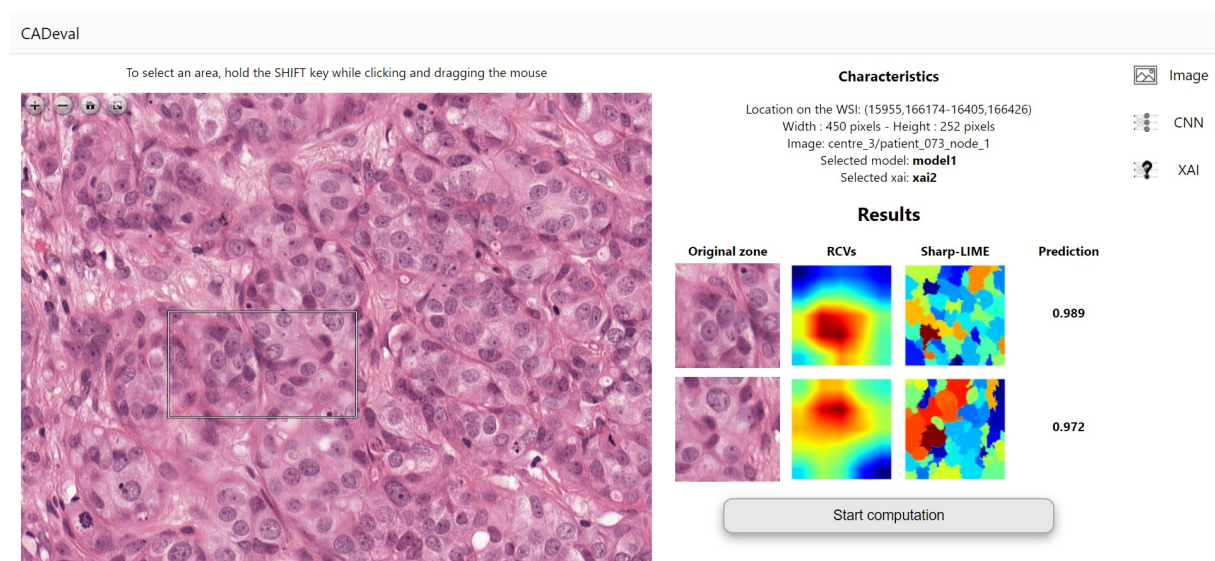


Travail de Bachelor 2021

Développement d'une interface web pour l'évaluation du diagnostic assisté par ordinateur (CADeval)



Etudiant : Nicolas Constantin

Professeur : Dr Henning Müller

Déposé le : 4 août 2021

Site web : www.hevs.ch

RÉSUMÉ

Ce projet s'inscrit dans le contexte d'un travail de Bachelor de la filière Informatique de gestion à la HES-SO Valais-Wallis. Il a été proposé par le professeur Henning Müller en collaboration avec Mara Graziani, doctorante à l'Institut d'informatique de gestion de la HES-SO de Sierre [IIG].

Ce travail est réalisé en collaboration avec l'unité eSanté de l'IIG. Des recherches y sont menées sur des algorithmes pour expliquer les méthodes d'aide à la décision pour la prédiction de cancers. L'utilisation d'outils de Deep Learning [DL] aidant les pathologistes dans le but de déceler des cancers à partir d'images pourrait être d'une grande aide pour la médecine.

Le travail de l'étudiant est de développer une interface web pour pouvoir naviguer sur des images médicales, sélectionner des zones d'image et lancer un script qui va rechercher l'éventuelle présence d'un cancer dans la zone sélectionnée.

Les scripts utilisant du DL seront développés en parallèle par Mara Graziani. Leur objectif est de calculer une prédiction de cancer. De plus, si l'utilisateur le désire, ils renverront des images aidant à comprendre le fonctionnement de l'algorithme de DL.

Mots-clés : Diagnostic assisté par ordinateur, Cancer, Deep Learning, Interprétabilité

AVANT-PROPOS

En cas de cancer, il est primordial d'établir un diagnostic clair. Il est possible que des métastases du cancer soient présentes dans le système lymphatique du patient. Dans ce cas, un traitement plus agressif est nécessaire. Pour améliorer cette recherche, des méthodes de DL sont mises au point. Leur but est d'analyser des images et de décider si des métastases du cancer sont présentes ou non. Cela ne remplace cependant pas l'avis d'un expert. L'un des plus gros inconvénients des algorithmes de DL réside dans la difficulté à expliquer la raison d'une réponse donnée. L'objectif de l'étudiant est, dans un premier temps, de développer un prototype d'interface web servant à aider les médecins à comprendre le comportement des modèles de DL.

Dans un second temps, l'étudiant devra analyser les temps de travail des différents algorithmes afin de décider si ces derniers devraient être utilisés avec la puissance du CPU ou du GPU. Des tests utilisateurs seront également effectués pour tester la qualité du prototype. Une vidéo explicative de l'outil sera envoyée à des professionnels du domaine. Un scénario leur sera ensuite proposé. Une fois celui-ci terminé, ils rempliront un questionnaire. Leurs réponses seront récupérées et analysées dans le but d'améliorer la qualité de l'interface développée.

REMERCIEMENTS

Je souhaite remercier toutes les personnes qui m'ont aidé pour la réalisation de ce travail de Bachelor.

Un merci particulier à :

Dr Henning Müller, mon professeur responsable, pour son aide et ses conseils durant toute la durée de ce travail.

Mme Mara Graziani, la mandante de ce travail, pour ses conseils, son suivi et pour la réalisation de la partie Deep Learning du projet.

M. Roger Schaer pour avoir assuré les accès aux serveurs, ainsi que pour son aide à la résolution des problèmes techniques durant ce travail.

Mme Michèle Constantin et M. Quentin Nater qui ont relu ce document.

TABLE DES MATIÈRES

TABLE DES ILLUSTRATIONS	VI
LISTE DES ABRÉVIATIONS	VIII
GLOSSAIRE	IX
1. INTRODUCTION	1
1.1. CONTEXTE.....	1
1.2. MOTIVATION.....	1
1.3. OBJECTIFS.....	1
2. ÉTAT DE L'ART	3
2.1. LA SOLUTION EXISTANTE : DESUTO	3
2.2. GESTION D'IMAGES MÉDICALES	4
2.3. ALGORITHMES DE DL EN LIEN AVEC LES IMAGES.....	5
2.4. CAD POUR LE CANCER DU SEIN	7
2.5. TESTS UTILISATEURS POUR CAD	9
2.6. MÉTHODES D'EXPLICATION D'IA	10
2.7. DOCKER	13
3. CHOIX TECHNOLOGIQUE.....	17
3.1. LANGAGE DE PROGRAMMATION	17
3.2. VISIONNEUSE D'IMAGE, OPENSEADRAGON	18
3.3. PLUGIN D'ANNOTATION.....	19
4. IMPLÉMENTATION.....	21
4.1. SÉLECTEUR DE PARAMÈTRE.....	21
4.2. ZONE DE GESTION D'IMAGE	24
4.3. ZONE DE RÉSULTATS.....	25
4.4. DÉPLOIEMENT	29
4.5. AUTRES SERVICES	30
5. RÉSULTATS.....	32
5.1. TESTS DE LATENCE	32
5.2. ÉVALUATIONS UTILISATEURS	36
6. DISCUSSION	37
6.1. AMÉLIORATIONS POSSIBLES	37

6.2. RÉTROSPECTIVE PERSONNELLE SUR LE TRAVAIL	38
7. CONCLUSION	39
7.1. CONCLUSION PERSONNELLE	39
RÉFÉRENCES.....	41
ANNEXES I : RÉSULTATS COMPLETS DES TESTS DE LATENCE	44
ANNEXE II : PRODUCT BACKLOG DU PROJET	46
ANNEXE III : JOURNAL DE BORD DU PROJET	47
DÉCLARATION DE L'AUTEUR	49

