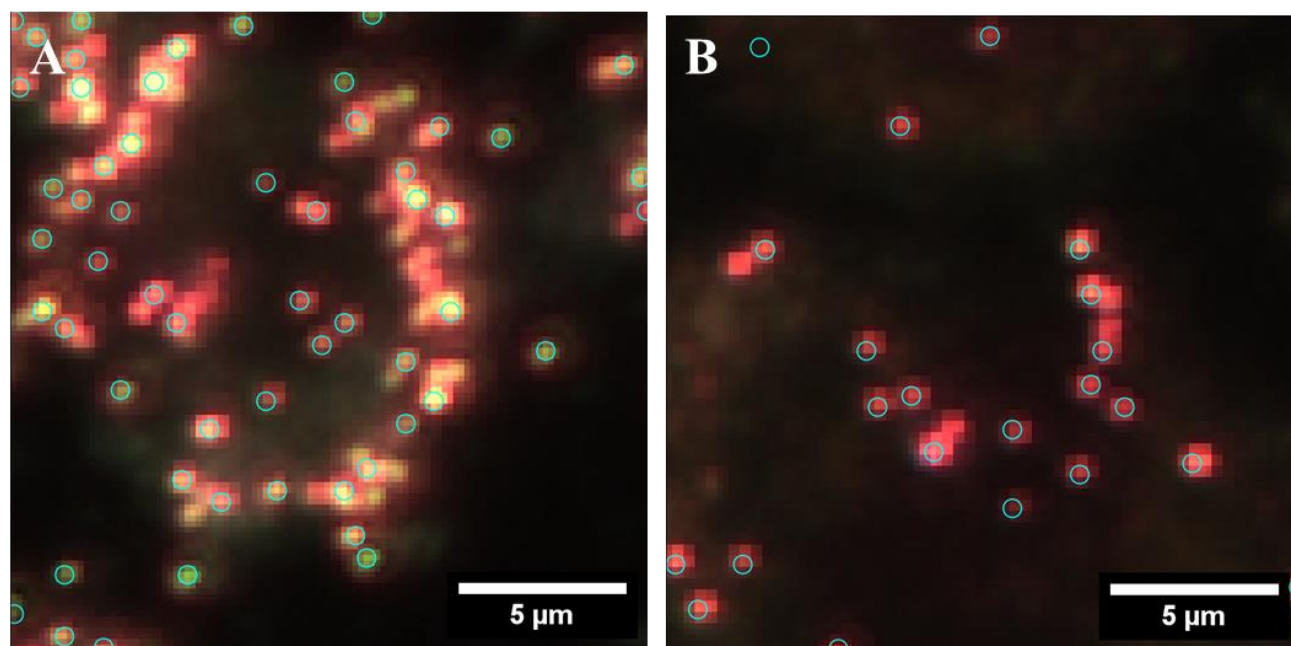


Figure 5.1 – Limites de détection de l'algorithme *Find Maxima* de NPs sur cellules SKBR-3 avec un seuil de tolérance au bruit optimisé pour l'image de droite. A) Image à haute densité de NPs.
B) Image à faible densité de NPs

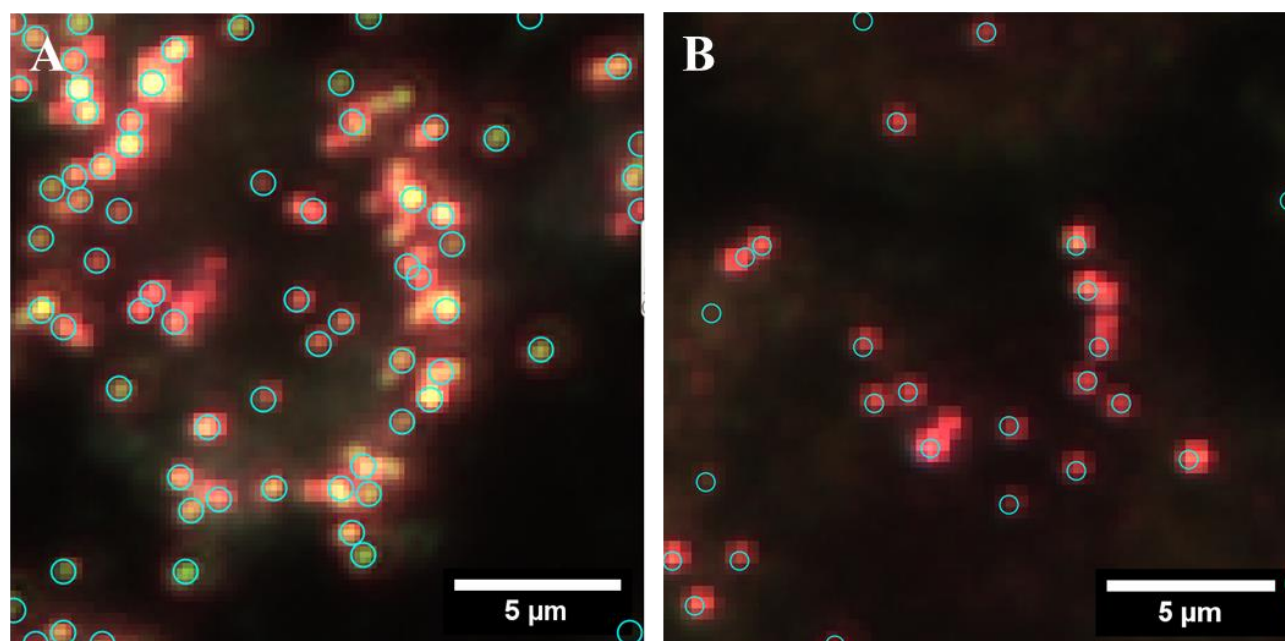


Figure 5.2 – Limites de détection de l’algorithme *Find Maxima* de NPs sur cellules SKBR-3 avec un seuil de tolérance au bruit optimisé pour l’image de gauche. A) Image à haute densité de NPs. B) Image à faible densité de NPs

Comme on peut le voir à la Figure 5.1, l’algorithme *Find Maxima* performe différemment pour une image à haute densité de NPs que pour à une image à basse densité de NPs. L’algorithme semble manquer un bon nombre de NPs avec le seuil optimisé pour l’image à basse densité. À la Figure 5.2, on peut voir que l’ajustement du seuil de tolérance au bruit de *Find maxima* à l’image à haute densité provoque l’apparition de fausses détections dans l’image à faible densité de NPs. Ce faisant, la détection de NPs est limitée par la performance de l’algorithme de détection utilisé. De plus, l’utilisation de cet algorithme dont la performance varie en fonction du seuil établi rend son utilisation plus fastidieuse puisqu’elle requiert l’ajustement manuel du seuil de tolérance au bruit et peu propice à une utilisation clinique. Ceci vient limiter la plage dynamique de la méthode IP.

Il est possible de confirmer ces observations à l’aide d’une évaluation de l’algorithme naïf *Find Maxima*. Pour ce faire, différents seuils de tolérance au bruit peuvent être utilisés pour la détection de NPs, puis leur précision, *recall* et *FScores* peuvent par la suite être évalués. De manière expérimentale, il a été déterminé que les seuils les plus performants pour les images de test se situent entre 3 et 19 (en valeur d’intensité de pixel pour des images 8 bits). La Figure 5.3 permet de constater l’impact de l’ajustement du seuil de tolérance au bruit de l’algorithme naïf *Find*

Maxima sur la précision et le *recall* lors de la détection de NPs. Ensuite, la Figure 5.4 montre la performance de *Find Maxima* avec ce même intervalle de seuils de tolérance au bruit.

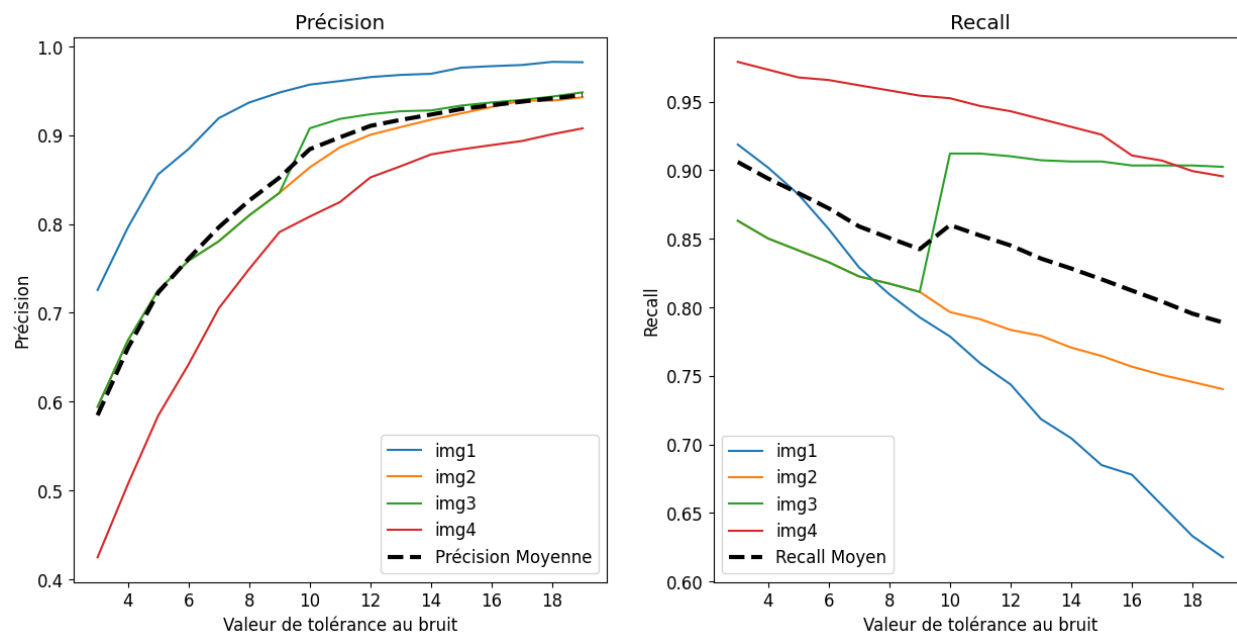


Figure 5.3 – Comparaison de la précision (gauche) et du *recall* (droite) de l’algorithme *Find Maxima* sur les images de test #1 à #4 pour un intervalle de valeurs du paramètre de tolérance au bruit de l’algorithme naïf.

Il est possible de constater que le seuil de tolérance au bruit a un grand impact sur la précision et le *recall* de l’algorithme naïf. De manière générale, plus le seuil de tolérance au bruit augmente, plus la précision (capacité de l’algorithme à éviter les fausses détections) augmente. Cependant, plus cette valeur de tolérance au bruit augmente, plus le *recall* (capacité à détecter les NPs et de ne pas en oublier) diminue. Ceci est attendu, puisque plus la valeur de tolérance au bruit est près de 0, plus il devient facile de détecter des maxima. De plus, une valeur de tolérance au bruit faible fait en sorte qu’il devient difficile de faire la distinction entre du bruit dans l’image et des NPs.

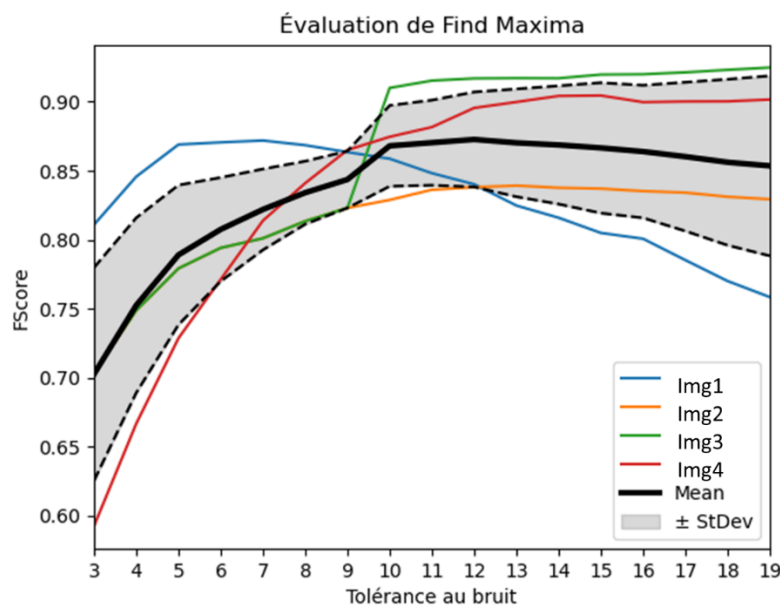


Figure 5.4 – *FScores* de l'algorithme naïf *Find Maxima* selon différents seuils de tolérance au bruit lors de la détection de NPs dans les images de test.

La Figure 5.4 qui utilise le *FScore* pour comparer les différents seuils de tolérance au bruit de l'algorithme *Find Maxima* permet de combiner la précision et le *recall* afin d'obtenir le meilleur algorithme naïf. Ce dernier se trouve au seuil de tolérance au bruit d'une valeur de 12 (en intensité de pixel). Toutefois, il est possible de constater que cette valeur n'est pas idéale pour l'image #1 qui obtient sa meilleure performance à un seuil de tolérance au bruit de valeur 7. Si cette valeur avait été utilisée pour détecter des NPs sur les images #2, #3 et #4, la performance obtenue (*FScore*) par l'algorithme naïf n'aurait pas été maximisée. Ceci démontre l'une des problématiques majeures associées à la détection de NPs par un algorithme naïf, soit le fait qu'il doit être ajusté (seuil de tolérance au bruit) pour chaque échantillon, puisqu'il n'est pas tolérant à des variations dans les échantillons (variations qui peuvent être dues à des épaisseurs d'échantillons, au plan focal lors de l'acquisition, à la densité de NPs, etc.).

Une fois *Find Maxima* évalué, il est possible de procéder à l'évaluation et à la caractérisation des modèles d'IA pour la détection de NPs.

5.2 Résultats d'entraînement

L'entraînement d'un modèle peut en révéler beaucoup sur sa capacité à effectuer des prédictions sur de nouvelles données. En effet, lorsque la fonction d'erreur (*Loss*) associée aux données d'entraînement continue de s'améliorer, tandis que la fonction d'erreur associée aux données de validations ne s'améliore plus, le modèle est probablement en train de trop s'ajuster aux données d'entraînements. Ceci limite potentiellement sa capacité de généralisation et le modèle est donc en train d'effectuer de *l'overfitting*. Il est possible de voir quels paramètres affectent ou non la vitesse à laquelle le modèle converge. Pour ce faire, seulement les modèles entraînés avec la fonction d'erreur *Dice* seront comparés dans cette section. La Figure 5.5. permet de comparer les fonctions d'erreur *Dice* sur les données d'entraînements et de validations de différents modèles en fonction des architectures de modèles, des structures d'encodages et du paramètre de la taille de lot (*batch size*).

La valeur de la fonction d'erreur qui est représentée sur l'axe des ordonnées des graphiques présentés aux Figure 5.5, Figure 5.6 et Figure 5.7 est une valeur d'erreur (*Dice Loss*) située entre 0 et 1 que l'on tente de minimiser lors de l'entraînement du modèle. Il est possible de remarquer tout au long de cette section que les modèles atteignent une limite près de 0,5. Cette limite est possiblement due à une mauvaise qualité des masques, qui fait en sorte que les modèles ne sont pas capables d'obtenir une meilleure détection des pixels associés au centre des NPs. Il faut aussi noter que cette métrique est pertinente pour l'entraînement des modèles, mais qu'elle ne permet pas d'évaluer la performance de ces modèles dans le contexte de détection de NPs. En effet, l'objectif final est d'obtenir une position pour chaque nanoparticule, le nombre de pixels correctement classés associés à une nanoparticule nous importe peu. La valeur de l'axe des abscisses représente le nombre d'itérations.

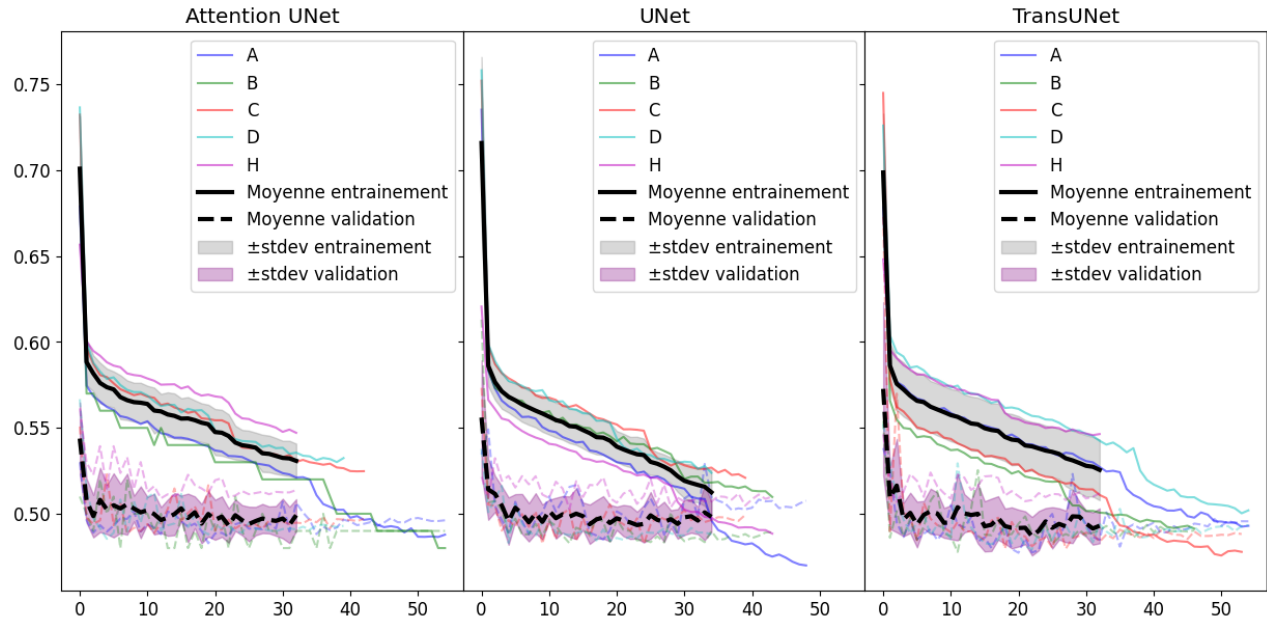


Figure 5.5 – Comparaison de l’entraînement en fonction de l’architecture. Les courbes d’entraînement des modèles individuels sont en couleur. Les modèles utilisés pour cette comparaison sont les modèles dont la fonction d’erreur est la *Dice Loss*.