TABLE DES ILLUSTRATIONS

Figure 1 - Exemple de l'interface web Desuto - Image de l'auteur	3
Figure 2 - Illustration d'image pyramidale (van Ginneken, Kerkstra, & Meakin, 2012a)	4
Figure 3 - Illustration du DL comme branche du ML - Inspirée de javapoint.....	5
Figure 4 - Explication de la meilleure performance du DL par rapport aux anciennes méthodes de ML (Seif, 2018)	6
Figure 5 - Exemple de CNN (Amini, 2021a)	7
Figure 6 - Comparaison des performances d'analyse humaine par rapport à une machine (Beck, 2019).....	8
Figure 7 - Exemple de Grad-CAM - Image venant du projet CADeval	11
Figure 8 - Exemple de LIME - Image venant du projet CADeval.....	12
Figure 9 - Exemple de RCVs - Image créées par Mara GrazianiSharp-LIME	12
Figure 10 - Exemple de Sharp-LIME - Image créées par Mara Graziani	13
Figure 11 - Illustration du contenu d'un container Docker - Image de l'auteur	14
Figure 12 - Représentation de la matrice infernale - Image de l'auteur	14
Figure 13 - Architecture d'une machine virtuelle (Bashari Rad, Bhatti, & Ahmadi, 2017, p. 229).....	15
Figure 14 - Architecture d'un container Docker (Bashari Rad, Bhatti, & Ahmadi, 2017, p. 230)...	15
Figure 15 - Résultats de la comparaison des différents frameworks javascript (Levlin, 2020, p. 70)	18
Figure 16 - Comparaison des plugins d'annotation - Image de l'auteur	20
Figure 17 - Parties de l'application CADeval - Image de l'auteur	21
Figure 18 - Sélection des centres - Image de l'auteur.....	22
Figure 19 - Sélection des images - Image de l'auteur	22
Figure 20 - Sélection du modèle CNN - Image de l'auteur	23
Figure 21 - Sélection du XAI - Image de l'auteur	23
Figure 22 - Boutons du lecteur d'image - Image de l'auteur.....	24
Figure 23 - Gestion d'une zone - Image de l'auteur.....	25
Figure 24 - Caractéristiques d'une image et bouton pour lancer le modèle - Image de l'auteur ..	26
Figure 25 - Message indiquant que le modèle travaille - Image de l'auteur	26
Figure 26 - Message d'erreur, la zone est trop grande ou trop petite - Image de l'auteur.....	27
Figure 27 - Message d'erreur, la zone n'est pas valide pour le modèle - Image de l'auteur.....	27
Figure 28 - Exemple d'autre erreur durant les calculs du CNN - Image de l'auteur	28
Figure 29 - Exemple de résultat sans XAI - Image de l'auteur.....	29
Figure 30 - Exemple de résultat avec XAI - Image de l'auteur	29
Figure 31 - Interaction entre les composants - Image de l'auteur.....	30
Figure 32 - Synthèse des tests de latence sur GPU avec XAI - Image de l'auteur	33
Figure 33 - Synthèse des tests de latence sur GPU sans XAI - Image de l'auteur	34

Figure 34 - Synthèse des tests de latence sur CPU avec XAI - Image de l'auteur.....	34
Figure 35 - Comparaison directe entre un test sur CPU et sur GPU - Image de l'auteur	35
Figure 36 - Synthèse des tests de latence sur CPU sans XAI - Image de l'auteur.....	35
Figure 37 - Prototype de l'affichage de la carte thermique en dessous des résultats actuels - Image de l'auteur.....	37

LISTE DES ABRÉVIATIONS

API	Application Programming Interface - Interface de programmation d'application (FR)
CAD	Computer Aided Diagnostic - Diagnostic assisté par ordinateur (FR)
CNN	Convolutional Neural Network
CPU	Central Processing Unit (processeur)
CSS	Cascading Style Sheet
DL	Deep Learning
DOM	Document Object Model
GPU	Graphics Processing Unit (carte graphique)
Grad-CAM	Gradient-weighted class activation mapping
HES-SO	Haute École Spécialisée de Suisse Occidentale
HTML	Hyper-Text Markup Language
HTTPs	Hyper-Text Transfer Protocol secured
IA	Intelligence Artificielle
IIG	Institut Informatique de Gestion de la HES-SO Valais-Wallis
JSON	JavaScript Object Notation
LIME	Local interpretable model-agnostic explanations
ML	Machine Learning
NAS	Network Attached Storage - Serveurs de stockage en réseau (FR)
Osd-Annotations	OpenSeadragon-Annotations
RAM	Random Access Memory (Mémoire vive)
RCVs	Regression Concept Vectors

RGPD	Règlement Général sur la Protection des Données
Sharp-LIME	Sharpening Local Interpretable Model-agnostic Explanation
VM	Virtual Machine - machine virtuelle (FR)
WSI	Whole Slide Images
XAI	Explainable Artificial Intelligence - Explications d'intelligence artificielle (FR)

GLOSSAIRE

Algorithme	Instructions et opérations permettant de résoudre une série de problèmes
API	Interface qui rend des services disponibles de manière simple
Cache d'une application	Données stockées temporairement dans la mémoire
Computer Vision	Analyse d'images par des ordinateurs
Document Object Model	Interface de programmation permettant de manipuler la structure d'un document HTML (Mozilla Developer Network, 2021)
Framework	Environnement de travail
Frontend	Partie visible par l'utilisateur dans une application
GitHub	Outil de partage de code
Histopathologie	Réalisation d'un diagnostic à partir de l'étude des tissus au microscope
Hyperviseur	Plateforme permettant de virtualiser un système d'exploitation
Inférence	Techniques permettant d'introduire des caractéristiques dans des groupes
Intelligence artificielle	Techniques permettant à une machine de reproduire un comportement humain

Kernel	Noyau d'un système d'exploitation
Machine virtuelle	Reproduction virtuelle d'un système d'exploitation
Métastase	Tumeur présente dans une autre partie du corps que celle où est présente le cancer principal
NAS	Périphérique de stockage connecté au réseau
Nodule pulmonaire	Lésion au niveau des poumons
NVIDIA	Entreprise américaine de fabrication de processeurs graphiques, de cartes graphiques et de puces graphiques.
Open source	Licence permettant d'utiliser, modifier et partager le code source du logiciel
Pattern	Modèle simplifié ou répété d'une structure
Plugin	Extension d'un logiciel permettant d'y ajouter des fonctionnalités
Projet Eurostars	Programme proposant des projets d'innovation en collaboration avec plusieurs pays dans le monde
Readme	Fichier texte contenant des informations sur les autres fichiers du dossier dans lequel il se trouve
Superpixels	Groupe de pixels qui ont des caractéristiques en commun
Système lymphatique	Circuit parcourant tout le corps et comprenant des vaisseaux, des ganglions lymphatiques et certains organes
Tumeur	Grosseur due à une multiplication démesurée de cellules

1. Introduction

1.1. Contexte

Ce travail est la synthèse de trois ans d'études en Informatique de Gestion à la HES-SO Valais-Wallis. Il doit être réalisé en 300 à 360 heures de travail. De nombreux sujets ont été proposés et l'étudiant, après avoir sélectionné ses préférences, s'est vu remettre ce projet de développement d'interface web pour l'évaluation du diagnostic assisté par ordinateur [CAD]. Ce travail a été écrit entre le 10 mai et le 4 août 2021.

1.2. Motivation

Le 27 avril 2021, le Parlement européen a voté le Règlement Général sur la Protection des Données [RGPD]. Cette loi régit entre autres la gestion et le stockage des données. Malgré le fait que la Suisse ne fasse pas partie de l'Union Européenne, elle y est quand même soumise.

Dans l'article 22 de la RGPD, il est spécifié qu'une personne a le droit de demander qu'une décision la concernant ne soit pas uniquement fondée sur un traitement automatisé par une intelligence artificielle [IA]. (Parlement européen, 2016).

Cela signifie que dans le cas de l'utilisation d'un algorithme de DL pour détecter un cancer, il est obligatoire de présenter des explications de ses décisions si le patient le demande. Cependant, les réseaux de neurones sont semblables à des boîtes noires. Il n'est pas possible de savoir exactement ce qu'il s'y passe. Toutefois, il existe plusieurs techniques qui permettent d'expliquer les décisions des modèles de DL utilisés pour les CAD.

1.2.1. Motivation personnelle

Durant trois ans d'études à la HES-SO Valais-Wallis, j'ai pu découvrir plusieurs aspects de l'informatique et plusieurs langages de programmation. J'ai particulièrement apprécié de découvrir les domaines des langages web, des algorithmes ainsi que l'introduction au Machine Learning [ML]. Parmi les sujets de Bachelor proposés, celui de l'évaluation du diagnostic assisté par ordinateur était celui qui correspondait le plus à mes préférences. Il mêle l'application d'un langage web avec l'utilisation de modèles de DL, ce qui m'intéresse énormément. Je n'avais jamais approché le domaine médical, mais c'est un sujet très vaste qui m'a toujours intrigué, et c'est avec grand plaisir que j'en découvre un aspect avec ce travail.

1.3. Objectifs

L'objectif de ce projet est de répondre à la motivation en développant une interface web qui va s'inspirer de la solution existante, Desuto (voir le chapitre 2.1 la concernant). Elle devra avoir les fonctionnalités suivantes :

- Afficher des images Whole Side Images [WSI], comme expliqué dans le chapitre 2.2 Gestion d'images médicales
- Sélectionner une zone de l'image affichée
- Choisir l'image affichée, le modèle de DL désiré ainsi que le type d'explication recherchée
- Afficher la prédiction du modèle
- Afficher les explications demandées

1.4. Méthodologie

La méthodologie utilisée durant ce travail est inspirée de la méthodologie Agile. Cette dernière met en avant la communication et la collaboration entre équipes de développement et client. Le point le plus important de la méthodologie Agile est la communication. Durant la réalisation de ce projet, une réunion hebdomadaire est planifiée avec Mara Graziani, la mandante de ce travail, ainsi qu'avec Dr Henning Müller, le professeur responsable. Durant celle-ci, deux points sont abordés. Tout d'abord, les travaux effectués lors de la semaine sont présentés et les difficultés rencontrées sont discutées au besoin. Une fois cette partie terminée, les modifications à apporter et les ajouts souhaités pour la semaine suivante sont définis. Un journal de bord est rédigé, comprenant tous les travaux effectués par l'étudiant.

2. État de l'art

2.1. La solution existante : Desuto

En 2015, un prototype nommé Desuto (figure 1) a été développé par l'IIG dans le cadre d'un projet Eurostars, en collaboration avec l'entreprise ContextVision¹. Il permet d'afficher des images médicales depuis une base de données présente sur un serveur. Il est également possible d'ajouter nos propres images directement sur le site. Ces images sont zoomables. Un outil pour sélectionner des régions de l'image a également été créé, permettant de les annoter. Une fois qu'une zone a été sélectionnée, des informations s'affichent sur le côté. On peut y voir la taille de la zone sélectionnée, un aperçu de la zone ainsi que des images similaires prises dans la base de données. Un lien cliquable existe également pour rechercher plus d'images similaires. L'objectif de cette interface web était de pouvoir faire de la récupération d'images. Un système était nécessaire pour de l'aide au diagnostic. Il était possible de faire ressortir des régions avec une forte probabilité de présence de tumeurs. (Schaer, Otàlora, Jimenez-del-Toro, Atzori, & Müller, 2019, pp. 2-8).

Les objectifs de DESUTO étaient différents de ceux de CADeval.

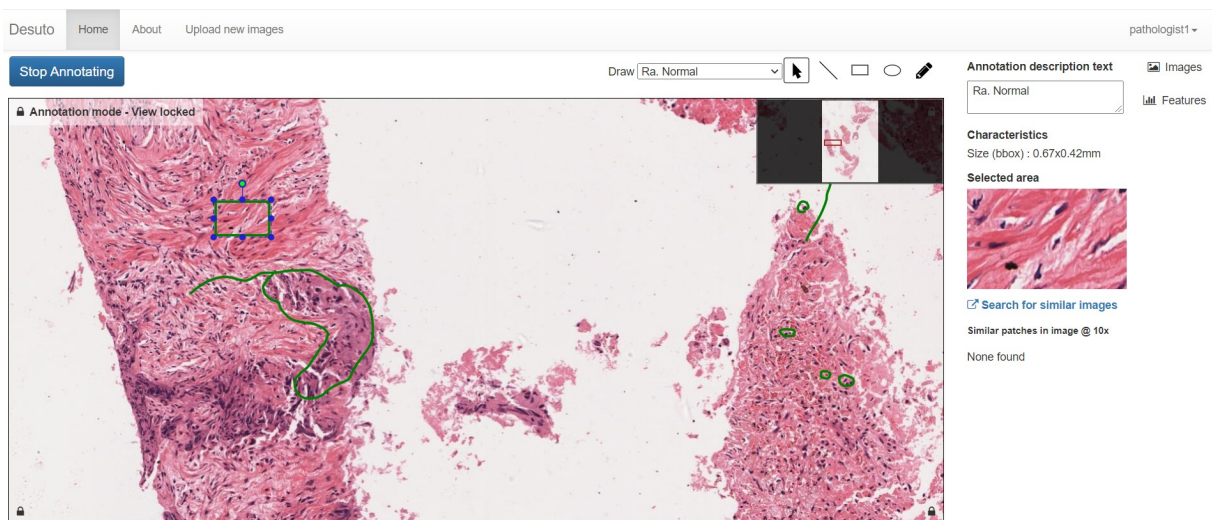


Figure 1 - Exemple de l'interface web Desuto - Image de l'auteur

¹ Entreprise suédoise spécialisée dans l'analyse d'images médicales grâce à de l'intelligence artificielle
(<https://www.contextvision.com/>)

2.2. Gestion d'images médicales

Les images médicales utilisées pour ce projet sont des images dites WSI. Bândi P. et al. expliquent que ce sont des images très grandes, dépassant souvent les 100 000 X 100 000 pixels (cité dans Graziani, et al., 2020, pp. 782-783). En raison de leur taille elles ne peuvent pas être traitées comme une image standard.

Concernant les WSIs, plusieurs standards d'image existent. Les images traitées durant ce projet sont en format BigTIFF. Les lecteurs classiques ne pouvant pas lire des images aussi grandes, un lecteur particulier est donc requis pour les afficher. Ce lecteur requiert des images multi-échelles, à structure pyramidale, comme montré dans la figure 2. L'outil IIPImage a été utilisé pour transformer les images BigTIFF en images zoomables.

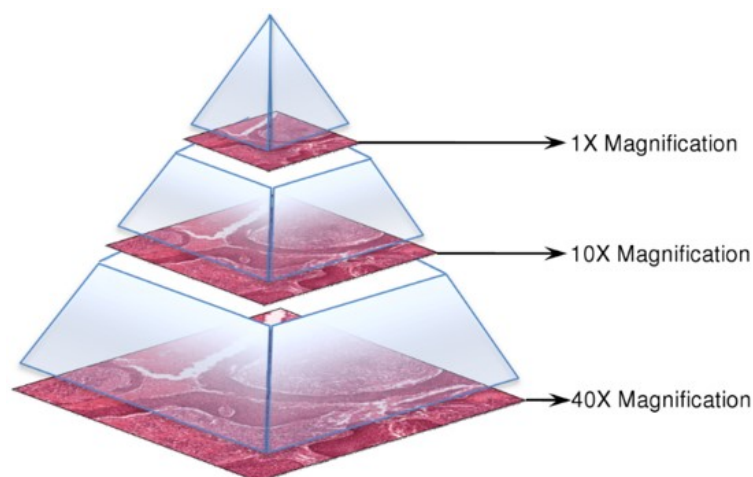


Figure 2 - Illustration d'image pyramidale (van Ginneken, Kerkstra, & Meakin, 2012a)

2.2.1. Banque d'images

Les images utilisées dans ce travail sont prises dans la banque d'images du défi Camelyon17. C'est une compétition scientifique organisée par le groupe d'analyse d'images diagnostiques (DIAG) et le département de pathologie du centre médical de l'université Radboud (Radboudumc) à Nijmegen, aux Pays-Bas. L'objectif du défi est d'évaluer des algorithmes pour la détection et la classification automatique des métastases du cancer du sein dans des images de coupes histologiques de ganglions lymphatiques. (van Ginneken, Kerkstra, & Meakin, 2012b).