Il est possible de voir à la Figure 5.5 que les différentes architectures ont peu d’impact sur la dynamique d’entraînement. On peut toutefois remarquer que les modèles de type *TransUNet* semblent avoir plus de variabilité dans la vitesse à laquelle ils convergent. Cette étape de vérification permet d’assurer une performance adéquate des modèles et confirme leur capacité à effectuer un apprentissage sur les données fournies.

Il est ensuite possible de vérifier l’impact des différents hyperparamètres sur l’entraînement. En commençant par la taille de lot (*batch size*) pour des paires de modèles correspondants (dont seulement le paramètre de taille de lot est différent). Les fonctions d’erreur lors de l’entraînement de ces différents modèles comparables sont présentées à la Figure 5.6.

Rappel : La taille de lot représente le nombre de parcelles utilisées par étape d’entraînement lors d’une itération du modèle.

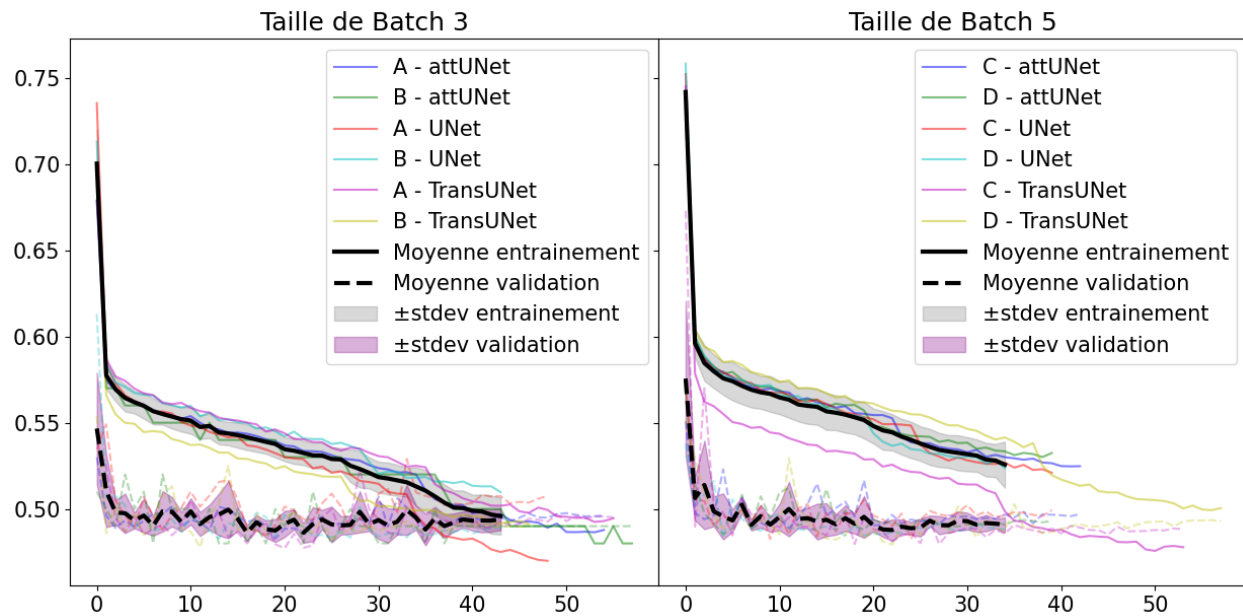


Figure 5.6 – Comparaison de l’entraînement en fonction de la taille de lot (*batch*). Les courbes d’entraînement des modèles individuels sont en couleur. Les modèles utilisés pour cette comparaison sont les modèles dont la fonction d’erreur est la *Dice Loss*.

La Figure 5.6 permet de visualiser l’impact de la taille de batch sur l’entraînement. En effet, les modèles semblent atteindre une certaine convergence, et ce peu importe la taille de lot utilisée. Il est possible de remarquer une variabilité légèrement plus importante pour les modèles dont la taille de lot (*batch*) est 5.

Il est possible de poursuivre cette analyse en comparant des modèles correspondants ayant différentes structures d’encodages (*backbones*). Cette comparaison est montrée à la Figure 5.7, où les modèles contenant les structures d’encodages *VGG16*, *VGG19* et *ResNet50* sont comparés.

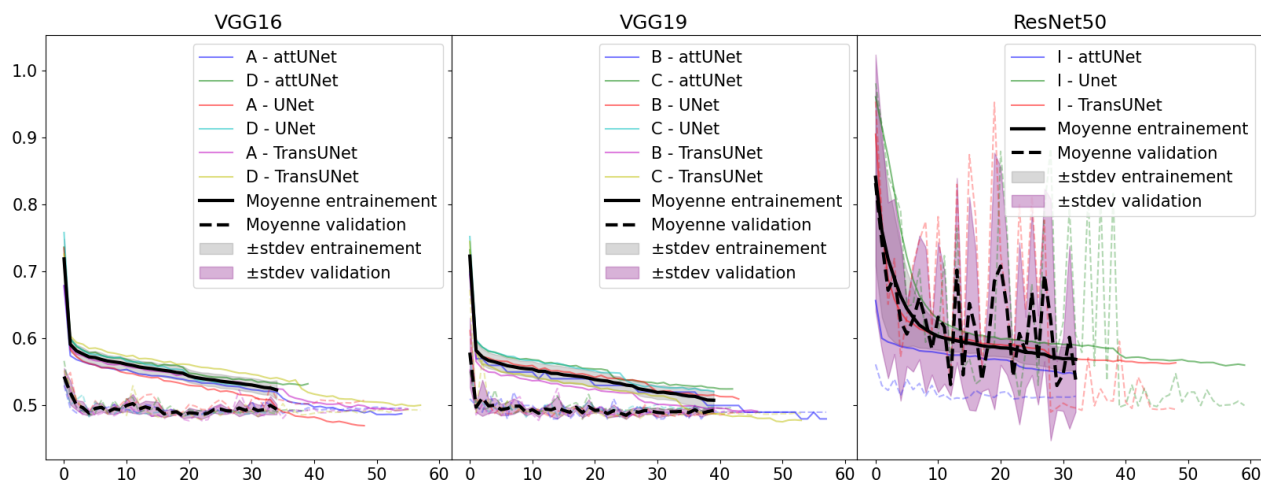


Figure 5.7 – Comparaison de l’entraînement en fonction des structures d’encodages (*backbones*). Les courbes d’entraînement des modèles individuels sont en couleur. Les modèles utilisés pour cette comparaison sont les modèles dont la fonction d’erreur est la *Dice Loss*.

La Figure 5.7 permet de voir l’impact des structures d’encodages (*backbones*) sur l’entraînement. Premièrement, l’entraînement des modèles ayant pour structure d’encodage *VGG16* ou *VGG19* semble très similaire, quoique *VGG19* semble prendre plus d’itération avant d’atteindre un plateau. Cependant, il semblerait que la fonction d’erreur de validation pour la structure *ResNet50* soit très lourdement bruitée. Ceci peut signifier un taux d’apprentissage trop élevé pour le modèle, mais aussi que le modèle a tendance à se retrouver dans des minimums locaux de la fonction d’erreur lors de l’entraînement. Ceci pourrait rendre le modèle plus à risque d’*overfitting*, puisque cela signifie qu’en essayant de minimiser l’erreur pour les données d’entraînement, il lui arrive d’augmenter considérablement son erreur lorsqu’il effectue des prédictions sur des données de validation. Si l’entraînement est arrêté à une itération où le modèle a une erreur élevée sur les données de validation, il est peu probable que ce modèle soit généralisable à de nouvelles données.

5.3 Évaluation préliminaire des modèles

Une fois les modèles entraînés, il est ensuite possible de comparer leur performance sur différentes images auxquelles ils n’ont jamais été exposés. Ces images représentent les données de test décrites préalablement au Tableau 4.2 de la section Méthodologie. Pour chacun des 27 modèles entraînés, une prédiction est effectuée sur chaque image de test. La performance de détection de NPs est ensuite comparée à un masque de vérité manuel. Ceci permet de calculer les performances des

modèles sur chaque image de test par les 3 critères d'évaluation mentionnés dans la méthodologie, soit la précision, le *recall* et le *FScore*. Quoique chaque métrique soit importante pour évaluer la performance du modèle, le *FScore* est la mesure la plus globale que nous possédons pour cette tâche. Il est donc possible d'effectuer une évaluation préliminaire des modèles en comparant la moyenne de leurs *FScores* obtenus lors de prédictions sur chacune des 4 images de test. Cette comparaison globale peut-être visualisée à la Figure 5.10.

Rappel : Un *FScore* idéal a une valeur de 1 ou 100%, alors que le pire *FScore* possible prendrait une valeur de 0.

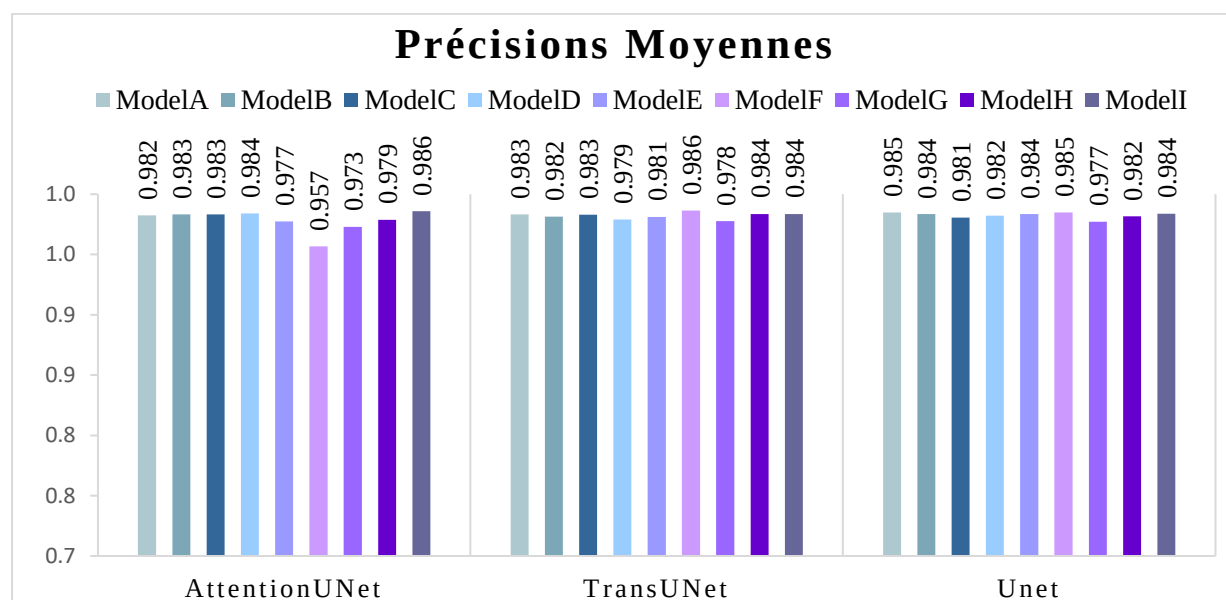


Figure 5.8 – Précisions moyennes (sur les quatre images de test) des 9 combinaisons d'hyperparamètres pour chacune des trois architectures principales (27 modèles évalués sur 4 images).

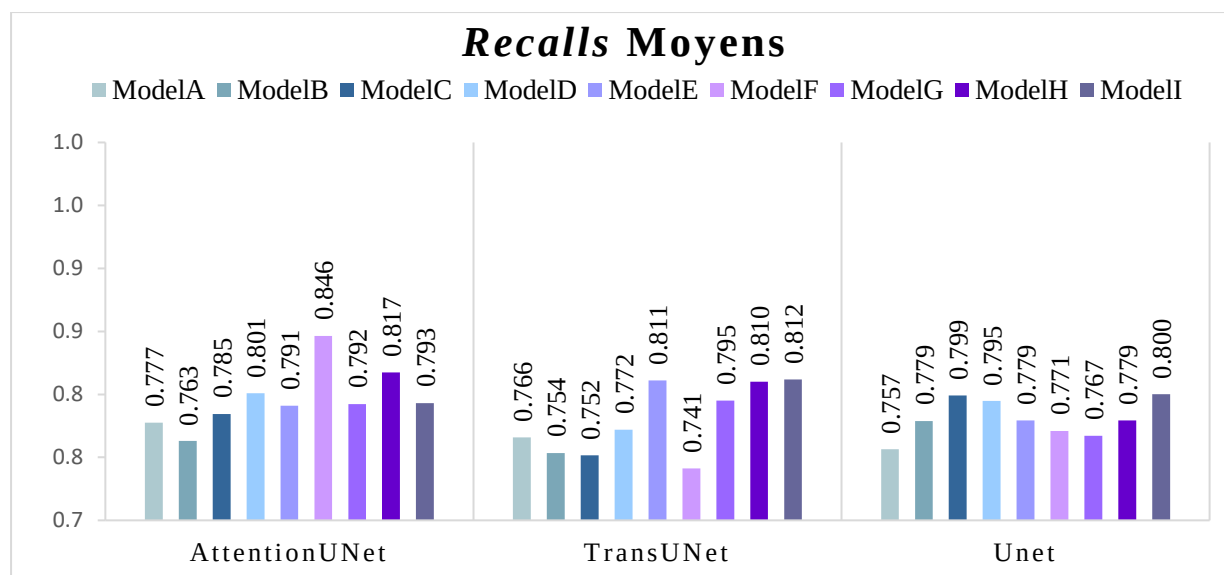


Figure 5.9 - *Recalls* moyens (sur les quatre images de test) des 9 combinaisons d'hyperparamètres pour chacune des trois architectures principales (27 modèles évalués sur 4 images).

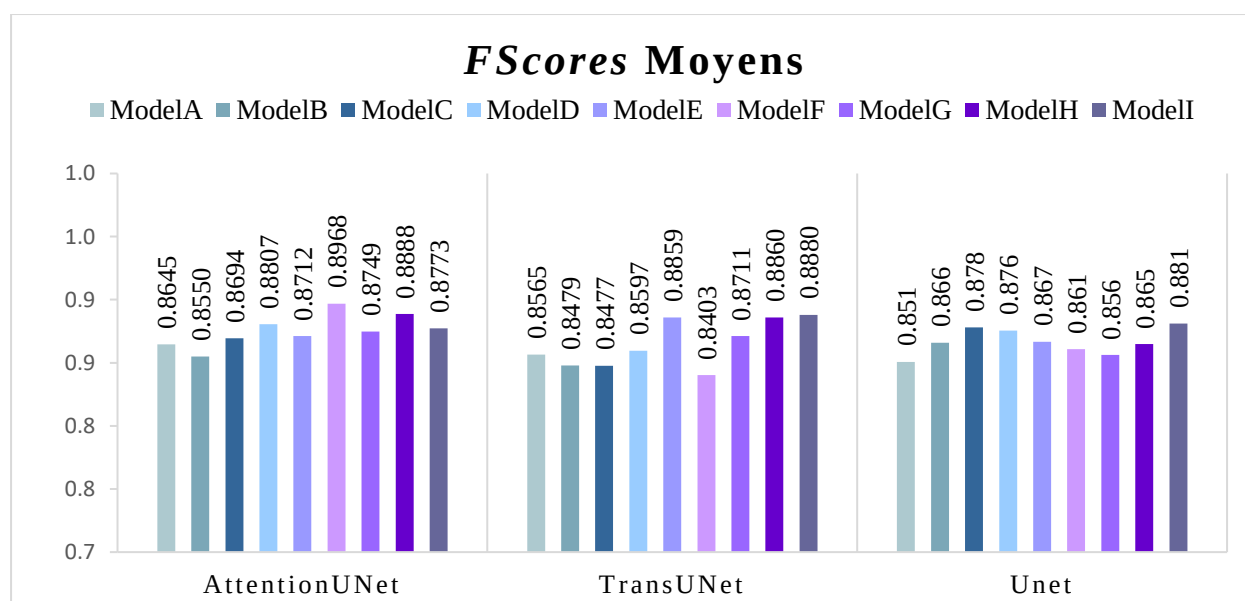


Figure 5.10 – *FScores* moyens (sur les quatre images de test) des 9 combinaisons d'hyperparamètres pour chacune des trois architectures principales (27 modèles évalués sur 4 images).

Les trois figures précédentes (Figure 5.8, Figure 5.9 et Figure 5.10) permettent de constater la performance des modèles résultants des différentes combinaisons d'hyperparamètres. Premièrement, il est important de remarquer que la précision est très souvent élevée pour les modèles entraînés par rapport au *recall*. Ceci nous indique que les modèles ont une certaine difficulté à détecter toutes les NPs d'un échantillon (beaucoup de FN), mais qu'ils font rarement de fausses détections (peu de FP). Ceci signifie que les différences de *FScores* entre les différents modèles ne sont pas toujours représentatives de l'écart de performance entre deux modèles. Toutefois, puisque la comparaison finale se fera par les *FScores*, cette métrique est utilisée tout au long de la section résultats.

Les différences entre les meilleurs et les pires modèles peuvent sembler faibles lors de la comparaison des *FScores* (0,840 pour le pire modèle contre 0,897 pour le meilleur), mais lorsque l'on compare les *recalls*, on constate que la différence entre *Attention UNet* - F et *TransUNet* - F passe à plus de 10 % (0,741% contre 0,846% pour le meilleur modèle).