Bândi P. et al. affirment qu'actuellement, il s'agit de l'un des ensembles de données les plus grands et les plus représentatifs existant pour la recherche en histopathologie sur les ganglions lymphatiques du cancer du sein (cité dans Graziani, et al., 2020, p. 787).

2.3. OpenSeadragon

OpenSeadragon est un lecteur d'images zoomables open-source orienté web pour des images zoomables à haute résolution. Il est implémenté en Javascript pur. (OpenSeadragon, 2015).

Le principe est simple. L'image affichée par ce lecteur est en basse résolution par rapport à celle d'origine. Il est ensuite possible de zoomer dans l'image. Lorsque qu'elle est agrandie, seule la partie visible est chargée. Cela permet de ne jamais avoir besoin de charger toute l'image et de conserver ainsi sa qualité. Ce lecteur sera implémenté dans ce projet.

2.4. Algorithmes de DL en lien avec les images

Le DL est une branche du ML comme illustré dans la figure 3.

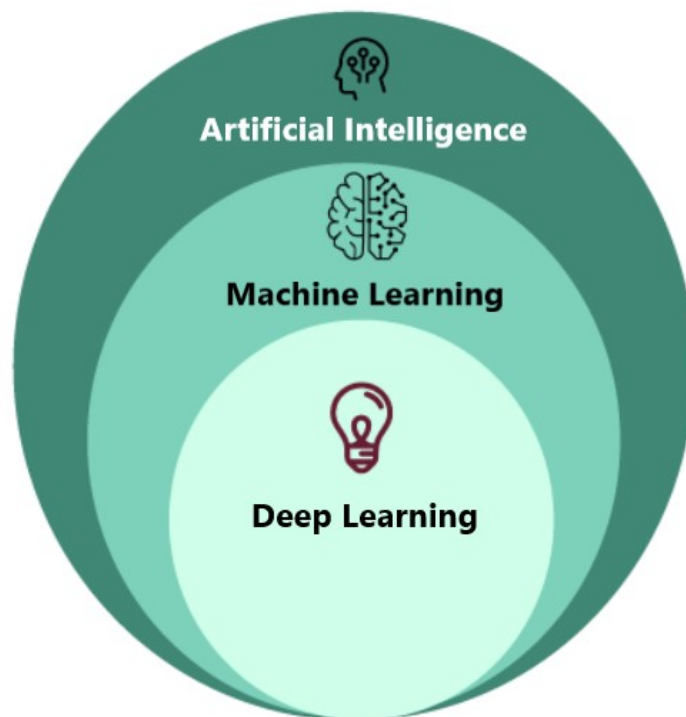


Figure 3 - Illustration du DL comme branche du ML - Inspirée de javapoint²

Le ML est l'étude d'algorithmes informatiques qui permettent aux programmes de s'améliorer automatiquement en gagnant de l'expérience (Mitchell, 1997, p. 1).

Grâce au ML, une machine est capable d'apprendre sans avoir explicitement été programmée pour cela. Le DL va un peu plus loin, en étant capable d'extraire des patterns depuis des données en utilisant des réseaux de neurones. (Amini, 2021b).

² Article « Difference between Machine Learning and Deep Learning » <https://www.javatpoint.com/machine-learning-vs-deep-learning>

Comme illustré par la figure 4, les algorithmes de DL s'adaptent également beaucoup mieux que les algorithmes de ML classiques avec de grands volumes de données (Seif, 2018).

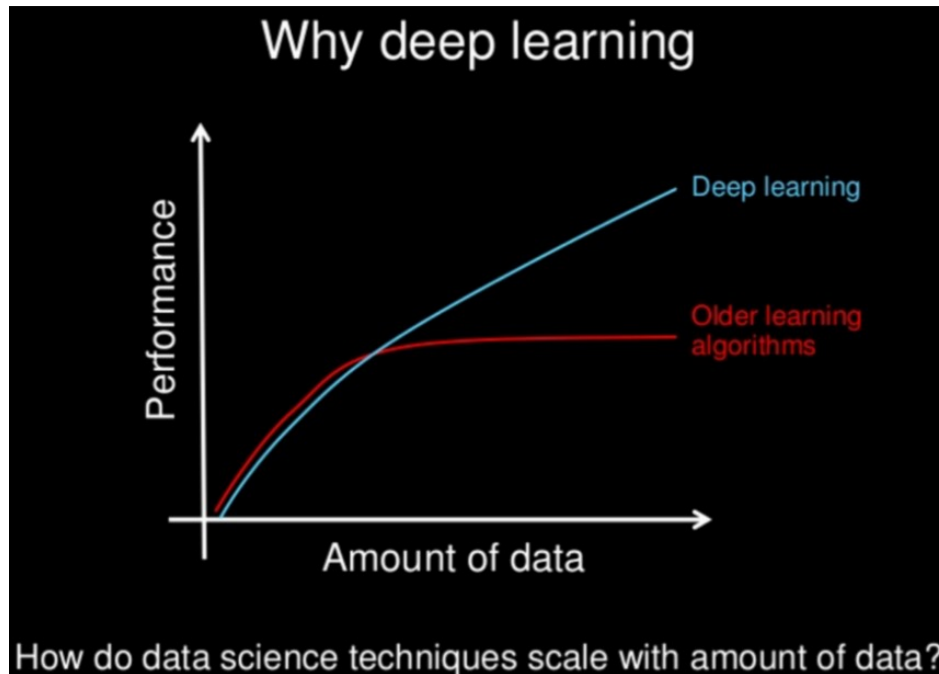


Figure 4 - Explication de la meilleure performance du DL par rapport aux anciennes méthodes de ML (Seif, 2018)

Le DL fonctionne fondamentalement comme une boîte noire. On connaît les données qui entrent dans l'algorithme ainsi que leur cheminement, mais on ne sait pas exactement ce qu'il s'y passe.

2.4.1. Convolutional Neural Network

Le Convolutional Neural Network 2D [CNN] est un algorithme de DL spécialisé dans le traitement de tableaux en deux dimensions. Il est donc très utilisé pour de la reconnaissance d'images. Le principe basique du CNN est de découper les images entrantes en morceaux d'images plus petits. Chaque petit morceau d'image sera ensuite traité par un neurone. (Amini, 2021a).

Les CNNs possèdent des paramètres qui sont entraînés sur de grands ensembles de données. On voit dans l'exemple de la figure 5 le chemin emprunté par un segment d'image. Celui-ci va activer différents neurones les uns après les autres. Ils se comportent comme des filtres, détectant les spécificités de chaque image. Finalement, le modèle de CNN va définir s'il reconnaît quelque chose dans l'image. Pour cela, il va le comparer à sa base de données en mémoire et tenter de le classer dans une catégorie connue. En regroupant les résultats de tous les segments d'images analysés, la machine va trouver tous les composants d'images qu'il connaît. Il est par exemple capable de trouver précisément tous les humains dans une foule, de reconnaître des chats ou des chiens, ou encore, dans un contexte médical, de détecter des cellules cancéreuses parmi une multitude de cellules saines.

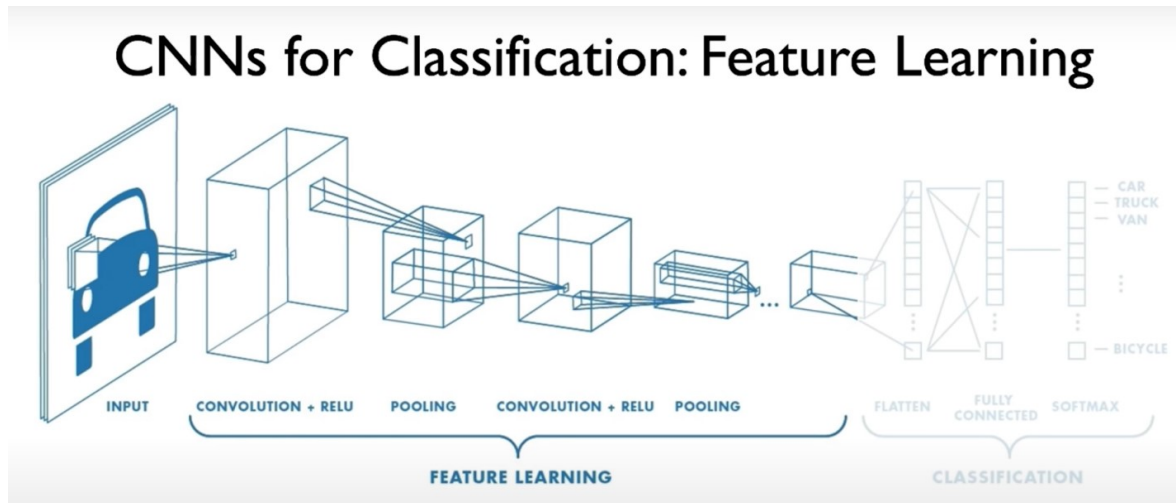


Figure 5 - Exemple de CNN (Amini, 2021a)

2.5. CAD pour le cancer du sein

Dans la majorité des cas, pour détecter un cancer du sein, les médecins font une mammographie régulière des patients à risques. Cela s'appelle le screening. Lorsqu'il est avéré qu'un patient est atteint d'un cancer du sein, les pathologistes pratiquent l'histopathologie. Cette technique consiste en la recherche de métastases du cancer du sein dans le système lymphatique. En cas de présence de métastases, un traitement plus agressif est nécessaire.

2.5.1. Pourquoi utiliser un CAD ?

L'histopathologie est une tâche difficile pour un humain. Sur la figure 6, on peut constater le taux d'erreurs faites par un pathologiste dans diverses situations. Il apparaît qu'en compétition, un pathologiste ne fait que 3.5% de fausses prédictions, tandis qu'en situation ordinaire sur des très petites tumeurs, le taux d'erreurs peut monter jusqu'à 42%, soit presque une chance sur deux. (Beck, 2019).

Pour essayer d'améliorer la qualité de détection de cancers, les chercheurs ont tenté de développer des algorithmes pour aider les pathologistes à trouver les tumeurs. Ils ont pour cela programmé des IA. En leur donnant des images de cellules, ces algorithmes tentent de trouver des patterns leur permettant d'affirmer la présence ou l'absence de cancer. Dans la figure 6, on peut voir que les modèles de DL font actuellement moins d'1% d'erreur dans leurs prédictions.

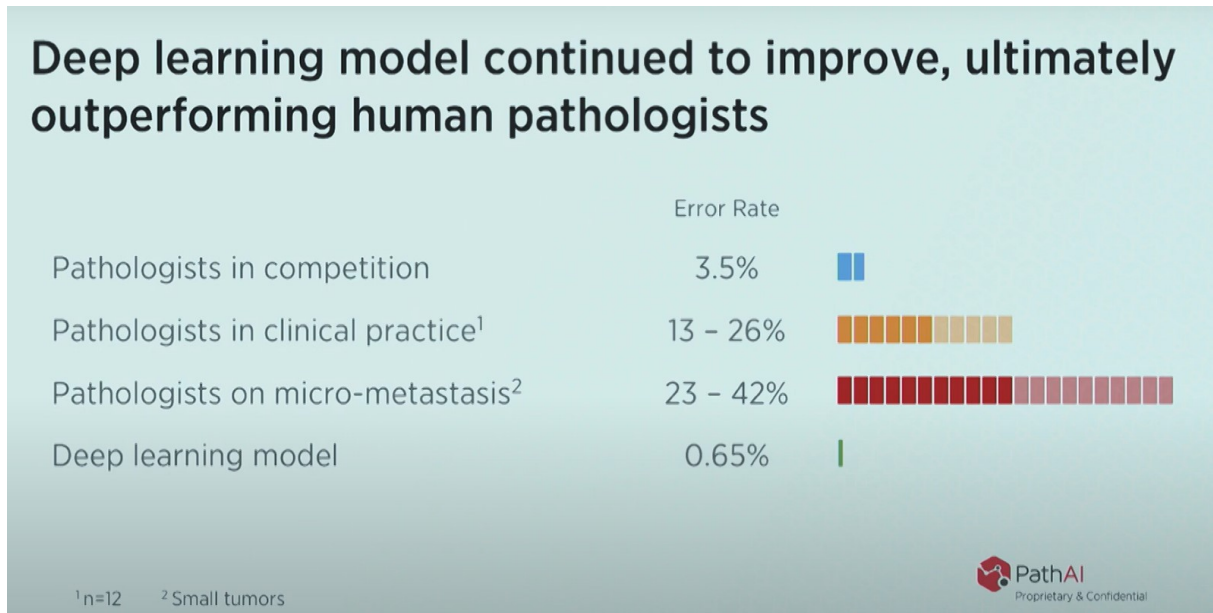


Figure 6 - Comparaison des performances d'analyse humaine par rapport à une machine (Beck, 2019)

Cela ne reste cependant qu'un outil à la disposition des pathologistes. Un outil informatique ne peut pas remplacer un humain lors de la décision finale de l'application d'un traitement. Quand un pathologiste fait un diagnostic, il en prend l'entière responsabilité. S'il décide de suivre la décision d'une IA, c'est lui qui en assume tous les risques.

Dr Henning Müller explique qu'actuellement les outils informatiques basés sur des images ne sont pas couramment utilisés. Ils sont testés dans beaucoup d'hôpitaux pour des applications spécifiques comme les mammographies ou les nodules pulmonaires. Concernant la détection de cancer, la plupart des hôpitaux ne sont pas encore équipés pour permettre l'utilisation de solutions assistées par ordinateur. Concernant l'hôpital du Valais, les premiers scanners pour images seront installés cet automne. (Communication personnelle, 30 juin 2021).

2.5.2. Entraînement de l'algorithme

Tout algorithme de DL doit recevoir des données initiales pour pouvoir ensuite analyser des données. Cette partie s'appelle l'entraînement de l'algorithme.

L'ensemble de données pour l'entraînement doit être rigoureusement choisi. Il devrait contenir toutes les anomalies que le système va ensuite pouvoir reconnaître, mais aussi des cas sains. De plus, il devrait contenir des données représentatives de la distribution de probabilité de la population cible (proportion homme/femme, malade/sain...). En pratique, il est fréquent qu'un sous-groupe de patients soit sur- ou sous-représenté. Cela peut engendrer des baisses de performances de l'algorithme. (Petrick, et al., 2013, p. 087001-4).

L'algorithme doit ensuite être testé pour vérifier son efficacité. Pour l'ensemble de données de test, les conditions devraient être les mêmes que pour celui d'entraînement. Il faut cependant faire attention à ce que les données soient toutes différentes entre les deux ensembles de données.

Chaque donnée de test doit avoir une « référence standard ». C'est une information du statut du patient contenant la maladie concernée et d'autres informations telles que la localisation et l'étendue de celle-ci. (Petrick, et al., 2013, p. 087001-6).

Pour évaluer la qualité de l'algorithme, chaque donnée de test va être confiée à celui-ci, et une comparaison sera faite entre sa réponse et la référence standard de l'image. Moins l'algorithme fera d'erreurs, plus il sera efficace.