De la Figure 5.10, il est possible d'identifier les modèles performants et moins performants. Cette moyenne des *FScores* nous indique que les modèles F et H pour l'architecture *Attention UNet* et les modèles E, H et I pour l'architecture *TransUNet* semblent à première vue les plus performants. Cette analyse préliminaire suggère que les modèles *UNet* sans mécanismes d'attention sont limités dans la performance qu'ils peuvent avoir pour cette tâche. De la Figure 5.10, il est aussi possible de remarquer une grande variabilité dans la performance des modèles ayant une architecture *TransUNet*. Les modèles B, C et F obtiennent de mauvais résultats alors que les modèles E, H et J obtiennent de bons résultats.

Tel que mentionné dans la section Méthodologie, une des quatre images, l'image #2 (Figure 4.3) est particulièrement différente des autres, puisque son acquisition a été effectuée avec des paramètres différents. Les modèles ont donc seulement été exposés à une poignée d'images avec des conditions similaires à celles de l'image de test #2. Il est donc attendu qu'un modèle qui a tendance à faire de l'*overfitting* (peu généralisable) ait plus de difficulté à performer sur cette image. Il est possible de voir ces différences de performances en évaluant les modèles sur cette image uniquement tel que montré à la Figure 5.11.

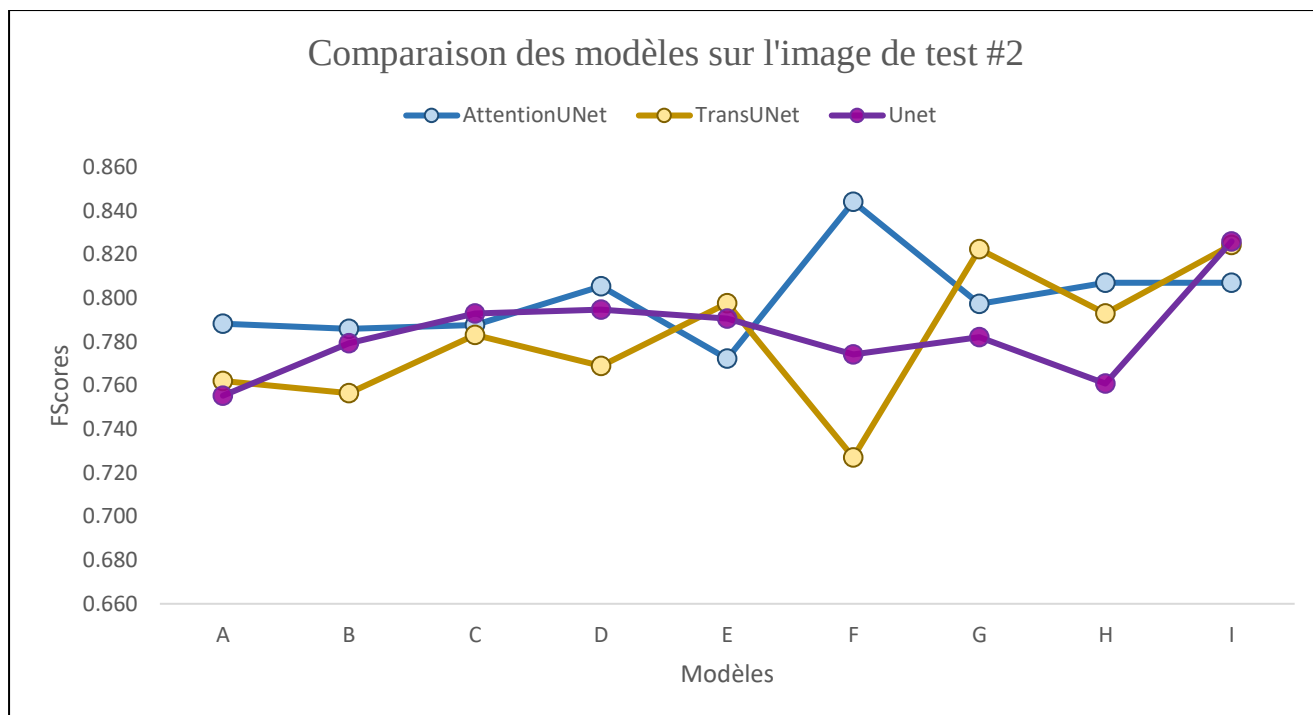


Figure 5.11 – Comparaison des performances des modèles sur l'image de test #2.

De la figure ci-dessus, il est possible de remarquer une grande différence entre la performance des modèles F provenant des architectures *Attention UNet* et *TransUNet*. La performance du modèle F *Attention UNet* est largement supérieure à la performance du modèle F *TransUNet*. De manière générale, l'*Attention UNet* semble performer mieux que le *TransUNet* sur cette image de test. En effet, la performance de l'*Attention UNet* est meilleure dans tous les cas sauf pour les modèles E, G et I.

5.3.1 Comparaison des hyperparamètres

Les deux sections suivantes (5.3.1.1 et 5.3.1.2) permettront de constater l'impact des différents hyperparamètres généraux et d'entraînements sur les performances des modèles résultants. La métrique utilisée pour effectuer cette comparaison sera le *FScore*, puisqu'elle sera la métrique utilisée lors de la comparaison finale avec les algorithmes naïfs. Il est toutefois important de rappeler qu'une bonne précision semble facilement atteignable pour les modèles « intelligents » présentés dans ce mémoire, mais qu'un bon *recall* (plus de 0,8) est plus difficile à atteindre. Ce faisant, de faibles différences de *FScores* peuvent parfois représenter une différence de *recall* qui est plus importante.

Les performances des modèles sur chaque image (4 images de test) sont représentées par des points qui seront répartis en fonction de leur *FScore* (axe des ordonnées). Ce faisant, pour chaque modèle comparé, il y a 4 points sur le graphique.

D'abord, toutes les performances (4 Images x 27 Modèles) sont affichées (en fonction de l'hyperparamètre que possède chaque modèle), ensuite, seules les performances de modèles dits comparables seront affichées. Pour rappel, les modèles dits comparables sont ceux dont uniquement l'hyperparamètre évalué varie entre les différents modèles.

Un exemple sur le paramètre « architecture » est présenté à la Figure 5.12. Dans cet exemple, les performances des modèles sur les différentes images de test (1 à 4) ont été indiquées pour constater qu'une concentration de point en haut de la boîte à moustaches ne signifie pas un modèle excellent, puisqu'il s'agit probablement tout simplement de l'image de test #3 qui semble plus facile à détecter. Il sera donc important dans cette comparaison des hyperparamètres de se baser sur la répartition des données (position des quartiles (25% , médiane et 75%) qui sont indiqués par la position de la boîte). Puisque l'impact des images n'est pas le facteur d'intérêt dans cette comparaison, elles ne seront pas indiquées dans les prochaines comparaisons. Les moyennes de *FScores* seront ensuite comparées entre les différents modèles comparables.

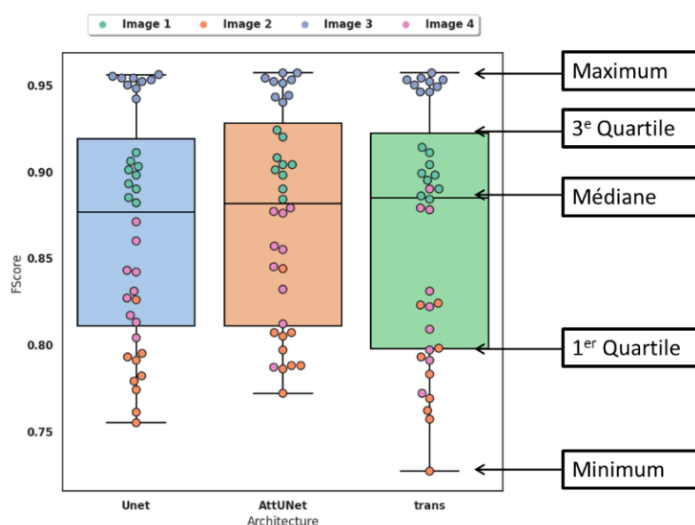


Figure 5.12 – Exemple de comparaison de la performance (*FScore*) des modèles sur les images de test en fonction de leur architecture. Dans cet exemple, les performances (*FScores*) des 9 combinaisons d'hyperparamètres sur chacune des 4 images de test sont évaluées. Ceci signifie que pour UNet par exemple, il y a 9 x 4 points représentant les *FScores* obtenus. Au total, dans l'ensemble du graphique, il y a 27 x 4 points représentant des performances de modèles sur les images de test.

5.3.1.1 Comparaisons des Hyperparamètres généraux

Les différents hyperparamètres généraux, tels que la structure d'encodage (*backbone*), la taille de lot (*batch size*) et la taille de parcelle (*patch size*) peuvent avoir une influence sur la capacité de prédiction des modèles. Les performances des modèles sont comparées en deux temps. Dans un premier temps, à la Figure 5.13, tous les modèles qui possèdent l'hyperparamètre évalué sont comparés. Dans un second temps, à la Figure 5.14, seulement des modèles comparables dont la seule différence est le paramètre d'intérêt sont comparés.

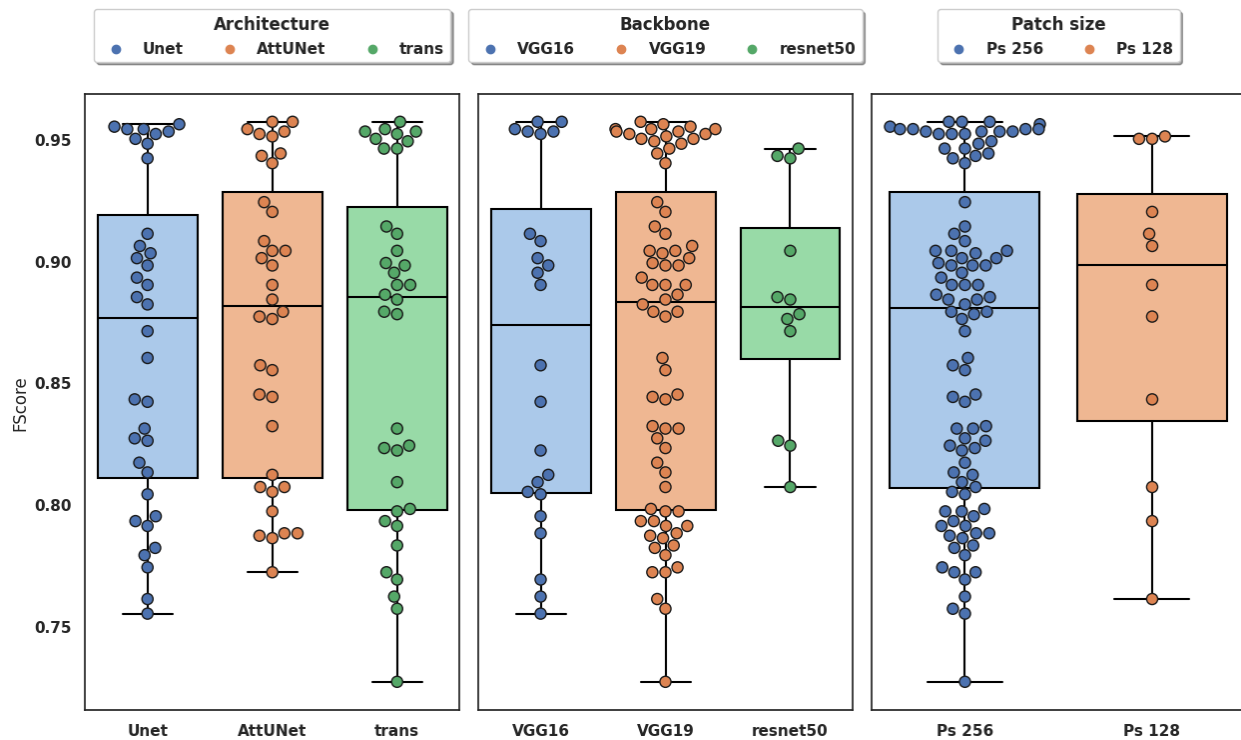


Figure 5.13 – Comparaison des *FScores* de tous les modèles sur les 4 images de test en fonction des différents hyperparamètres généraux. De gauche à droite, les hyperparamètres évalués sont : 1- L'architecture, 2 – La structure d'encodage (*backbone*), 3 - La taille de parcelle (*patch size*).

La Figure 5.13 permet d'observer un moins grand écart entre les meilleures et les pires performances des *Attention UNets* par rapport aux *UNets* et aux *TransUNets*.

Pour ce qui est des structures d'encodage (*backbones*), les performances semblent généralement meilleures pour la structure *ResNet50*, puisque les performances sont toutes situées au-dessus de 0,8 de *FScore*. Toutefois, il faut rester conscient du fait que les modèles avec *ResNet50* sont moins

représentés et qu'ils ne permettent pas non plus d'obtenir les performances les plus élevées (probablement une moins bonne performance sur l'image de test #3).

Pour les tailles de parcelles (*patch size*), il semblerait que les modèles ayant été entraînés avec une taille de 128 x 128 pixels performant mieux que les modèles entraînés avec une taille de parcelle (*patch*) de 256 x 256 pixels. En effet, le 1^{er} quartile et la médiane sont supérieurs pour cette taille (128 x 128).

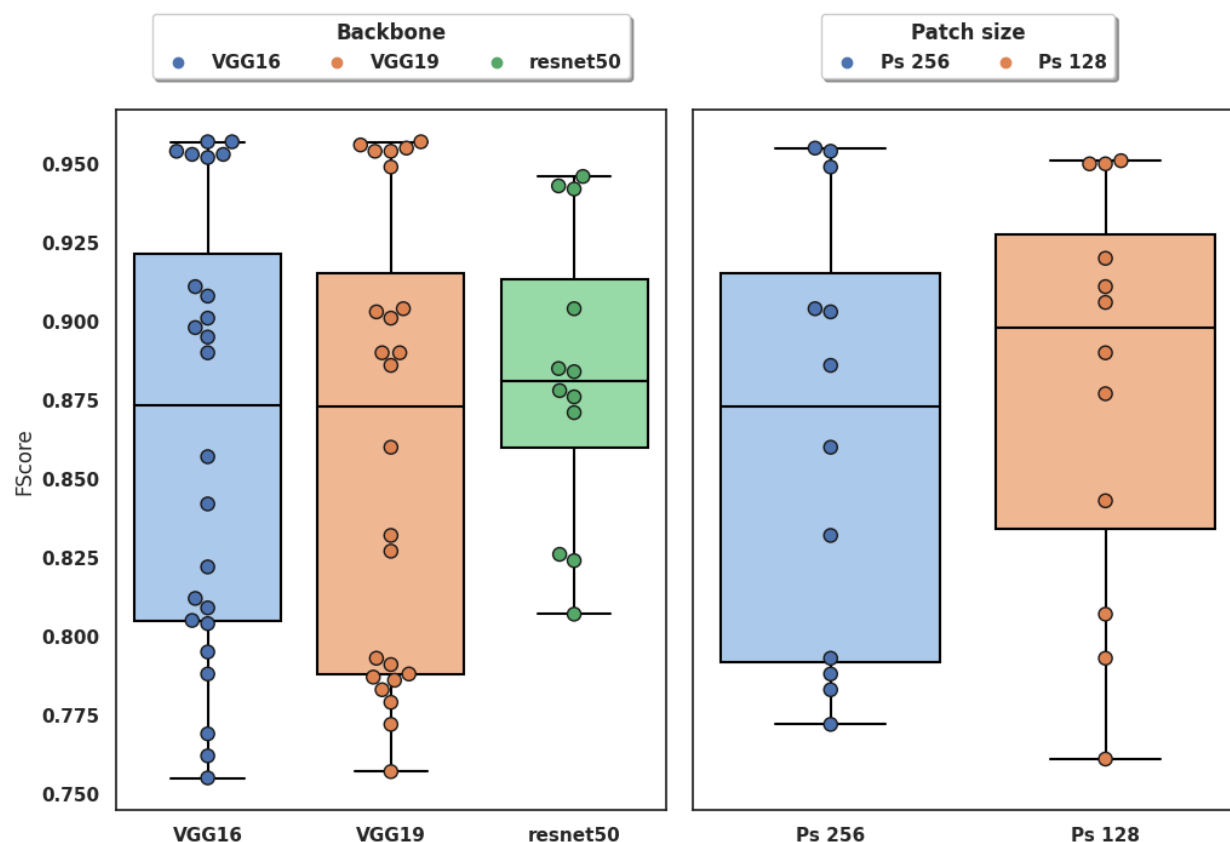


Figure 5.14 – *FScores* moyens de paires de modèles comparables selon les différents hyperparamètres généraux. Comparaison des *FScores* des modèles comparables sur les 4 images de test en fonction des différents hyperparamètres généraux. De gauche à droite, les hyperparamètres évalués sont : 1 – La structure d'encodage (*backbone*), 2 - La taille de parcelle (*patch size*)

La Figure 5.14 permet une comparaison plus fiable de l'influence des hyperparamètres sur la performance des modèles, puisqu'elle se limite à comparer uniquement les modèles dont seul le paramètre évalué diffère (voir Tableau 4.7). Elle permet de remarquer la performance de

l'architecture ResNet50 qui semble mieux performer que *VGG16* et *VGG19*, puisque la médiane et le dernier quartile (fin de la boîte) sont supérieurs à ceux des autres structures. *VGG16* semble performer légèrement mieux que *VGG19*, puisque le 1^{er} quartile se situe plus haut.

La taille de parcelle (*patch size*) semble aussi avoir eu un léger impact sur le *FScore*. En effet, pour la taille de 128 x 128, tous les quartiles sont situés plus haut que ceux des performances obtenues par la taille de 256 x 256.

Il est ensuite possible de comparer de manière très générale les modèles correspondants en évaluant leur *FScore* moyen pour ces différents hyperparamètres. Pour ce faire, au Tableau 5.1 sont comparées les performances en fonction des architectures et au Tableau 5.2 sont comparées les performances des modèles comparables obtenus avec les hyperparamètres généraux. La comparaison des architectures permet d'établir *l'Attention UNet* comme étant l'architecture permettant d'obtenir le plus constamment des modèles performants, tandis que les moyennes de performances de *TransUNet* et *UNet* sont comparables. La comparaison des hyperparamètres par les *FScores* moyens confirme la faible différence entre les modèles entraînés avec ces différents hyperparamètres, mais confirme aussi les observations précédentes. En effet, les modèles semblent avoir mieux performé avec *VGG16* qu'avec *VGG19*, mais encore mieux avec *ResNet50*. La taille de parcelle semble en effet avoir fait une différence (**0,880** contre 0,865).