2.6. Tests utilisateurs pour CAD

2.6.1. Pourquoi faire des tests utilisateurs ?

Étant donné que les technologies digitales assistant les pathologistes ne sont pas encore très répandues, il est nécessaire de faire des tests en collaboration avec les professionnels pour être sûr de répondre aux besoins de la profession (Tonekaboni, Joshi, McCradden, & Goldenberg, 2019, pp. 1-2).

Si des pathologistes utilisent des solutions d'aide à la décision pour la première fois, il est important de leur assurer que le système fonctionne. Miller dit que si les pathologistes font confiance à l'IA, ils pourront plus facilement obtenir de bons résultats lors de son utilisation. (cité dans Hoffman, Mueller, Klein, & Litman, 2019, p. 3).

Afin d'assurer une qualité de logiciel optimal, il est important de développer le CAD de pair avec son évaluation. Les performances des algorithmes, leur efficacité d'utilisation et les comparaisons avec d'autres systèmes similaires d'utilisation devraient impérativement être prises en compte. (Petrick, et al., 2013, p. 087001-3).

Les tests utilisateurs sont utiles pour vérifier plusieurs aspects :

- Le système doit être simple et intuitif pour que les utilisateurs finaux puissent l'employer aisément.
- Le système doit être performant et optimisé.
- Toutes les fonctionnalités importantes pour les utilisateurs sont implémentées dans le système.

(Markonis, et al., 2015, pp. 774-783).

Ces trois vérifications peuvent être communes à toutes les applications. D'autres critères de vérification peuvent évidemment être ajoutés à la liste en fonction du système testé.

2.6.2. Comment faire des tests utilisateurs ?

Il existe de nombreuses façons de faire des tests utilisateurs. Souvent, les testeurs assistent à une démonstration du fonctionnement de l'outil qu'ils vont devoir évaluer, accompagnée d'instructions détaillées sur le système. Ils vont pouvoir ensuite réaliser un scénario préparé à l'avance. Celui-ci va être analysé pour pouvoir déterminer les améliorations à effectuer dans le système. Pour récupérer les informations nécessaires durant le scénario, il existe plusieurs possibilités :

- Les utilisateurs peuvent être filmés pendant qu'ils testent le système. Dans ce cas, il est possible de leur demander de « penser à voix haute », c'est-à-dire qu'ils expliquent pourquoi ils font chaque action. Cela permet de voir leur compréhension du système.
- Une personne peut les observer et prendre des notes en direct derrière eux. Il est également possible de leur poser des questions pendant et après leur test.
- Un questionnaire peut être rempli par les testeurs après leur expérience.

La méthode choisie varie selon ce que l'on veut faire tester. La plupart du temps, pour le test d'application informatique, les testeurs remplissent un questionnaire après avoir effectué leur scénario.

2.7. Méthodes d'explication d'IA

En raison de l'obligation d'appliquer la RGPD, il est obligatoire de pouvoir donner une explication des décisions des algorithmes de DL.

De plus, ces processus en boîte noire sont incompréhensibles par les humains. Les utilisateurs finaux ne sont pas nécessairement familiers avec le DL. Ces algorithmes peuvent être utilisés dans des pratiques quotidiennes à hauts risques. Certains diagnostics recommandent un traitement agressif et ont un impact important sur la vie des patients. Pour expliquer leur jugement, les médecins ont une sémantique qui se concentre sur la taille, la forme ou l'aspect des régions malades. Les algorithmes, eux, expliquent leur choix en pixels et en réponses de neurones. Il faut donc trouver des manières d'expliquer les choix des modèles utilisés, c'est pourquoi les méthodes d'explication d'IA [XAI] ont été inventées. (Graziani, Andrearczyk, Marchand-Maillet, & Müller, 2020a, p. 1).

Les premiers travaux qui ont été réalisés dans ce sens sont les travaux d'Erhan, Dumitru, et al. « Visualizing higher-layer features of a deep network. » à l'université de Montréal en 2009, et ceux

de Zeiler, Matthew, et Fergus, « Visualizing and understanding convolutional networks. » publiée par Springer pendant la conférence européenne sur la « computer vision » à Cham en 2014.

Il existe de nombreux types de XAI. Pour ce travail, quatre types sont utilisés, le Grad-CAM, le LIME, le RCVs et le Sharp-LIME.

2.7.1. Grad-CAM

La technique Gradient-weighted class activation mapping [Grad-CAM] peut être utilisée sur presque n'importe quel CNN pour la classification d'images. Ce XAI produit une carte thermique de l'image d'entrée. Plus une partie de l'image correspond à la classe recherchée, plus elle sera colorée en rouge. (Cian, van Gemert, & Lengyel, 2020, p. 4).

La figure 7 montre un exemple de Grad-CAM.

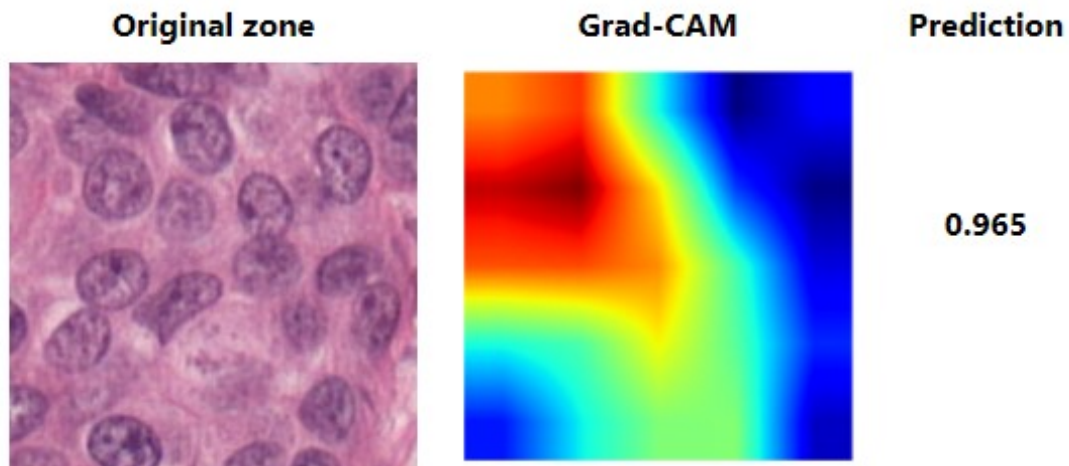


Figure 7 - Exemple de Grad-CAM - Image venant du projet CADeval

2.7.2. LIME

Local interpretable model-agnostic explanations [LIME] est une technique d'explication agnostique, c'est-à-dire qu'il fonctionne sans connaître le modèle qu'il analyse. Il peut être appliqué à plus ou moins tout type de classificateur. Le modèle va générer une carte à partir des superpixels de l'image originale. Les superpixels qui contribuent positivement à la prédiction seront en vert et ceux qui y contribuent négativement seront en rouge. (Cian, van Gemert, & Lengyel, 2020, p. 4).

La figure 8 montre un exemple de LIME.

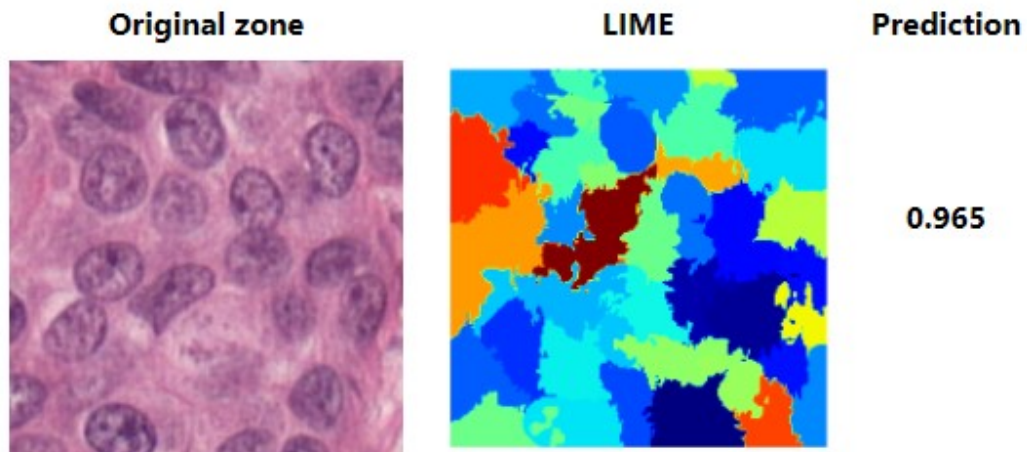


Figure 8 - Exemple de LIME - Image venant du projet CADeval

2.7.3. RCVs

Regression concept vectors [RCVs] est une technique de XAI qui prend la matrice de sortie d'un CNN comme base pour son algorithme. Il peut donc être utilisé pour tous les CNNs qui utilisent des images. La matrice va être transformée en un tableau à une dimension, qui sera ensuite traité par le système. Après plusieurs calculs, le RCVs déterminera une évaluation de plusieurs facteurs choisis. Il va évaluer pour ces facteurs si, dans l'image, ils ont servi ou non à déterminer le résultat, et à quel taux. (Graziani, Andrearczyk, Marchand-Maillet, & Müller, 2020).

La figure 9 montre un exemple de RCVs.

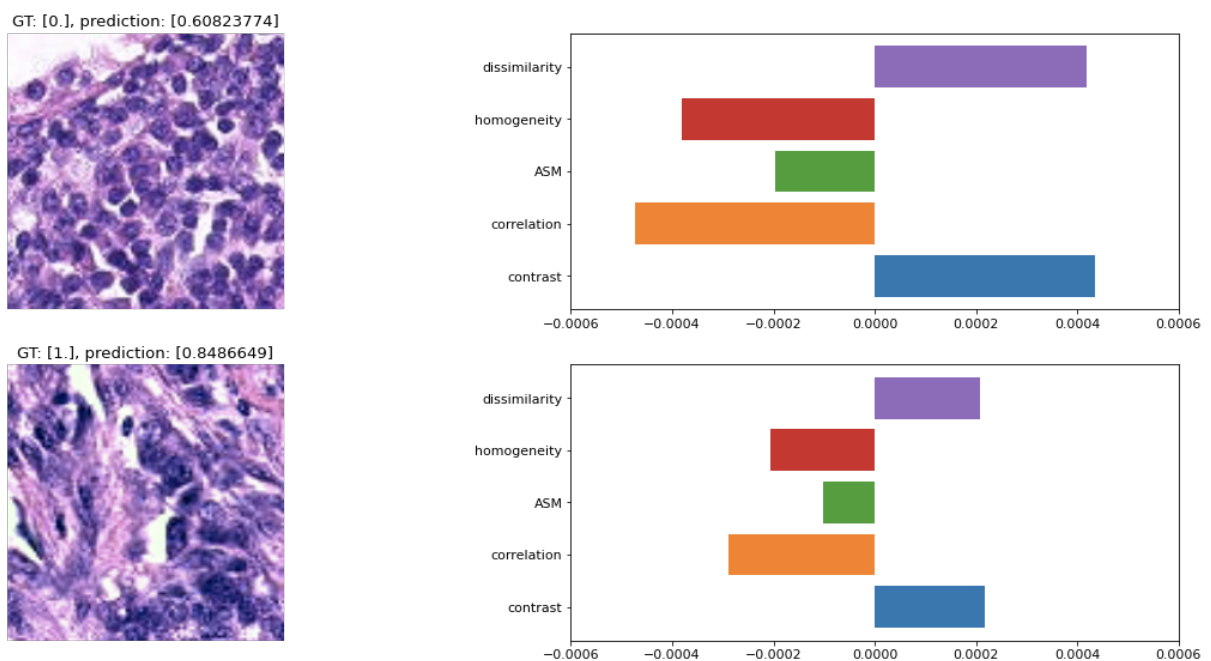


Figure 9 - Exemple de RCVs - Image créées par Mara GrazianiSharp-LIME

2.7.4. Sharp-LIME

Sharpening Local Interpretable Model-agnostic Explanation [Sharp-LIME] est une modification du modèle LIME. Ce dernier est beaucoup utilisé en radiologie et en histopathologie. Il a été créé entre autres pour être plus facilement compréhensible par les experts du domaine que les techniques employées couramment telles que LIME et Grad-CAM. Le contour des noyaux des cellules est utilisé pour définir les superpixels à la place d'autres techniques de segmentation employées par LIME. L'arrière-plan des superpixels est ensuite divisé en neuf carrés de taille fixe. Cela permet de diminuer la différence d'importance entre les noyaux de cellules et les zones d'arrière-plan, ce qui réduirait l'efficacité du XAI. Les zones sont ensuite mises en couleur comme pour l'algorithme LIME. Les zones qui contribuent positivement à la prédiction seront en bleu et celles qui y contribuent négativement seront en rouge. (Graziani & De Sousa, et al., 2021, pp. 3-5).

La figure 10 montre un exemple de Sharp-LIME.

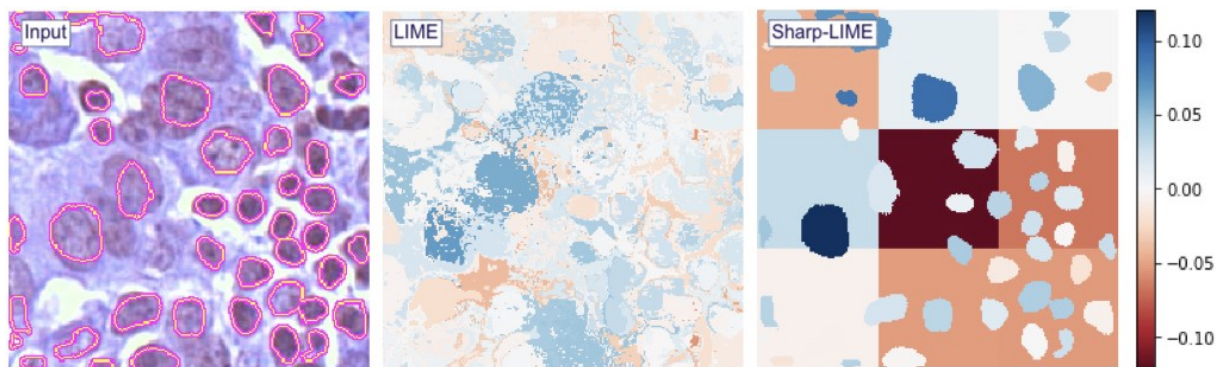


Figure 10 - Exemple de Sharp-LIME - Image créées par Mara Graziani

2.8. Docker

Boettiger C. explique que Docker est une plateforme open source qui exécute des applications et qui facilite le processus de développement et de distribution. Les applications utilisées avec Docker sont stockées dans des systèmes appelés containers. Ces containers sont développés dans le but d'être utilisés de manière isolée par rapport au système. (cité dans Bashari Rad, Bhatti, & Ahmadi, 2017, p. 228).

2.8.1. Images Docker

Docker est constitué de plusieurs couches. Ces couches sont divisées en deux parties, les images et les containers. Les images sont en lecture seule. Elles représentent en général le kernel et les différentes parties de l'application qui ne changent jamais. Le container représente la couche modifiable de l'application. C'est également lui qui exécute le processus. (Bashari Rad, Bhatti, & Ahmadi, 2017, pp. 228-229).

On peut imaginer cela comme des blocs qui s'empilent les uns sur les autres. La figure 11 illustre le contenu d'un container Docker avec ses différents composants.