Tableau 5.1 – Comparaison sommaire des *FScores* en fonction de l'architecture

Architecture	Attention UNet	TransUNet	UNet
Moyenne	0,875	0,865	0,867

Tableau 5.2 – Comparaison sommaire des hyperparamètres généraux par les moyennes des *FScores* de leurs modèles correspondants

	<i>Backbones</i>			Taille de <i>Patches</i>	
Hyperparamètres	VGG16	VGG19	ResNet50	128	256
Modèles et <i>FScores</i> Moyens	D = 0,872	C = 0,865	I = 0,882	H = 0,880	C = 0,865
	A = 0,857	B = 0,856			
Moyenne	0,8645	0,8605	0,882	0,880	0,865

5.3.1.2 Comparaison des hyperparamètres d'entraînement

Tout comme les paramètres généraux, les paramètres d'entraînement sont pertinents lorsqu'il s'agit d'évaluer leur impact sur les modèles résultants. Pour ce faire, les performances des 27 modèles sur les 4 images de test sont d'abord comparées en fonction des hyperparamètres de tous les modèles à la Figure 5.15. Ensuite, à la Figure 5.16 sont comparées les performances des modèles correspondants dont seulement l'hyperparamètre d'intérêt diffère.

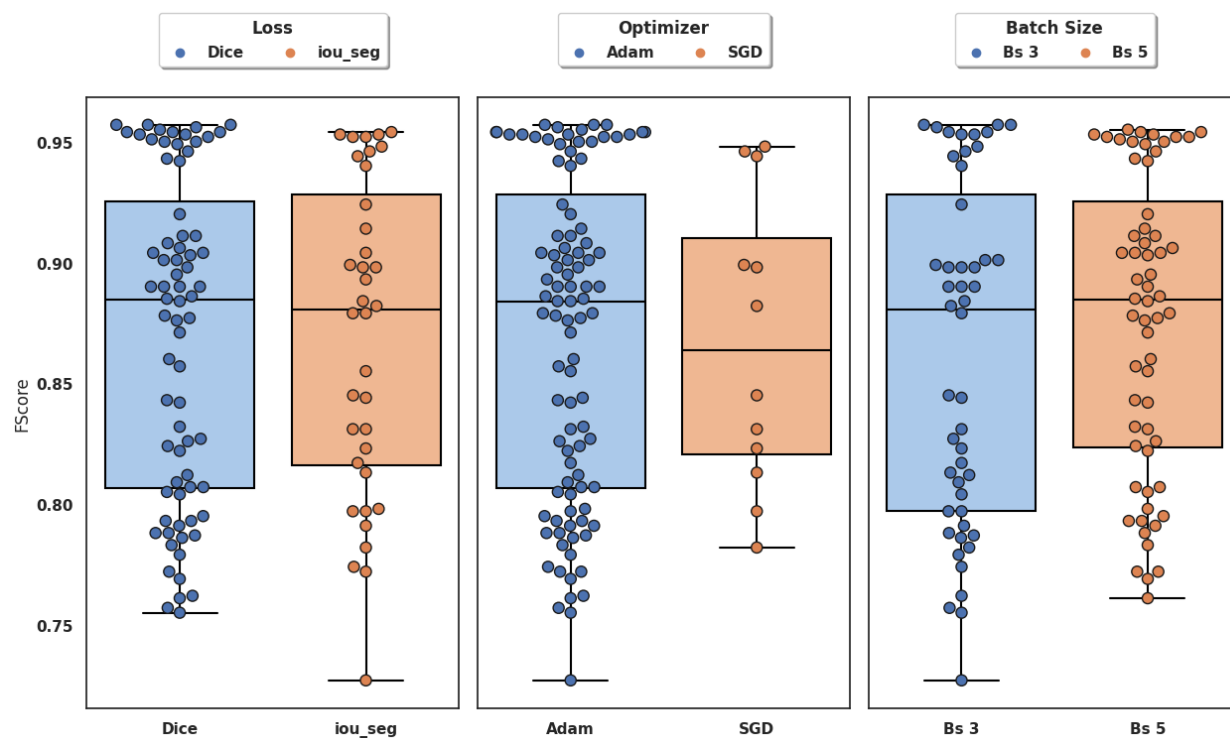


Figure 5.15 – Comparaison des hyperparamètres d’entraînements, de gauche à droite, les hyperparamètres testés sont : 1 – Fonction d’erreur (*Loss*), 2 – Optimiseur (*Optimizer*), 3 – Taille de lot (*Batch size*)

De la Figure 5.15, la comparaison des *FScores* obtenus avec différentes fonctions d’erreurs (à gauche) permet de voir une performance relativement équivalente entre les modèles entraînés par la fonction *Dice* et ceux entraînés avec la fonction *IOU*. Il est aussi possible de remarquer dans le graphique du centre que l’optimiseur *Adam* qui est beaucoup plus fréquent dans les modèles entraînés n’est pas nécessairement plus performant que *SGD* avec un léger momentum qui semble obtenir des *FScores* similaires. Malgré une différence dans la position des différents quartiles, il est difficile de comparer les deux groupes de performances compte tenu de la grande différence entre leur nombre de données.

Finalement, les modèles ayant une taille de lot de 5 lors de l’entraînement semblent légèrement meilleurs que ceux entraînés avec une taille de lot de 3, mais l’impact ne semble pas particulièrement important. En effet, la taille de lot de 5 possède une médiane légèrement supérieure, un premier quartile plus élevé et un intervalle de données plus compact que la taille de lot de 3.

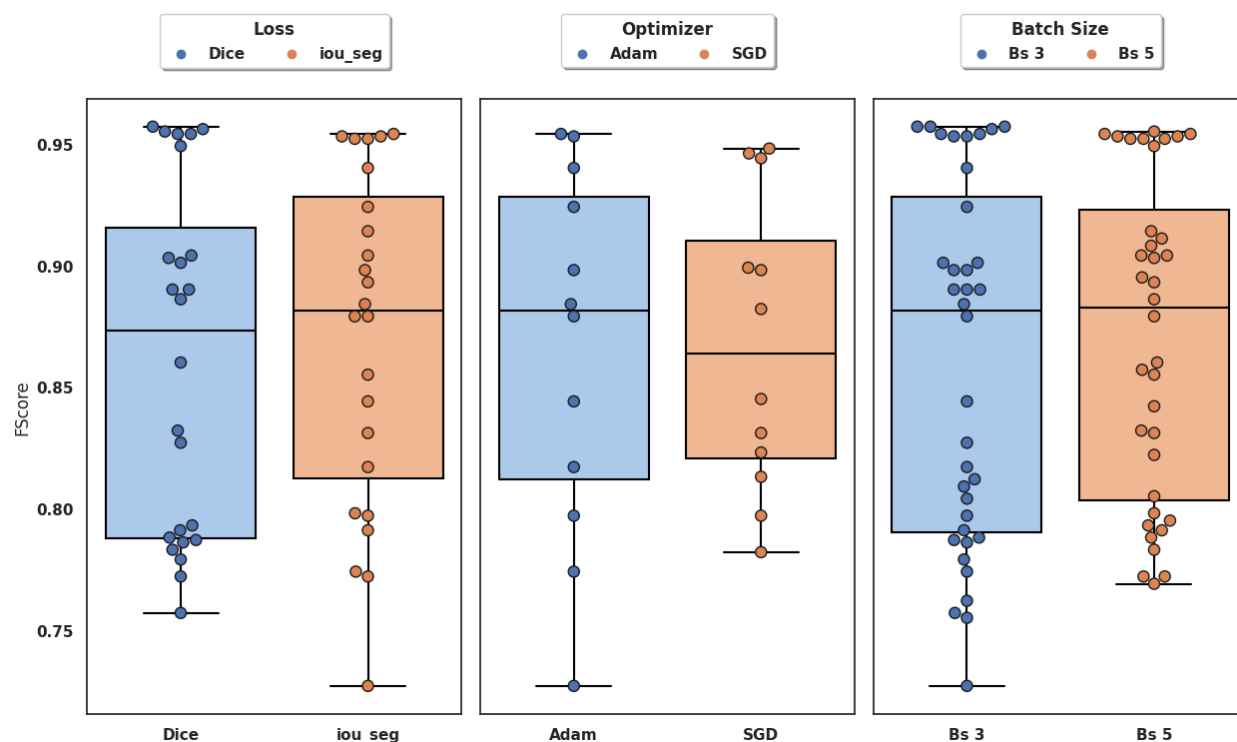


Figure 5.16 – Comparaison des hyperparamètres d’entraînement par des paires de modèles comparables. De gauche à droite, les paramètres évalués sont : 1 – Fonction d’erreur (*Loss*), 2 – Optimiseur (*Optimizer*), 3 – Taille de lot (*Batch size*)

La Figure 5.16 permet de mieux évaluer l’impact de ces hyperparamètres sur les modèles, puisque les modèles choisis sont dits comparables (voir Tableau 4.7). On peut voir à gauche que les performances associées à la fonction d’erreur *Dice* donnent des quartiles et une médiane inférieurs à ceux de la fonction d’erreur *IOU*. Il semblerait donc que la fonction *IOU* ait permis d’obtenir des modèles légèrement plus performants.

Pour les optimiseurs comparés à la Figure 5.16 (centre), on remarque une variabilité plus importante dans les modèles entraînés avec *Adam* que dans ceux entraînés par *SGD*, mais une meilleure médiane pour les modèles entraînés avec *Adam*.

Pour les tailles de lots (*batch size*), il semblerait que la taille ait peu d’influence sur la performance, mais que la taille de 5 permette tout de même d’obtenir un meilleur premier quartile.

La comparaison des moyennes au Tableau 5.3 permet de confirmer les performances équivalentes des deux optimiseurs, une performance en moyenne supérieure pour la fonction d'erreur *IOU*, ainsi qu'une performance légèrement supérieure pour la taille de lot de 5.

Tableau 5.3 – Comparaison des moyennes de *FScores* des modèles correspondants pour les différents hyperparamètres d'entraînement

	Fonction d'erreur (<i>Loss</i>)		Optimiseur		Taille de lots (<i>Batch</i>)	
Hyperparamètres	Dice	IOU	Adam	SGD	3	5
Modèles et <i>FScores</i> Moyens	B = 0,856	F = 0,866	F = 0,866	G = 0,867	A = 0,857	D = 0,872
	C = 0,865	E = 0,875			B = 0,856	C = 0,865
					E = 0,875	F = 0,866
Moyenne	0,8605	0,8705	0,866	0,867	0,8627	0,8677

Suite à cette analyse, il est possible de revenir sur les modèles les plus performants afin d'affiner l'entraînement de ces modèles pour extraire le meilleur modèle de détection possible. Les analyses précédentes ayant permis d'établir que les paramètres utilisés sont tous viables; puisqu'aucun hyperparamètre est associé à des performances réellement mauvaises. Il est tout de même possible de revenir sur ces observations. Pour les paramètres généraux, les meilleurs résultats furent obtenus par l'architecture *Attention UNet*, par la structure d'encodage (*backbone*) *ResNet50*, ainsi que par la taille de parcelle (*patch*) de (128 x 128) pixels.

Pour les paramètres d'entraînements, la fonction d'erreur *IOU* semble légèrement meilleure que *Dice*. Pour l'optimiseur, les résultats obtenus par *Adam* et *SGD* semblent très similaires, mais *Adam* permet d'obtenir une meilleure médiane, alors que *SGD* semble offrir une moins grande variabilité entre les différentes performances. Finalement, la taille de lot de 5 images par étape d'entraînement semble (d'après la comparaison des moyennes) avoir permis d'obtenir les meilleurs résultats.

En reprenant les valeurs de *FScores* moyens de la Figure 5.10, les modèles F et H pour l'architecture *Attention UNet* et le modèle I pour l'architecture *TransUNet* sont les plus performants. Les paramètres de ces modèles peuvent être réutilisés afin d'essayer d'entraîner un meilleur modèle en effectuant de la validation croisée. Tel que décrit dans la section 4.5, la validation croisée (*cross validation (CV)*) permet d'entraîner le modèle sur différentes sections des données qui n'auraient pas été utilisées pour l'entraînement ou la validation autrement. Les performances sur les images de test des modèles issus de chacune de ces validations croisées sont donc comparées afin d'extraire le meilleur modèle possible pour la détection de NPs.

5.4 Choix final du modèle

Les hyperparamètres ayant permis d'obtenir les meilleurs modèles, soit les modèles F et H de l'architecture *Attention UNet* et le modèle I de l'architecture *TransUNet* sont réutilisés ici afin de faire un entraînement plus fin du modèle. Pour ce faire, 5 modèles par groupe d'hyperparamètres sont entraînés sur des séparations différentes des données d'entraînement et de validation (voir Tableau 5.4. Les performances de ces modèles sur chacune des images de test sont comparées à la Figure 5.17.

Tableau 5.4 – Sommaire des séparations pour l'entraînement fin des modèles

Modèle	Séparation des données				
<i>Attention UNet</i> F	Séparation 1 (CV1)	Séparation 2 (CV2)	Séparation 3 (CV3)	Séparation 4 (CV4)	Séparation 5 (CV5)
<i>Attention UNet</i> H	Séparation 1 (CV1)	Séparation 2 (CV2)	Séparation 3 (CV3)	Séparation 4 (CV4)	Séparation 5 (CV5)
<i>TransUNet</i> I	Séparation 1 (CV1)	Séparation 2 (CV2)	Séparation 3 (CV3)	Séparation 4 (CV4)	Séparation 5 (CV5)

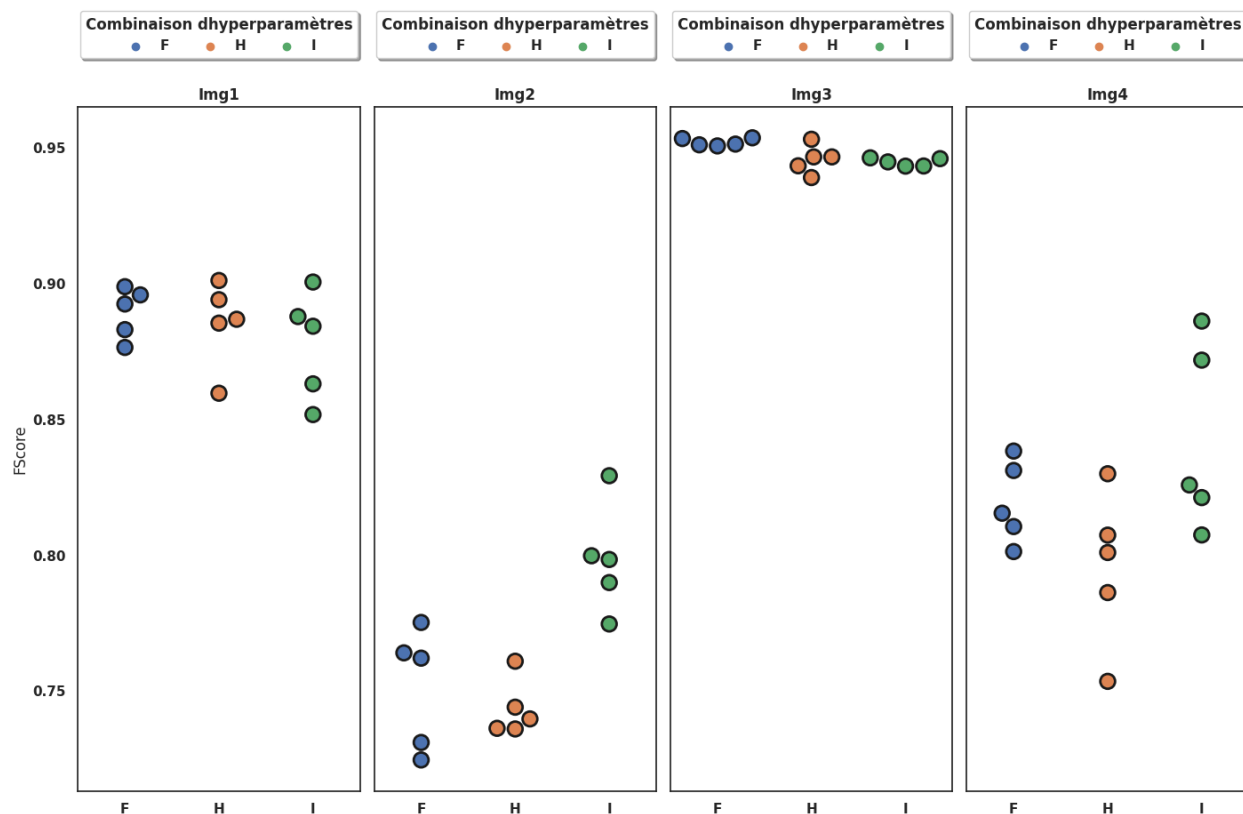


Figure 5.17 – Comparaison des performances des modèles issus de 5 séparations de données différentes pour chacun des groupes d’hyperparamètres des modèles les plus performants (F – Bleu, H – Orange et I – Vert) sur chacune des 4 images de tests (de gauche à droite : image de test #1, image de test #2, image de test #3 et image de test #4).

Les résultats issus de ces entraînements avec validation croisée permettent de constater une performance similaire des modèles ayant les hyperparamètres F et H sur les images de test, mais une performance supérieure pour le modèle ayant les hyperparamètres I; surtout pour les images 2 et 4. L’image de test #2 est particulièrement différente des autres, une bonne performance sur cette image est donc un indicatif que le modèle a une capacité de généralisation intéressante.

La Figure 5.18 compare (à gauche) les 3 catégories de modèles (abscisse) en fonction de leur *FScores* obtenus (pour chacune des 5 séparations de données (CV)) et (à droite), les performances de tous les modèles en fonction de la séparation des données (séparations en abscisse). Les performances (*FScores*) des modèles indiqués à la Figure 5.18 sont une moyenne sur les quatre images de test.

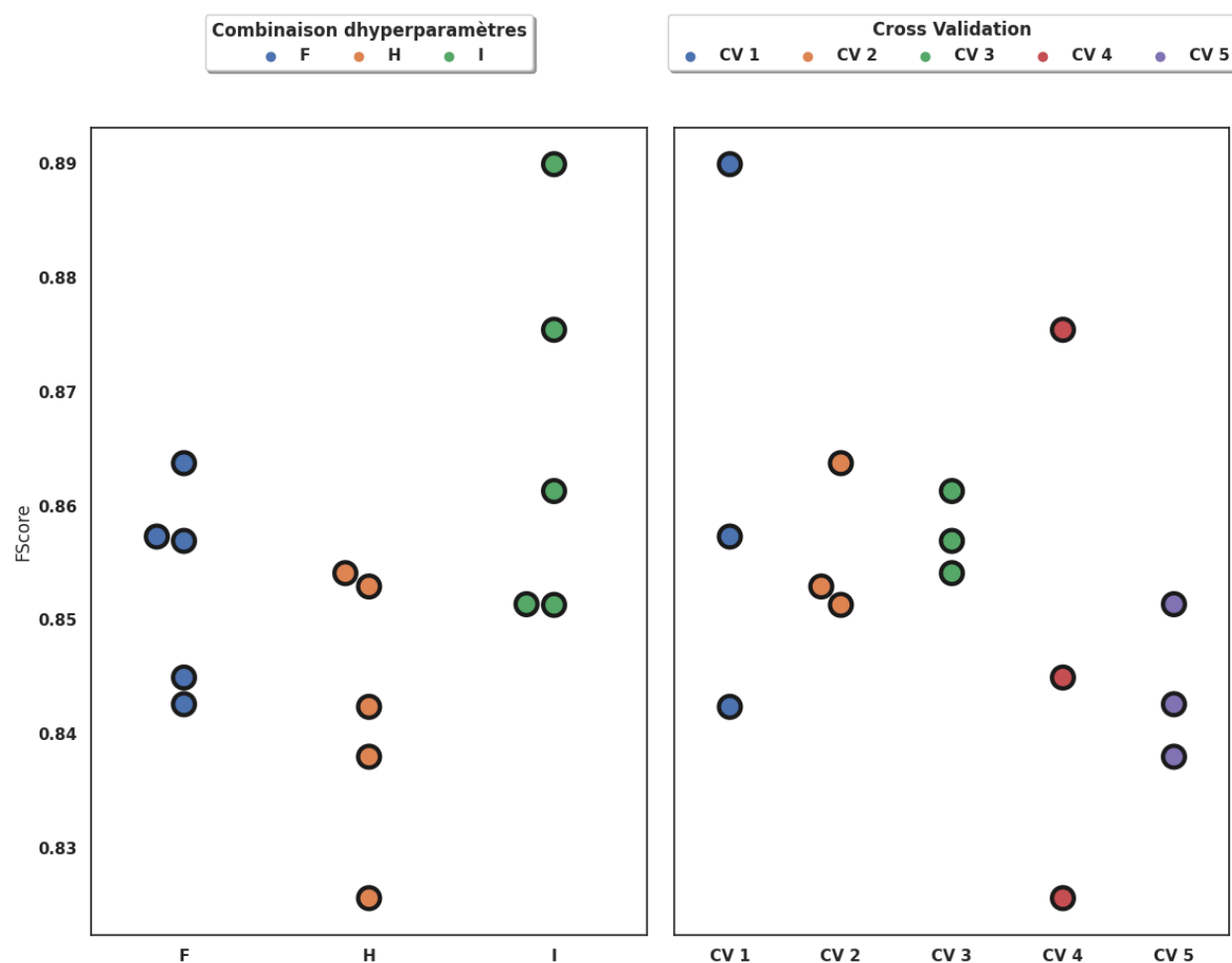


Figure 5.18 – Comparaison des modèles obtenus. À gauche, les 5 modèles (1 par séparation de donnée) entraînés avec les hyperparamètres F, H et I sont comparés. À droite, les modèles obtenus pour les 5 séparations de données sont comparés.

La Figure 5.18 - gauche permet de confirmer que lors de cet entraînement avec différentes séparations des données, les modèles de type *TransUNet* avec les hyperparamètres I ont mieux performés que les 2 autres catégories de modèles. La Figure 5.18 – graphique de droite permet de comparer l'influence de la séparation des données sur les performances des modèles. Cette différence est parfois très importante, comme il est possible de le voir entre la séparation 3 et la séparation 5. Pour la séparation 3, le pire modèle obtenu a tout de même mieux performé que le meilleur modèle issu de la séparation 5. Ceci rappelle l'importance des données d'entraînement sur la qualité finale du modèle. Les modèles ayant obtenu le meilleur *FScore* moyen (avant et après l'entraînement avec validation croisée) sur les quatre images sont présentés au Tableau 5.5.

Tableau 5.5 – Comparaison des *FScores* Moyens des meilleurs modèles

Modèles	Modèle F – <i>Attention UNet</i>	Modèle H – <i>Attention UNet</i>	Modèle I - <i>TransUNet</i>	Modèle I_CV1 - <i>TransUNet</i>
<i>FScore</i> Moyen	0,8968	0,8888	0,8880	0,8899

Le modèle le plus performant est donc le modèle F de l'*Attention UNet* entraîné avec la séparation de données initiale. Le modèle I avec la séparation de donnée #4 du *TransUNet* a bien performé avec un *FScore* moyen de 0,8899. Il est maintenant possible de comparer le modèle le plus performant aux meilleures performances de l'algorithme naïf *Find Maxima* au Tableau 5.6. Pour cette comparaison, nous allons d'abord comparer la précision, le *recall* et le *FScore* obtenus par : le modèle entraîné, l'algorithme *Find Maxima*, lorsqu'il est spécialement ajusté au meilleur seuil possible pour chaque image (travail manuel fastidieux) et à l'algorithme *Find Maxima* lorsqu'on maintient le seuil constant. Ensuite, au Tableau 5.8 les *Fscores* pour chaque image de ces trois méthodes seront indiqués.

Tableau 5.6 – Comparaison des performances moyennes entre le modèle choisi et *Find Maxima*

Modèles	Précision Moyenne	<i>Recall</i> Moyen	<i>FScore</i> Moyen
Modèle F – <i>Attention UNet</i>	0,957	0,846	0,897
<i>Find Maxima</i> - Meilleur seuil	0,915	0,859	0,885
<i>Find Maxima</i> – 1 seul seuil	0,911	0,846	0,873

Du Tableau 5.6, il est possible de constater que le modèle entraîné obtient une performance supérieure pour la précision et le *FScore* et une performance égale à *Find Maxima* avec 1 seul seuil pour le *recall*. Ceci nous indique que l'algorithme « intelligent » a une meilleure capacité ne pas se tromper lors de l'identification d'une NP, mais que l'algorithme a une performance similaire à *Find Maxima* (sur ces images de test) en ce qui a trait à l'identification du plus grand nombre de NP possible.

Tableau 5.7 – Comparaison du meilleur modèle à l’algorithme naïf *Find Maxima*

	<i>FScore</i> – Image #1	<i>FScore</i> – Image #2	<i>FScore</i> – Image #3	<i>FScore</i> – Image #4	<i>FScore</i> Moyen
modelF <i>Attention UNet</i>	0,924	0,844	0,940	0,879	89,68 %
Find Maxima Meilleur seuil pour chaque image	0,872	0,839	0,925	0,905	88,53%
Find Maxima un seul seuil pour toutes les images	0,840	0,838	0,917	0,895	87,26%

Le Tableau 5.7 permet de remarquer que le modèle d’apprentissage machine performe moins bien sur l’image de test #4. Ceci est vrai pour la plupart des modèles entraînés et est probablement dû au fait que l’image de test #4 est très difficilement représentée par les données d’entraînement. Il est possible au Tableau 5.8 de le constater par une augmentation de l’écart de performance entre l’algorithme naïf et l’algorithme entraîné lorsque les performances sur l’image de test #4 sont absentes.

Tableau 5.8 – Comparaison des meilleurs modèles sans l’image test #4

Modèles	Précision Moyenne	<i>Recall</i> Moyen	<i>FScore</i> Moyen
modelF – <i>Attention UNet</i>	0,962	0,853	90,27 %
<i>Find Maxima</i> - Meilleur seuil	0,926	0,837	87,87 %
<i>Find Maxima</i> – 1 seul seuil	0,922	0,821	86,66 %

Avec toutes les images de test, l’amélioration apportée sur le *FScore* par les modèles entraînés est de seulement 1,15% (différence de précision : +4,2 %, différence de *recall* : -1,3%), lorsque le

seuil de tolérance au bruit de *Find Maxima* est choisi à la perfection, puis de 2,42% (différence de précision : +4,6 %, différence de *recall* : 0%) si l'on prend un seuil optimisé pour ces images.

Sans l'image de test #4 qui semble mal représentée par les données d'entraînement, on obtient une amélioration de 2,4% (différence de précision : +3,6%, différence de *recall* : +1,6 %) par rapport aux seuils choisis parfaitement, puis de 3,61% (différence de précision : +4%, différence de *recall* : +3,2 %) par rapport au seuil constant de *Find Maxima*.

Ces différences ne sont pas majeures, mais démontrent tout de même que l'IA peut effectuer un meilleur travail de détection automatique de NPs qu'un algorithme naïf. L'approche utilisée démontre aussi une meilleure capacité d'adaptation des modèles « intelligents », puisque *Find Maxima* requiert un travail d'ajustement de seuil qui est effectué manuellement. Idéalement, avant d'être utilisé à plus grande échelle, ce modèle aurait besoin d'être entraîné sur de plus grandes bases de données, puisqu'il semble manquer de capacité de généralisation. En effet, l'image de test #4 semble avoir été très difficile à prédire pour le modèle par rapport à *Find Maxima* qui a mieux performé sur cette image.

Le temps de prédiction de l'algorithme « intelligent » est relativement rapide, mais dépend de l'ordinateur utilisé pour effectuer la prédiction. Les temps pour détecter les nanoparticules pour une image de 2048 x 2048 pixels sont montrés au Tableau 5.9. Ces temps sont obtenus en utilisant les ressources de Google Colab.

Tableau 5.9 – Temps de prédiction

Capacité de l'ordinateur	CPU seul - Colab	GPU A100 - Colab	GPU V100 - Colab	GPU T4 - Colab
Temps de prédiction	111 s	0,522 s	1,28 s	5 s

Les images de test utilisées contiennent des densités de NPs élevées, mais l'image supplémentaire (Image de test #5) présentée à la Figure 4.3 contient une densité de NPs très importante. Il est possible de comparer visuellement la performance du meilleur algorithme d'apprentissage machine à *Find Maxima* (seuil optimisé pour l'image précise). Cette comparaison est montrée aux Figure 5.19 et Figure 5.20.

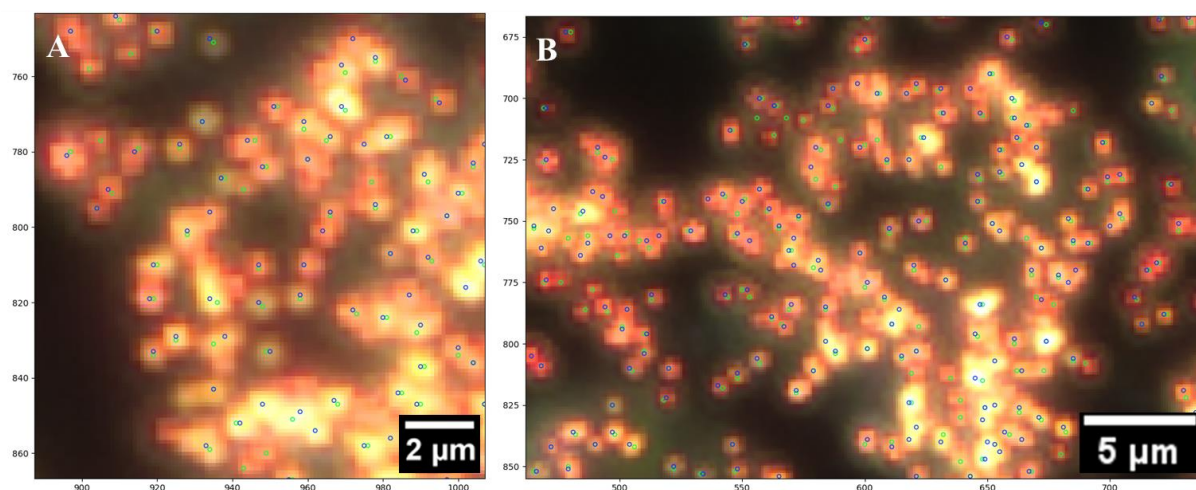


Figure 5.19 – Comparaison de la performance du modèle *Attention UNet F* et de *Find Maxima* pour une image à haute densité de NPs (image de test #5). Les détections par *Find Maxima* à un seuil de tolérance au bruit optimisé de valeur 7 sont en vert et les détections par le modèle entraîné sont en bleu. Ces comparaisons sont observables sur deux sections (A et B) de l'image de test #5.

De la Figure 5.19 et de l'agrandissement de la section A) présentée à la Figure 5.20, il est possible de remarquer une meilleure performance de l'algorithme « intelligent ». Cette différence de performance est quantifiée au Tableau 5.10.

Note : L'image de test #5 n'a pas été évaluée manuellement afin d'en extraire les positions exactes des NPs à partir de la microscopie 60x à huile. C'est pourquoi la quantification de ces sections de l'image #5 se base sur les détections des deux algorithmes uniquement. Cette quantification n'est qu'une démonstration rapide de la performance de l'algorithme « intelligent » pour des images à haute densité de NPs.

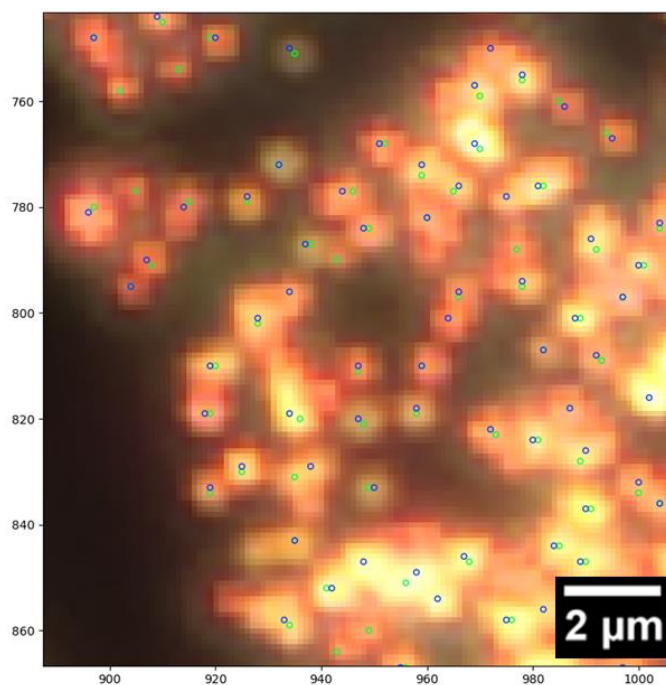


Figure 5.20 – Agrandissement de la section de l’image de test #5 A) de la Figure 5.19. Les cercles bleus représentent les détections effectuées par l’algorithme *Attention UNet – F*, tandis que les cercles verts représentent les détections effectuées par l’algorithme *Find Maxima* à un seuil de tolérance au bruit optimisé de 7.

Tableau 5.10 – Quantification des détections par *Find Maxima* et par le modèle entraîné *Attention UNet - F*

Images	Détections <i>Find maxima</i>	Détections Modèle	Pourcentage de détections réussies
Section #1 présentée à la Figure 5.19 A	<i>Find Maxima</i>	57	71,3 %
	Modèle	73	91,3 %
	Total de détections (pour les deux algorithmes)	80	100 %
Section #2 présentée à la Figure 5.19 B	<i>Find Maxima</i>	158	71,8 %
	Modèle	184	83,6 %
	Total de détections (pour les deux algorithmes)	220	100 %

Ces images permettent de constater la performance de l'algorithme entraîné par rapport à *Find Maxima* dans des échantillons à très hautes densités de NPs tel que montré à la Figure 5.20. Il est possible de voir que les NPs rapprochées sont difficiles à détecter pour *Find Maxima*, tandis que l'algorithme entraîné performe adéquatement. Ceci peut indiquer que les algorithmes «intelligents» ont une meilleure capacité de détection pour des images à haute densité de NPs et pourraient augmenter la plage dynamique de la méthode de marquage IP.

5.5 Résultats exploratoires

Les résultats des sections précédentes confirment qu'une détection automatisée des NPs est non seulement faisable, mais aussi souhaitable. Cette détection automatique permet une quantification des nanoparticules dans un échantillon.

Cependant, en s'inspirant de la méthode de détection de protéine actuelle (IHC) qui consiste à appliquer une coloration brunâtre, il est possible de représenter la présence des NPs sur un échantillon par une carte de densité. Cette carte de densité, une fois superposée à l'image permet une meilleure visualisation des régions à fortes densités de NPs et donc à forte expression de la protéine ciblée.

Ce marquage artificiel pourrait permettre aux pathologistes d'identifier les régions d'intérêts (ou de forte expression de la protéine) de manière plus intuitive. Cette approche est seulement exploratoire et n'a pas fait l'objet de validation par des pathologistes.

Cette coloration prend des teintes brunâtres de plus en plus foncées en fonction de la densité de NPs. Les images entrantes et sortantes de ce marquage artificiel sont présentées à la Figure 5.21.

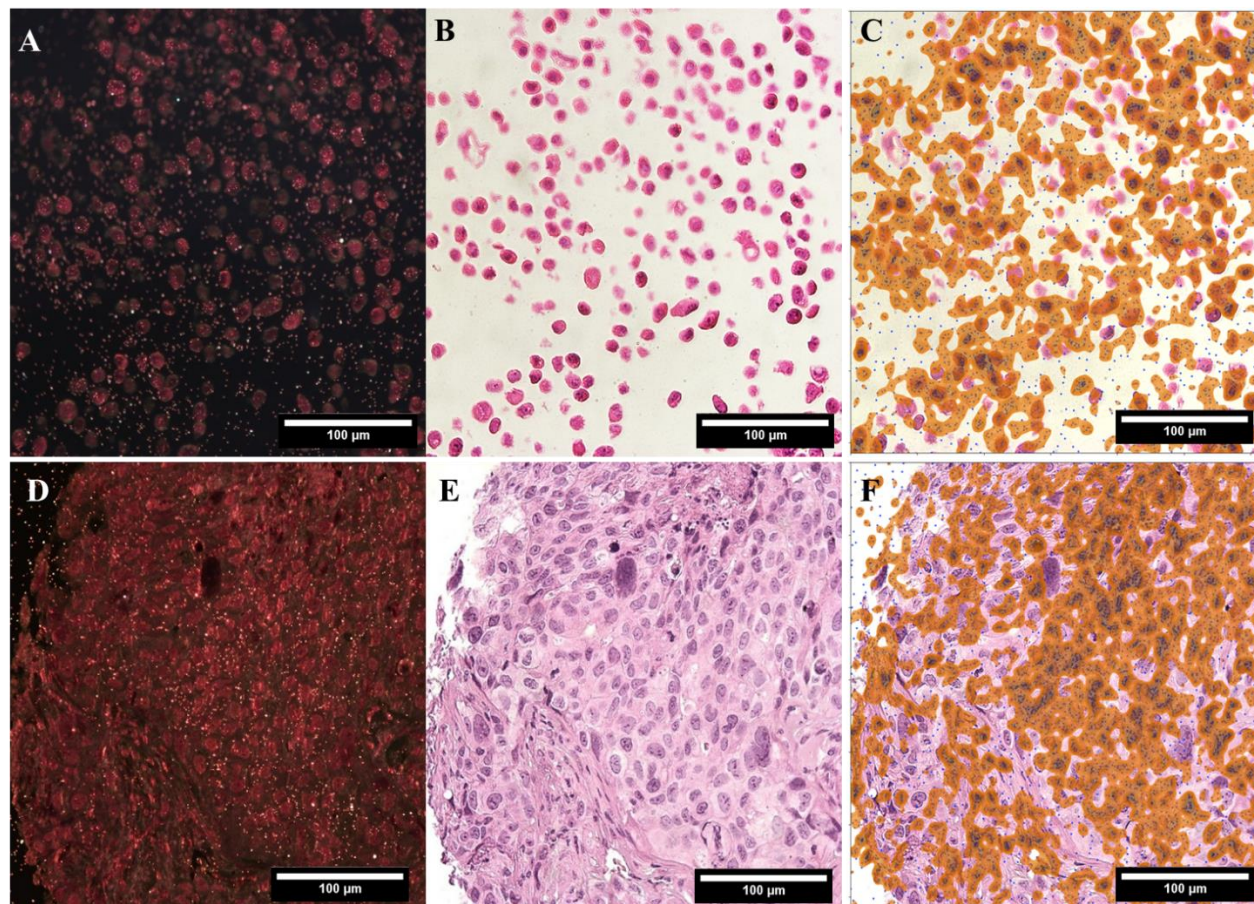


Figure 5.21 – Méthode de marquage numérique de l’immunoplasmonique. Échantillon cellulaire marqué par NPs - A) en Illumination latérale, B) en Champ clair et C) en marquage artificiel. Échantillon tissulaire (HER2 positif) - D) en Illumination latérale, E) en champ clair et F) en marquage artificiel

Le marquage consiste simplement à une détection automatique des NPs par le modèle F de l’architecture *Attention UNet* suivi de la génération d’une distribution KDE (*kernel density estimation*) qui est superposée à l’image en champ clair afin d’obtenir une visualisation intuitive de la distribution des NPs sur l’échantillon. L’image F) de la Figure 5.21 est optimisée de trois manières différentes à la Figure 5.22. En modifiant les différents paramètres de la fonction (*seaborn.kdeplot()*) issue de la librairie *Seaborn* permettant de générer des graphiques KDE superposés à des images, il est possible d’optimiser ce marquage pour obtenir l’apparence désirée.

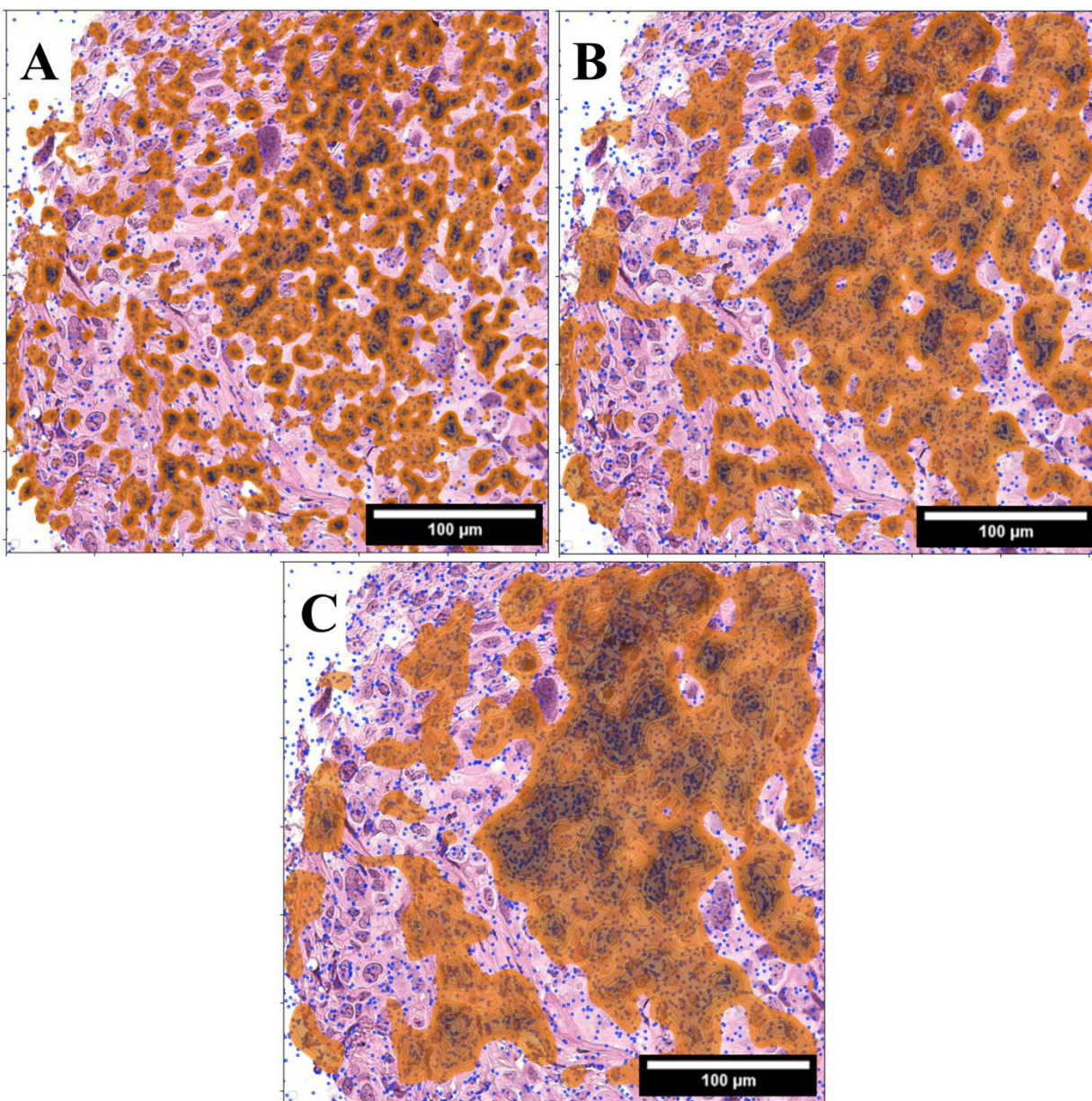


Figure 5.22 - Marquage artificiel d'un échantillon tissulaire après une détection automatique des NPs par l'algorithme « intelligent » *Attention UNet* - F. Cette figure présente 3 différentes possibilités de représentation de la densité des NPs détectées sur les images en champ clair. Ces représentations utilisent les mêmes paramètres de la fonction *kdeplot()* décrits à la section 4.6 de la Méthodologie, en faisant varier la valeur du paramètre de bande passante, tel que :

A) $Bw = 0,12$ B) $Bw = 0,22$ et C) $Bw = 0,32$.

Cette méthode de marquage est seulement présentée à titre exploratoire, et propose une nouvelle méthode de marquage artificielle des échantillons en champs clair après une détection des NPs par un algorithme « intelligent ».

CHAPITRE 6 DISCUSSION

6.1 Discussion sur les hyperparamètres des modèles choisis

Les différents résultats permettent la comparaison de plusieurs modèles en fonction de leurs différents hyperparamètres, notamment, l'architecture, la structure d'encodage (*backbone*), la taille de parcelle (*patch*), la taille de lot (*batch*) , la fonction d'erreur (*Loss*) et l'optimiseur. Tous ces entraînements ont permis d'obtenir des modèles qui ont la capacité de détecter des NPs sur des cellules. Cependant, la capacité de généralisation des modèles sur différentes images varie et fait en sorte que certains modèles sont tout simplement plus performants que d'autres. Les distributions des *FScores* obtenus par différents hyperparamètres permettent de comparer l'influence de ces derniers sur la performance globale des modèles résultants.

6.1.1 Effet de l'architecture

Les architectures choisies ont eu un impact sur la performance générale des modèles. En effet, il semblerait que pour une banque de données de petite taille, comme celle-ci, *l'Attention UNet* soit particulièrement efficace. L'architecture *Attention UNet* a produit les 2 meilleurs modèles avant validation croisée (*Attention UNet F* et *H*), tout en produisant en moyenne de meilleurs modèles que les 2 autres types d'architectures (*FScore* moyen de 0,875 contre 0,865 pour *TransUNet* et 0,867 pour *UNet*). Cette architecture utilise des portes d'attentions (AG) qui permettent au modèle une reconstruction plus optimale des images afin d'effectuer une prédiction. Il est possible que ce mécanisme ait eu un impact plus important parce que les objets analysés sont particulièrement petits. En effet, *l'Attention UNet* a déjà été reconnu comme étant plus performant qu'un *UNet* lors de la segmentation sémantique de petits organes en imagerie CT (*computed tomography*) [56].

Pour sa part, le *TransUNet* a permis d'obtenir de bons résultats (*FScore* de 0,888) et d'autres moins bons (*FScore* de 0,840). Ce modèle dispose d'un mécanisme d'attention globale grâce au module de *Transformer* présent à sa couche la plus profonde. Il est possible que ce mécanisme ait augmenté la performance des modèles dans certains cas, puis ait diminué cette performance dans d'autres cas. Il est possible que la taille de la banque de données ait limité de manière plus importante le *TransUNet* que les deux autres architectures, rendant ainsi les performances de ces modèles plus variables. En effet, l'entraînement d'encodeurs d'images de type *Transformer* est plus demandant en ressources de calcul et requiert plus de données d'entraînement qu'un modèle convolutif [64].

Ceci semble suggérer que les *TransUNet* doivent être employés avec prudence lors de l'entraînement sur de petites banques de données.

Finalement, le *UNet* semble quant à lui avoir performé de manière équivalente en moyenne que le *TransUNet* (*FScore* moyen de 0,867 contre 0,865), mais n'a pas permis d'obtenir d'excellents modèles (*Fscore* moyen maximal de 0,881). Ceci suggère que l'ajout d'un mécanisme d'attention global ou local permet d'augmenter la performance d'un modèle de segmentation sémantique lors de la détection de NPs.

6.1.2 Effet de la structure d'encodage (*backbone*)

Pour ce qui est des structures d'encodages (*backbones*), *VGG16* et *VGG19* ont produit des modèles comparables (*FScores* moyens de 0,865 contre 0,861 respectivement), tandis que *ResNet50* permet d'obtenir de meilleurs résultats en moyenne (*Fscore* moyen de 0,882). Cependant, les trois meilleures combinaisons d'hyperparamètres impliquent soit *VGG19* (F & H) ou *ResNet50* (I). Ceci semble indiquer que les modèles comportant plus de couches convolutives ont été plus performants. Cette augmentation de performance est probablement liée au fait que des modèles comportant plus de couches convolutives permettent d'extraire des informations plus subtiles des images et d'extraire plus de relations entre les valeurs de pixels. Toutefois, une augmentation de la profondeur rend le modèle plus à risque de *l'overfitting*, puisqu'il est plus facile pour le modèle de s'ajuster parfaitement aux données d'entraînements (et donc de moins bien performer sur les données de validation et de test).

6.1.3 Effet de la taille de parcelles (*Patch*)

La taille de parcelle semble aussi avoir eu un impact sur la performance du modèle, puisque les quelques modèles entraînés avec une taille de parcelle de 128 x 128 pixels ont obtenus une moyenne de performance supérieure aux autres modèles (*Fscore* moyen de 0,880 contre 0,868 pour les tailles de parcelle de (256 x 256)). Ceci est potentiellement lié au fait que la tâche implique la détection d'objets de quelques pixels seulement. En effet, lorsque l'on diminue la taille des parcelles, les relations entre les pixels très rapprochés sont probablement plus représentées, permettant une meilleure identification des NPs, mais le modèle perd de l'information globale de l'image qu'il aurait eue avec des tailles de parcelles plus grandes (par exemple, les effets dus à un angle de la lame de microscopie ou la présence de cellules) [65].

6.1.4 Effet de la taille de lot (*Batch*)

La comparaison des modèles produits par différentes tailles de lots a permis d'obtenir des résultats comparables, quoique légèrement supérieurs pour les tailles de lot de 5 (*FScores* moyens de 0,867 contre 0,863 pour les tailles de lot de 3). Cette amélioration pour la taille de lot de 5 est surprenante, puisque l'augmentation de la taille de lot est considérée comme un facteur diminuant la performance des modèles [66]. Il est possible que l'augmentation de la taille de lot ait eu un effet régularisateur sur les données qui a limité l'*overfitting* des modèles. La différence entre les modèles entraînés avec une taille de lot de 5 vs 3 est très faible, et le meilleur modèle (*Attention UNet F*) utilisait une taille de lot de 3. On peut donc dire que l'impact de cet hyperparamètre n'était pas un facteur majeur dans l'entraînement de ces modèles.

6.1.5 Effet de la fonction d'erreur (*Loss*) et de l'optimiseur

Les paramètres d'entraînement semblent avoir eu un léger impact sur la performance. En effet, les modèles entraînés avec comme fonction d'erreur *IOU* ont obtenu des performances légèrement supérieures aux modèles entraînés avec la fonction d'erreur *Dice* (0,871 vs 0,861), alors que l'impact des optimiseurs était négligeable (0,866 vs 0,867). La différence entre *IOU* et *Dice* est surprenante, puisque les données que nous tentons d'évaluer sont des données très déséquilibrées (proportion de pixels associés aux NPs très faible par rapport à tous les pixels de l'image) et normalement, *Dice* est une fonction d'erreur de choix pour ce type de données. Cependant, les deux approches permettent d'obtenir des résultats satisfaisants, puisque la combinaison de paramètres *F* (qui contient la fonction *IOU*) et la combinaison de paramètres *I* (qui contient la fonction *Dice*) ont permis d'obtenir les deux meilleurs modèles (*Attention UNet F*: 0,8968 contre *TransUNet CV1_I* : 0,8899).

6.2 Viabilité d'un modèle d'apprentissage machine pour la détection de NPs

Les résultats permettent de considérer la détection de nanoparticules automatisée comme une option viable. Cependant, les différences de performance moyennes entre les algorithmes de détection automatiques (89,68 %) et les algorithmes de détection naïf (87,26 %) soulevées dans le cadre de ce mémoire ne démontrent pas une augmentation de performance suffisante des algorithmes automatisés pour une utilisation industrielle ou clinique. En effet, la variabilité dans la performance (évaluation avec ou sans l'image #4) est un bon indicatif du fait que ces algorithmes

ont besoin d'être exposés à de plus vastes banques de données avant d'obtenir des performances plus régulières. Une augmentation de la banque de données ferait en sorte que ces algorithmes seraient généralisables à la détection de NPs dans des contextes d'application présentant certaines variations [67], [68]. Pour de petites banques de données, les performances augmentent rapidement lors d'une augmentation de la banque de données. Le fait de doubler la banque de données pourrait permettre d'augmenter de façon considérable la performance de l'algorithme.

Malgré cela, il faut tout de même souligner deux choses. Premièrement, les modèles de détection automatisés ont permis d'obtenir des résultats qui sont tout de même supérieurs à l'algorithme naïf. De plus, cette performance augmente à des valeurs qui peuvent réellement faire une différence dans le décompte des nanoparticules lors de l'évaluation sans l'image #4 (90,27% pour le modèle « intelligent » et 86,66% pour un seuil optimisé de *Find Maxima*).

Il est aussi particulièrement pertinent de s'attarder au fait que le modèle automatisé semble effectuer de bien meilleures détections que l'algorithme naïf sur l'image à haute densité de NPs (91,3% de détections pour l'algorithme « intelligent » contre 71,3% de détections pour *Find Maxima* sur la section A de la Figure 5.19 et 83,6% contre 71,8% pour la section B de la Figure 5.19). Cette performance sur des images à hautes densités de NPs de l'algorithme « intelligent » permettra potentiellement d'augmenter la plage dynamique de la méthode IP et, par le même fait, d'augmenter sa viabilité comme une alternative clinique de choix.

Deuxièmement, l'utilisation d'algorithmes naïfs requiert à l'utilisateur d'établir et de modifier un seuil de détection. Dans le contexte d'une application clinique, l'optimisation d'un seuil de détection pour chaque image serait inapplicable dans un contexte de manque de temps pour les pathologistes. Même si l'acquisition était soumise à une quelconque calibration, les intensités associées à chaque NP varient grandement en fonction de leur position dans l'échantillon, de leur concentration et du tissu sur lequel elle se trouve. Cette performance au moins équivalente d'un algorithme « intelligent » ouvre la possibilité d'une détection de NPs qui ne requiert pas d'ajustements manuels lors de la détection. Cependant, avant d'atteindre ce stade, les algorithmes « intelligents » devront être exposés à une plus grande quantité de données.

6.3 Limites de l'étude

Cette étude de l'utilisation de modèles d'IA pour la détection de NPs comporte plusieurs limites. Tout d'abord, le volume de données restreint a limité l'entraînement ainsi que l'évaluation des

modèles. En effet, l'utilisation de seulement 4 images de 1024 x 1024 pixels (quoiqu'elles contiennent des centaines, voir des milliers de NPs) est largement insuffisante pour effectuer des conclusions robustes sur la performance des modèles. Comme il a été suggéré à la section précédente de doubler la banque de données, une quantité correspondante d'images de test devrait représenter 15 à 20 % de la base de données. On peut donc viser une dizaine d'images de test pour avoir une évaluation plus informative.

De plus, le faible nombre de données d'entraînement a aussi limité la capacité des modèles à obtenir de meilleures performances lors de la prédiction. Le fait d'avoir utilisé seulement des images provenant de deux CMAs a probablement limité l'exposition du modèle à des concentrations plus faibles ou plus élevées de NPs que ce qu'il y avait dans ces deux CMAs. Heureusement, le manque de données a en partie été compensé par l'utilisation de techniques d'augmentation de données et par l'utilisation de poids préentraînés sur la vaste banque de données *Imagenet* [42]. De cette façon, les modèles ont appris beaucoup plus rapidement et avec beaucoup moins de données que s'ils n'avaient jamais subi ce préentraînement.

Une autre limitation est le fait d'avoir été restreint dans l'évaluation de différentes architectures de modèles ou différents hyperparamètres par le temps d'entraînement de chacun des modèles. Il aurait été intéressant par exemple d'évaluer la performance d'un modèle entièrement *Transformer* tel que le ViT [59] ou d'évaluer un modèle hybride comme le *TransUNet ++* [64] qui combine l'utilisation de portes d'attention (AG) de l'*Attention UNet* à l'utilisation de mécanismes d'attention globale par *Transformer* du *TransUNet*.

De plus, le taux d'apprentissage et le point d'arrêt lors de l'entraînement des modèles peuvent avoir un impact important sur les modèles résultants, mais il n'ont pas été modifiés afin de pouvoir comparer les résultats adéquatement. Les hyperparamètres évalués auraient aussi pu inclure plus de variabilité afin d'obtenir des conclusions plus approfondies. Par exemple, il aurait été intéressant d'avoir une comparaison des tailles parcelles entre 16 x 16 et 512 x 512 pixels ou d'avoir une comparaison de l'impact des tailles de batch entre 1 et 16. Ensuite, certains hyperparamètres n'ont pas été comparés dans le cadre de ce mémoire, tels que la fonction d'activation qui a été fixée à *ReLU* ou le nombre de neurones impliqués dans le module *Transformer* du *TransUNet* qui a été fixé à 1536.

Ensuite, une limitation majeure selon moi est la conception des masques de vérités automatiques. En effet, lors de la conception, les images haute résolution sont soumises à une détection par *Find Maxima*. Toutefois, tel que discuté dans ce mémoire, cette méthode n'est pas parfaite, et a induit des erreurs dans les masques de vérités automatiques. Ces erreurs font en sorte que les masques sur lesquels sont entraînés les modèles ne sont pas parfaits. Le fait d'entraîner les modèles avec des masques de vérités imparfaits a probablement induit un taux d'erreur presque inévitable dans les modèles de prédiction. Ce masque de vérité automatique fait aussi en sorte qu'il est difficile d'entraîner les modèles sur des images à très haute densité de NPs, puisque la qualité du masque de vérité risque de diminuer et d'engendrer un moins bon modèle par la suite.

Finalement, lorsque l'on se concentre sur l'application clinique de la méthode, soit la prédiction de la réponse au traitement des patients en fonction de la quantification de différentes protéines, il est important de rappeler que les NPs ne peuvent pas identifier l'intégralité des protéines de l'échantillon. En effet, les NPs sont de près de 100nm, ce qui occasionne une gêne stérique entre les NPs trop rapprochées et qui rend impossible l'identification de toutes les protéines. La méthode immunoplasmonique propose donc une quantification relative de l'expression d'une protéine en évaluant l'attachement des NPs sur l'échantillon. Il serait particulièrement pertinent d'effectuer des études rétrospectives lors desquelles nous comparerions la réponse au traitement de différents patients en fonction du marquage par IP et du marquage par IHC de leur biopsies. Si une corrélation est présente, il sera possible d'établir des seuils de NPs selon lesquels différents patients seraient éligibles aux différents traitements. Entre temps, la méthode pourrait tout de même servir d'information supplémentaire pour les pathologistes en combinant le marquage IHC au marquage par IP. Dans tous les cas, la méthode IP permettrait une alternative objective à l'évaluation de l'expression d'une protéine. Même si le résultat final de la prédiction de la réponse au traitement serait potentiellement similaire en IHC et en IP (classé des patients selon la présence du marqueur), l'IP permettrait d'avoir une mesure quantitative et ainsi, hypothétiquement, de réduire les erreurs associées à la classification des patients entre différentes catégories d'expression de la protéine d'intérêt.

CHAPITRE 7 CONCLUSION ET RECOMMANDATIONS

En terminant, les modèles « intelligents » offrent une solution souhaitable pour la détection de NPs sur des échantillons H&E FFPE. En effet, le modèle résultant de cette recherche performe mieux que les méthodes utilisées précédemment (0,8968 contre 0,8726) et permet de s'affranchir des complexités d'utilisation des méthodes naïves qui requièrent un ajustement manuel à chaque utilisation. Les modèles de détection automatiques de NPs ont permis d'obtenir des résultats prometteurs sur des images à haute densité de NPs, alors que les algorithmes naïfs perdent énormément en performance dans ce type de contexte. Cette approche permet donc potentiellement d'augmenter la limite de détection des NPs et d'augmenter la plage dynamique de la méthode IP.

En résumé, l'utilisation d'un algorithme « intelligent » est souhaitable pour les trois raisons suivantes :

- 1- L'algorithme « intelligent » obtient des performances supérieures aux algorithmes naïfs
- 2- L'algorithme « intelligent » ne requiert pas d'ajustement manuel entre chaque acquisition (cet aspect serait amélioré si la banque de données était plus importante)
- 3- L'algorithme « intelligent » semble mieux performer que les algorithmes naïfs sur des images à haute densité de NPs.

Cependant, la taille de la banque de données utilisée pour effectuer l'entraînement des modèles n'est pas suffisamment volumineuse pour permettre l'obtention d'un modèle final. De plus, l'entraînement fut effectué sur des cultures cellulaires colorées par H&E et fixées en blocs de paraffine, mais un modèle final applicable en clinique devrait être entraîné sur des images de tissus marqués par NPs.

7.1 Opportunités futures et applications

L'utilisation de l'IA pour l'évaluation d'échantillons marqués par IP en est à ses débuts. En effet, la détection de NPs effectuée dans ce mémoire comporte uniquement des NPs d'or de 100nm de diamètre. Cependant, l'un des avantages majeurs de la technologie IP est la possibilité de cibler différentes protéines à la fois sur un même échantillon par l'utilisation de différentes couleurs de NPs. Il serait donc possible de détecter en une étape les NPs et leur couleur par IA. Cette méthode passerait par l'entraînement de modèles à effectuer de la segmentation sémantique multi-classe (une classe pour chaque couleur de NP), plutôt que de la classification binaire telle qu'effectuée

dans ce mémoire. Ceci permettrait d'effectuer la quantification de l'IP pour différents marqueurs simultanément.

Il serait aussi possible d'améliorer la représentation et la quantification des NPs afin de se rapprocher des standards de quantification cliniques. En effet, tel que mentionné à la section 2.1.3 de la Revue de littérature, lors de la quantification de l'expression de protéines par IHC, les pathologistes doivent évaluer le nombre de cellules exprimant faiblement ou fortement la protéine d'intérêt. Afin d'obtenir une mesure similaire dans un contexte de marquage par IP, il serait alors pertinent de colocaliser des cellules segmentées sur des tissus en champ clair avec les détections de NPs obtenues par la détection automatisée. Ceci permettrait une évaluation d'échantillons qui serait comparable à la méthode clinique.

Les solutions informatiques futures de la méthode IP inspirées de ces travaux incluraient donc une détection automatisée de différentes couleurs de NPs, une détection automatisée des cellules en utilisant des méthodes documentées dans la littérature [69], puis une colocalisation des NPs sur les cellules afin d'établir les niveaux d'expressions protéiques des différentes cellules composant l'échantillon. De plus, afin de faciliter l'identification de zones de forte expression et de faciliter la visualisation du marquage IP par les pathologistes, il serait intéressant d'utiliser des représentations de densité de NPs sur les images tel que présenté à la section 5.5 des résultats.

En terminant, la prédiction de la réponse au traitement des patients atteints de cancer du sein est une étape cruciale qui peut affecter drastiquement le futur des patients. La méthode IP permet de combler des lacunes de quantifications de cette étape. Les outils logiciels basés sur l'IA rattachés à cette technologie ont le potentiel d'augmenter la plage dynamique de la méthode, tout en facilitant l'intégration de la technologie en clinique. L'augmentation du potentiel de la méthode IP permettra ultimement une meilleure stratification des patients afin de limiter les erreurs cliniques associées à la prédiction de la réponse aux immunothérapies contre le cancer du sein.

RÉFÉRENCES

- [1] Canadian Cancer Statistics Advisory Committee in collaboration with the Canadian Cancer Society, Statistics Canada and the Public Health Agency of Canada. Canadian Cancer Statistics 2023. Toronto, ON: Canadian Cancer Society; 2023.
- [2] M. Cianfrocca et L. J. Goldstein, « Prognostic and Predictive Factors in Early-Stage Breast Cancer », *The Oncologist*, vol. 9, n° 6, p. 606-616, nov. 2004, doi: 10.1634/theoncologist.9-6-606.
- [3] A. I. Fernandez *et al.*, « Examination of Low ERBB2 Protein Expression in Breast Cancer Tissue », *JAMA Oncol.*, vol. 8, n° 4, p. 607, avr. 2022, doi: 10.1001/jamaoncol.2021.7239.
- [4] L. Wang, C. Darviot, J. Zapata-Farfan, S. Patskovsky, D. Trudel, et M. Meunier, « Designable nanoplasmonic biomarkers for direct microscopy cytopathology diagnostics », *J. Biophotonics*, vol. 12, n° 11, p. e201900166, nov. 2019, doi: 10.1002/jbio.201900166.
- [5] V. Debien *et al.*, « Immunotherapy in breast cancer: an overview of current strategies and perspectives », *Npj Breast Cancer*, vol. 9, n° 1, p. 7, févr. 2023, doi: 10.1038/s41523-023-00508-3.
- [6] J. Owen, J. Punt, S. A. Stranford, et J. Kuby, *Kuby immunology*, 7. ed., International ed. Basingstoke: Macmillan Higher Education, 2013.
- [7] R. Sabatier *et al.*, « Prognostic and predictive value of PDL1 expression in breast cancer », *Oncotarget*, vol. 6, n° 7, p. 5449-5464, mars 2015, doi: 10.18632/oncotarget.3216.
- [8] E. S. McDonald, A. S. Clark, J. Tchou, P. Zhang, et G. M. Freedman, « Clinical Diagnosis and Management of Breast Cancer », *J. Nucl. Med.*, vol. 57, n° Supplement 1, p. 9S-16S, févr. 2016, doi: 10.2967/jnumed.115.157834.
- [9] M. Drăgănescu et C. Carmocan, « Hormone Therapy in Breast Cancer », *Chirurgia (Bucur.)*, vol. 112, n° 4, p. 413, 2017, doi: 10.21614/chirurgia.112.4.413.
- [10] on behalf of the Intensive Care in Hematological and Oncological Patients (iCHOP) Collaborative Group *et al.*, « New drugs, new toxicities: severe side effects of modern targeted and immunotherapy of cancer and their management », *Crit. Care*, vol. 21, n° 1, p. 89, déc. 2017, doi: 10.1186/s13054-017-1678-1.
- [11] C. for D. E. and Research, “FDA approves fam-trastuzumab deruxtecan-nxki for HER2-low breast cancer,” FDA, Aug. 2022, Accessed: Dec. 07, 2023. [Online]. Available: <https://www.fda.gov/drugs/resources-information-approved-drugs/fda-approves-fam-trastuzumab-deruxtecan-nxki-her2-low-breast-cancer>.
- [12] A. Indini, E. Rijavec, et F. Grossi, « Trastuzumab Deruxtecan: Changing the Destiny of HER2 Expressing Solid Tumors », *Int. J. Mol. Sci.*, vol. 22, n° 9, p. 4774, avr. 2021, doi: 10.3390/ijms22094774.
- [13] Y. Chen, L. Liu, R. Ni, et W. Zhou, « Advances in HER2 testing », in *Advances in Clinical Chemistry*, vol. 91, Elsevier, 2019, p. 123-162. doi: 10.1016/bs.acc.2019.03.004.
- [14] IHC world, « Standard Protocol for Formalin-Fixed Paraffin Embedded Tissue ». St-Micheal's Inspired Care, Inspiring science. [En ligne]. Disponible sur:

<https://research.unityhealth.to/wp-content/uploads/2017/05/Standard-Protocol-for-Formalin%E2%80%90Fixed-Paraffin-Embedded-Tissue.pdf>

- [15] A. C. Wolff *et al.*, « Human Epidermal Growth Factor Receptor 2 Testing in Breast Cancer: American Society of Clinical Oncology/College of American Pathologists Clinical Practice Guideline Focused Update », *Arch. Pathol. Lab. Med.*, vol. 142, n° 11, p. 1364-1382, nov. 2018, doi: 10.5858/arpa.2018-0902-SA.
- [16] D. G. Hicks et R. R. Tubbs, « Assessment of the HER2 status in breast cancer by fluorescence in situ hybridization: a technical review with interpretive guidelines », *Hum. Pathol.*, vol. 36, n° 3, p. 250-261, mars 2005, doi: 10.1016/j.humpath.2004.11.010.
- [17] R. D. Powell *et al.*, « Metallographic in situ hybridization », *Hum. Pathol.*, vol. 38, n° 8, p. 1145-1159, août 2007, doi: 10.1016/j.humpath.2007.05.004.
- [18] Foundation Medicine, Inc, « FoundationOne CDx™ Technical Information ». 30 novembre 2017. [En ligne]. Disponible sur: https://www.accessdata.fda.gov/cdrh_docs/pdf17/P170019C.pdf
- [19] X. Michalet *et al.*, « Quantum Dots for Live Cells, in Vivo Imaging, and Diagnostics », *Science*, vol. 307, n° 5709, p. 538-544, janv. 2005, doi: 10.1126/science.1104274.
- [20] Balogh LM, Kimoto E, Chupka J, Zhang H, Lai Y (2012) Membrane Protein Quantification by Peptide-Based Mass Spectrometry Approaches: Studies on the Organic Anion-Transporting Polypeptide Family. *J Proteomics Bioinform* 6:229-236.
- [21] P. A. Kaufman *et al.*, « Assessing the discordance rate between local and central HER2 testing in women with locally determined HER2-negative breast cancer », *Cancer*, vol. 120, n° 17, p. 2657-2664, sept. 2014, doi: 10.1002/cncr.28710.
- [22] P. Marcoux-Valiquette *et al.*, « Multiplexed Plasmonic Nano-Labeling for Bioimaging of Cytological Stained Samples », *Cancers*, vol. 13, n° 14, p. 3509, juill. 2021, doi: 10.3390/cancers13143509.
- [23] L. Wang, M. Hasanzadeh Kafshgari, et M. Meunier, « Optical Properties and Applications of Plasmonic-Metal Nanoparticles », *Adv. Funct. Mater.*, vol. 30, n° 51, p. 2005400, déc. 2020, doi: 10.1002/adfm.202005400.
- [24] J. Olson, S. Dominguez-Medina, A. Hoggard, L.-Y. Wang, W.-S. Chang, et S. Link, « Optical characterization of single plasmonic nanoparticles », *Chem. Soc. Rev.*, vol. 44, n° 1, p. 40-57, 2015, doi: 10.1039/C4CS00131A.
- [25] LP2L, Numerical tools – NFMie program, <http://lp2l.polymtl.ca/en/outils> (accessed: November 2023).
- [26] D. Rioux, S. Vallières, S. Besner, P. Muñoz, E. Mazur, et M. Meunier, « An Analytic Model for the Dielectric Function of Au, Ag, and their Alloys », *Adv. Opt. Mater.*, vol. 2, n° 2, p. 176-182, févr. 2014, doi: 10.1002/adom.201300457.
- [27] R. naturelles Canada, “Le spectre électromagnétique.” Accessed: Dec. 07, 2023. [Online]. Available: <https://ressources-naturelles.canada.ca/cartes-outils-publications/imagerie-satellitaire-photos-aer/tutoriels-sur-la-teledetection/introduction/le-spectre-electromagnetique/14624> .

- [28] R. Herizchi, E. Abbasi, M. Milani, et A. Akbarzadeh, « Current methods for synthesis of gold nanoparticles », *Artif. Cells Nanomedicine Biotechnol.*, vol. 44, n° 2, p. 596-602, févr. 2016, doi: 10.3109/21691401.2014.971807.
- [29] J. Turkevich, P. C. Stevenson, et J. Hillier, « A study of the nucleation and growth processes in the synthesis of colloidal gold », *Discuss. Faraday Soc.*, vol. 11, p. 55, 1951, doi: 10.1039/df9511100055.
- [30] FRENS, G. Controlled Nucleation for the Regulation of the Particle Size in Monodisperse Gold Suspensions. *Nature Physical Science* 241, 20–22 (1973). <https://doi.org/10.1038/physci241020a0>.
- [31] D. Rioux et M. Meunier, « Seeded Growth Synthesis of Composition and Size-Controlled Gold–Silver Alloy Nanoparticles », *J. Phys. Chem. C*, vol. 119, n° 23, p. 13160-13168, juin 2015, doi: 10.1021/acs.jpcc.5b02728.
- [32] A. C. Marques, P. J. Costa, S. Velho, et M. H. Amaral, « Functionalizing nanoparticles with cancer-targeting antibodies: A comparison of strategies », *J. Controlled Release*, vol. 320, p. 180-200, avr. 2020, doi: 10.1016/j.jconrel.2020.01.035.
- [33] S. Patskovsky, E. Bergeron, et M. Meunier, « Hyperspectral darkfield microscopy of PEGylated gold nanoparticles targeting CD44-expressing cancer cells », *J. Biophotonics*, vol. 8, n° 1-2, p. 162-167, janv. 2015, doi: 10.1002/jbio.201300165.
- [34] M. Qi, C. Darviot, S. Patskovsky, et M. Meunier, « Cost-effective side-illumination darkfield nanoplasmonic marker microscopy », *The Analyst*, vol. 144, n° 4, p. 1303-1308, 2019, doi: 10.1039/C8AN01891J.
- [35] Dwaithe, « Maxima finding algorithm recreated from implementation in ImageJ/Fiji ». 26 mai 2021. Consulté le: 7 décembre 2023. [En ligne]. Disponible sur: <https://github.com/dwaithe/MaximaFinder.git>
- [36] « Maximum Finder ». in Image J. Fiji. [En ligne]. Disponible sur: <https://imagej.net/ij/developer/source/ij/plugin/filter/MaximumFinder.java.html>
- [37] J. Schindelin *et al.*, « Fiji: an open-source platform for biological-image analysis », *Nat. Methods*, vol. 9, n° 7, p. 676-682, juill. 2012, doi: 10.1038/nmeth.2019.
- [38] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, et G. E. Dahl, « On Empirical Comparisons of Optimizers for Deep Learning ». arXiv, 15 juin 2020. Consulté le: 7 décembre 2023. [En ligne]. Disponible sur: <http://arxiv.org/abs/1910.05446>
- [39] Muraina, Ismail. “Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts.” 7th International Mardin Artuklu Scientific Research Conference. 2022.,
- [40] Y. LeCun, Y. Bengio, et G. Hinton, « Deep learning », *Nature*, vol. 521, n° 7553, p. 436-444, mai 2015, doi: 10.1038/nature14539.
- [41] Y. Lecun, L. Bottou, Y. Bengio, et P. Haffner, « Gradient-based learning applied to document recognition », *Proc. IEEE*, vol. 86, n° 11, p. 2278-2324, nov. 1998, doi: 10.1109/5.726791.
- [42] O. Russakovsky *et al.*, « ImageNet Large Scale Visual Recognition Challenge », *Int. J. Comput. Vis.*, vol. 115, n° 3, p. 211-252, déc. 2015, doi: 10.1007/s11263-015-0816-y.

- [43] M. D. Zeiler et R. Fergus, « Visualizing and Understanding Convolutional Networks », in *Computer Vision – ECCV 2014*, vol. 8689, D. Fleet, T. Pajdla, B. Schiele, et T. Tuytelaars, Éd., in *Lecture Notes in Computer Science*, vol. 8689. , Cham: Springer International Publishing, 2014, p. 818-833. doi: 10.1007/978-3-319-10590-1_53.
- [44] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », 2014, doi: 10.48550/ARXIV.1409.1556.
- [45] C. Szegedy *et al.*, « Going deeper with convolutions », in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, juin 2015, p. 1-9. doi: 10.1109/CVPR.2015.7298594.
- [46] « K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016, pp. 770–778. Accessed: Dec. 07, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVP_R_2016_paper.html ».
- [47] T.-Y. Lin *et al.*, « Microsoft COCO: Common Objects in Context », 2014, doi: 10.48550/ARXIV.1405.0312.
- [48] J. Long, E. Shelhamer, et T. Darrell, « Fully convolutional networks for semantic segmentation », in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, juin 2015, p. 3431-3440. doi: 10.1109/CVPR.2015.7298965.
- [49] O. Ronneberger, P. Fischer, et T. Brox, « U-Net: Convolutional Networks for Biomedical Image Segmentation », 2015, doi: 10.48550/ARXIV.1505.04597.
- [50] J. Gu *et al.*, « Recent advances in convolutional neural networks », *Pattern Recognit.*, vol. 77, p. 354-377, mai 2018, doi: 10.1016/j.patcog.2017.10.013.
- [51] International Journal of Engineering Applied Sciences and Technology, 2020 Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316 Published Online April 2020 in IJEAST (<http://www.ijeast.com>).
- [52] D. Hendrycks et K. Gimpel, « Gaussian Error Linear Units (GELUs) », 2016, doi: 10.48550/ARXIV.1606.08415.
- [53] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, et M. Jorge Cardoso, « Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations », in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, vol. 10553, M. J. Cardoso, T. Arbel, G. Carneiro, T. Syeda-Mahmood, J. M. R. S. Tavares, M. Moradi, A. Bradley, H. Greenspan, J. P. Papa, A. Madabhushi, J. C. Nascimento, J. S. Cardoso, V. Belagiannis, et Z. Lu, Éd., in *Lecture Notes in Computer Science*, vol. 10553. , Cham: Springer International Publishing, 2017, p. 240-248. doi: 10.1007/978-3-319-67558-9_28.
- [54] M. A. Rahman et Y. Wang, « Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation », in *Advances in Visual Computing*, vol. 10072, G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaiff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, et T. Isenberg, Éd., in *Lecture Notes in Computer Science*, vol. 10072. , Cham: Springer International Publishing, 2016, p. 234-244. doi: 10.1007/978-3-319-50835-1_22.

- [55] D. P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization », 2014, doi: 10.48550/ARXIV.1412.6980.
- [56] O. Oktay *et al.*, « Attention U-Net: Learning Where to Look for the Pancreas », 2018, doi: 10.48550/ARXIV.1804.03999.
- [57] Y. Guo, J. Stein, G. Wu, et A. Krishnamurthy, « SAU-Net: A Universal Deep Network for Cell Counting », in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, Niagara Falls NY USA: ACM, sept. 2019, p. 299-306. doi: 10.1145/3307339.3342153.
- [58] J. Chen *et al.*, « TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation », 2021, doi: 10.48550/ARXIV.2102.04306.
- [59] A. Dosovitskiy *et al.*, « An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale », 2020, doi: 10.48550/ARXIV.2010.11929.
- [60] H. Cao *et al.*, « Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation », in *Computer Vision – ECCV 2022 Workshops*, vol. 13803, L. Karlinsky, T. Michaeli, et K. Nishino, Éd., in Lecture Notes in Computer Science, vol. 13803. , Cham: Springer Nature Switzerland, 2023, p. 205-218. doi: 10.1007/978-3-031-25066-8_9.
- [61] « OpenCV. Open Source Computer Vision Library. » 2015.
- [62] Chollet, F., & others., « Keras ». GitHub, 2015. [En ligne]. Disponible sur: <https://github.com/fchollet/keras>
- [63] Yingkai (Kyle) Sha, « yingkaisha/keras-unet-collection: v0.1.12 ». Zenodo, 4 septembre 2021. doi: 10.5281/ZENODO.5449801.
- [64] A. Jamali, S. K. Roy, J. Li, et P. Ghamisi, « TransU-Net++: Rethinking attention gated TransU-Net for deforestation mapping », *Int. J. Appl. Earth Obs. Geoinformation*, vol. 120, p. 103332, juin 2023, doi: 10.1016/j.jag.2023.103332.
- [65] L. P. Soares, H. C. Dias, et C. H. Grohmann, « Landslide Segmentation with U-Net: Evaluating Different Sampling Methods and Patch Sizes », 2020, doi: 10.48550/ARXIV.2007.06672.
- [66] D. Masters et C. Luschi, « Revisiting Small Batch Training for Deep Neural Networks ». arXiv, 20 avril 2018. Consulté le: 3 janvier 2024. [En ligne]. Disponible sur: <http://arxiv.org/abs/1804.07612>
- [67] N. Orlando *et al.*, « Effect of dataset size, image quality, and image type on deep learning-based automatic prostate segmentation in 3D ultrasound », *Phys. Med. Biol.*, vol. 67, n° 7, p. 074002, avr. 2022, doi: 10.1088/1361-6560/ac5a93.
- [68] C. Luo, X. Li, L. Wang, J. He, D. Li, et J. Zhou, « How Does the Data set Affect CNN-based Image Classification Performance? », in *2018 5th International Conference on Systems and Informatics (ICSAI)*, Nanjing: IEEE, nov. 2018, p. 361-366. doi: 10.1109/ICSAI.2018.8599448.
- [69] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, et P.-A. Heng, « DCAN: Deep contour-aware networks for object instance segmentation from histology images », *Med. Image Anal.*, vol. 36, p. 135-146, févr. 2017, doi: 10.1016/j.media.2016.11.004.

ANNEXE A DÉFINITIONS INTELLIGENCE ARTIFICIELLE

Apprentissage Machine : Algorithmes optimisés à la suite d'itérations qui effectuent un «apprentissage» afin de réaliser une tâche. L'apprentissage machine peut être supervisé ou non.

Apprentissage machine supervisé : Type d'apprentissage machine le plus répandu. L'apprentissage est fait en comparant les essais du modèle à une vérité de terrain (ground truth). Ceci implique qu'il existe une référence.

Apprentissage machine non-supervisé : Type d'apprentissage machine qui ne possède pas de vérité de terrain, mais plutôt des objectifs à atteindre. Par exemple, algorithmes de jeux d'échecs qui viennent à bout des meilleurs joueurs d'échecs, puisqu'ils ont appris à gagner plutôt qu'à copier.

Apprentissage Profond (Deep learning) : Concept d'initialisation de différents poids initiaux qui sont changés dans des nœuds organisés en couches (Layers). Les nœuds reçoivent des entrées, effectue la somme de ces entrées et une fonction d'activation est appliquée afin de générer une sortie au nœud. L'ajout de la fonction d'activation permet l'introduction de non-linéarités et permet donc au modèle de comprendre des interactions complexes entre les données.

Backpropagation: Est le concept d'adapter les paramètres internes du modèle en fonction de l'évolution du gradient de la fonction de Loss à chaque couche de neurones. Ceci permet de mettre à jour les paramètres internes du modèle en tentant de minimiser la fonction d'erreur.

Classification : Tâche potentielle pour un algorithme d'apprentissage machine qui consiste à séparer des données en différentes classes prédéterminées selon leurs propriétés. L'objectif est d'obtenir des valeurs discrètes ou des distributions de probabilité selon chaque classe.

Convolution : Opération mathématique utilisée notamment dans les réseaux neuronaux convolutifs qui consiste à appliquer un noyau et à le multiplier aux valeurs de pixels d'une image. L'objectif est d'extraire informations sur les relations entre les différents pixels.

Filtre (Feature map): Résultat d'une convolution, cette nouvelle représentation des données peut donner de l'information sur la relation entre les différents pixels de l'image initiale dépendamment des poids du noyau utilisés. Généralement, le pas entre chaque convolution est supérieur à 1, ce

qui fait en sorte que les dimensions des feature maps sont souvent inférieures à celles de l'image initiale.

Fonction d'activation : Fonction appliquée en sortie de neurone qui permet, dans les couches cachées (Hidden Layers) d'introduire des non-linéarités dans les prédictions, tout en évitant une disparition du gradient lors de l'étape de **Backpropagation**. En sortie de modèle, cette fonction permet de rendre les données séparables afin d'effectuer une décision par rapport aux données (classification/régression).

Fonction d'erreur (Loss) : représente une mesure de l'**erreur** du modèle entre la valeur prédite et la valeur attendue (ex. : $MSELoss = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$ où y_i est la valeur attendue et $\hat{y}_i$ est la valeur prédite par le model) ou la distance de la prédiction par rapport à un critère de performance (ex. : $DiceLoss = 1 - Dice_{coef}$).

Forward Pass : Est le trajet qu'effectuent les données d'entrées du modèle en passant à travers les couches de neurones afin d'obtenir une mesure de l'erreur du modèle par en calculant la Loss en sortie du modèle.

Hyper paramètres : Les hyperparamètres sont les paramètres manuellement ajustables du modèle, alors que les paramètres sont les valeurs intrinsèques au modèle tel que les poids ou les valeurs dans les noyaux des convolutions.

Intelligence Artificielle : Terme général pour décrire toutes les tâches dites intelligentes qui peuvent être réalisées par un ordinateur.

Itérations (Epochs) : Étapes d'optimisations au cours desquelles les poids sont changés au taux d'apprentissage en suivant la fonction d'optimisation afin de minimiser la Loss.

Nœud ou neurone : Points d'intersection des modèles par apprentissage profond où les poids différents subissent la même opération

Noyau : Le noyau de convolution peut généralement prendre la forme d'une matrice 3x3 ou 5x5 de valeurs initialisées aléatoirement et il est utilisé afin d'extraire les informations des images par des convolutions.

Optimiseur (Optimizer) : Méthode qui détermine comment l'optimisation du modèle sera effectuée par rapport au gradient de la fonction de loss.

Sur-ajustement (*Overfitting*): Phénomène couramment observé en apprentissage machine. Il se produit lorsque le modèle est trop bien entraîné à effectuer des prédictions sur ses données d'entraînement, mais qu'il est moins performant à effectuer des prédictions sur de nouvelles données. On dit alors que le modèle est peu généralisable.

Poids (*Weights*) : Les poids seront les multiplicateurs de chaque variable qui composent les fonctions de décision (classification ou régression) du modèle. Pour une fonction linéaire on pourrait avoir : $f(x) = w_0 + x_1 w_1 + x_2^2 w_2 + \dots + x_n^n w_n$. Les poids sont initialisés de manière aléatoire au début de l'apprentissage, puis sont modifiés à chaque itération d'apprentissage afin de trouver la fonction qui minimise la fonction de *loss*.

Régression : Tâche potentielle pour un algorithme d'apprentissage machine qui consiste à trouver la tendance ou la fonction correspondant à des données. L'objectif est d'obtenir des valeurs continues.

Segmentation Sémantique : Tâche de classification permettant de classer chaque pixel d'une image.

Structure d'encodage (*Backbone*) : La structure d'encodage aussi appelée backbone est la séquence de convolution choisie pour la phase d'encodage d'une architecture *UNet*. Elle permet d'augmenter ou de diminuer la profondeur du réseau en augmentant ou en diminuant le nombre de couches convolutives appliquées aux images afin d'en extraire de l'information.

Taille de parcelle (*Patch size*) : Les parcelles sont des sous-régions ou sections de l'image, la taille de parcelle est la taille de chacune de ces sous-régions.

Taille de lot (*Batch size*) : La taille de lot représente le nombre de parcelles (ou patch) qui seront évaluées à chaque étape d'entraînement. Lorsque toutes les parcelles d'entraînement ont été évaluées lors d'étapes d'entraînements, le modèle passe à la prochaine itération.

Taux d'apprentissage (*Learning rate*) : Facteur multiplicatif appliqué à chaque itération de l'étape d'optimisation.

Vérité de terrain (*Ground truth*) : Il s'agit du résultat que l'on tente d'obtenir en sortie du modèle d'apprentissage machine à partir d'une donnée d'entrée. Ceci implique qu'il est possible, soit avec du travail manuel ou avec d'autres informations externes de savoir quel est le résultat qu'on attend de nos données d'entraînement.

ANNEXE B PROTOCOLE DE PRÉPARATION DES ÉCHANTILLONS CELLULAIRES MARQUÉS PAR IP

Note : Les protocoles mentionnés servent d'indications générales et n'ont pas été réalisés dans le cadre de cette maîtrise. Les échantillons ont été fournis suite à d'autres travaux de recherche sur le marquage IP.

1. Les échantillons cellulaires (Cell Micro Arrays (CMA)) ont été fournis par le Pr. Mader de l'UdeM à Cécile Darviot et Raphaële Fernet-Brochu afin d'effectuer un marquage IP. Les étapes impliquées sont de manière très générale :
 - a. Fixation des échantillons dans de la formaline
 - b. Mise en bloc de paraffine des échantillons
2. Les échantillons ont été tranchés, puis colorés par Hématoxyline et Éosine (H&E) au CRCHUM par Raphaële Fernet-Brochu et Cécile Darviot. Les étapes impliquées sont de manière très générale :
 - a. Tranchage du bloc de paraffine contenant l'échantillon à une épaisseur de $4\mu m$.
 - b. Coloration avec l'Hématoxyline de Gill, puis par l'Éosine alcoolique Y diluée 50:50 avec de l'éthanol 100%.
 - c. Ajoute de Toluol ou de substitut de Xylène
 - d. Montage de la lame de microscopie
3. Les nanoparticules ont été couvertes de PEG lors d'une étape de PEGylation réalisée par Cécile Darviot. Cette PEGylation comporte les étapes générales suivantes :
 - a. Activation des thiols du SH-PEG
 - b. Incubation des NPs d'or avec le SH-PEG activé
 - c. Concentration des NPs
 - d. Lavage de la solution
 - e. Caractérisation de la PEGylation
4. Les nanoparticules PEGylées ont été fonctionnalisées par un protocole EDC-sulfoNHS optimisé et appliqué par Cécile Darviot. Les étapes simplifiées de cette fonctionnalisation sont :
 - a. Ajout de EDC, suivi de sulfo-NHS à une solution de NPs
 - b. Retrait de l'excès de EDC et de sulfo-NHS, puis resuspension de la solution de NPs

- c. Ajout d'anticorps à la solution de NPs
 - d. Dilution afin d'obtenir la concentration désirée de NPs
5. Les nanoparticules fonctionnalisées ont été ajoutées à un échantillon selon un protocole d'IP appliqué par Cécile Darviot et Raphaële Fernet-Brochu. Les grandes étapes de ce protocole sont :
- a. Blocage de l'échantillon avec de la BSA
 - b. Ajout de NPs fonctionnalisées et incubation.
 - c. Rinçage afin de retirer les NPs non attachées à l'échantillon
 - d. Coloration H&E de l'échantillon