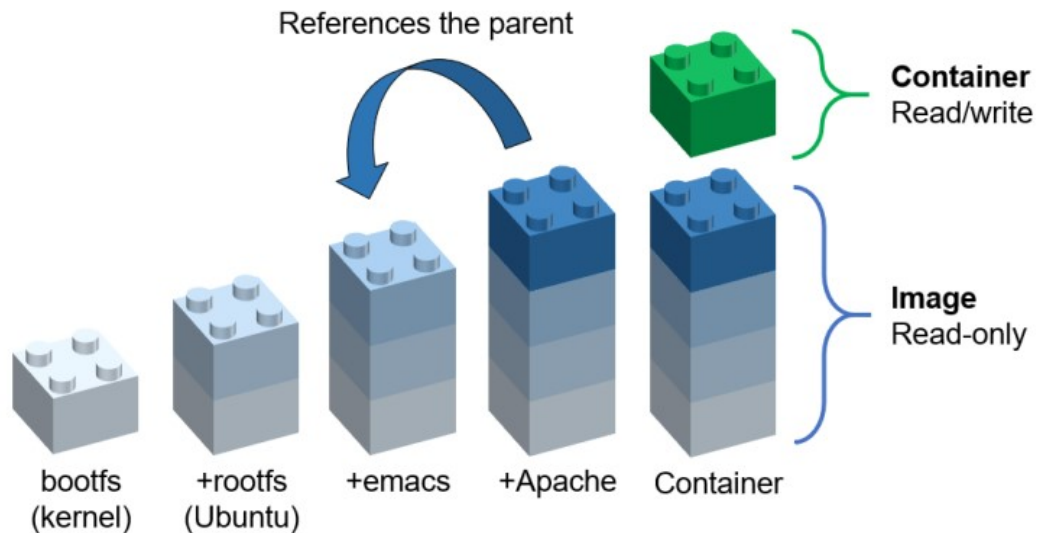


Figure 11 - Illustration du contenu d'un container Docker - Image de l'auteur

2.8.2. Compatibilité

En faisant tourner une application, des problèmes de compatibilité peuvent rapidement survenir. Une application peut utiliser plusieurs technologies qui doivent tourner sur plusieurs sortes de machines. Sans une technologie de container, la compatibilité entre tous les morceaux d'applications et toutes les machines doit être systématiquement vérifiée. Cela peut représenter une matrice infernale (figure 12).

Matrix from Hell





	Application Python 2.x	?	?	?	?	?
	Application Python 3.x	?	?	?	?	?
	Application C# dotnet core	?	?	?	?	?
	Application React	?	?	?	?	?
		Linux desktop	Macbook	Windows laptop	HTTP server	Cloud server

Figure 12 - Représentation de la matrice infernale - Image de l'auteur

Docker permet de résoudre le problème en utilisant des containers. Chaque application tourne dans son propre environnement en ayant le moins d'interaction possible avec les autres. Docker permet d'isoler chaque composant et assure la communication entre eux. Ainsi, il n'est plus nécessaire de s'inquiéter de la compatibilité. (Bashari Rad, Bhatti, & Ahmadi, 2017, p. 230).

2.8.3. Docker vs machine virtuelle

Docker fonctionne un peu comme une machine virtuelle [VM]. La principale différence entre un container Docker et une VM est son poids. Une VM contient une version complète d'un système d'exploitation et émule tous les composants physiques d'un ordinateur (figure 13). Docker se contente d'émuler les librairies nécessaires au fonctionnement de l'application ainsi que cette dernière (figure 14). Un container Docker est donc bien plus léger qu'une VM. (Bashari Rad, Bhatti, & Ahmadi, 2017, pp. 229-230).

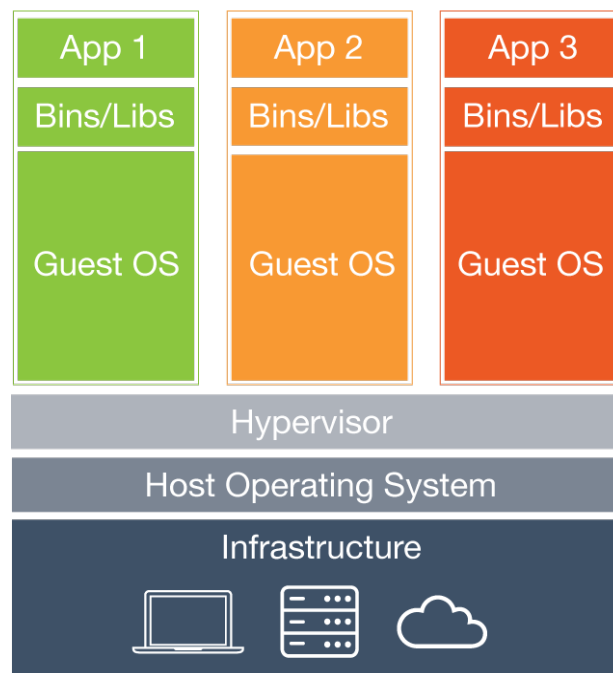


Figure 13 - Architecture d'une machine virtuelle (Bashari Rad, Bhatti, & Ahmadi, 2017, p. 229)

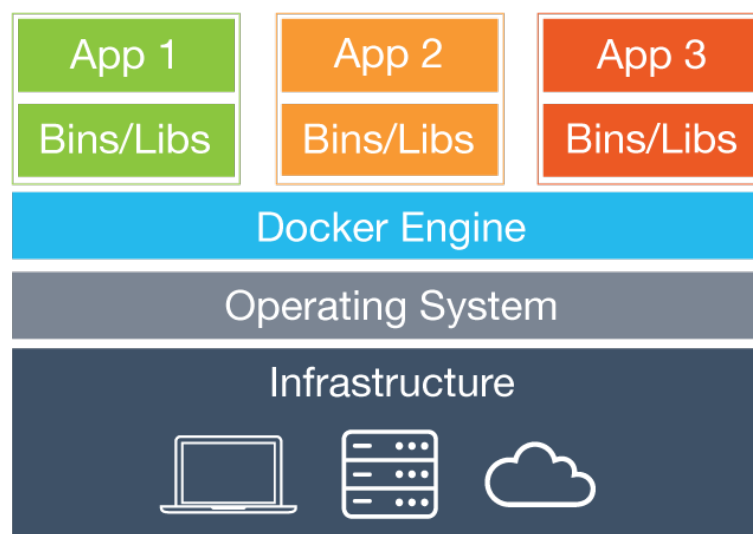


Figure 14 - Architecture d'un container Docker (Bashari Rad, Bhatti, & Ahmadi, 2017, p. 230)

2.8.4. Avantages de Docker

Les avantages principaux de Docker sont

- La vitesse : Vase T. explique que les containers étant très petits, ils sont créés très rapidement (cité dans Bashari Rad, Bhatti, & Ahmadi, 2017, p. 230).
- La portabilité : Vase T. indique que les applications construites à l'intérieur de containers Docker sont extrêmement portables. Elles peuvent être déplacées comme un seul élément sans que leurs performances soient réduites. (cité dans Bashari Rad, Bhatti, & Ahmadi, 2017, p. 230).
- L'évolutivité : Joy, A. M. explique qu'un container Docker a la capacité d'être déployé et exécuté sur plusieurs environnements différents. Des ajustements peuvent également être effectués facilement. (cité dans Bashari Rad, Bhatti, & Ahmadi, 2017, p. 230).
- La rapidité de livraison : Vase T. indique que le format d'un container Docker est standardisé. Ainsi, les programmeurs n'ont pas à s'inquiéter d'autre chose que de l'application à l'intérieur du container Docker. Ces containers pouvant fonctionner dans tous les environnements, ils sont déplaçables en production très facilement et rapidement. (cité dans Bashari Rad, Bhatti, & Ahmadi, 2017, pp. 230-231).
- La densité : Joy, A. M. explique que le container Docker n'utilise pas d'hyperviseur, contrairement aux VM. Pour cela, il est possible d'en faire tourner un plus grand nombre sur une seule machine. Il y a donc moins de gaspillage de ressources. (cité dans Bashari Rad, Bhatti, & Ahmadi, 2017, p. 231).

3. Choix technologique

Dans ce chapitre, tous les choix de technologie faits durant ce travail seront expliqués.

3.1. Langage de programmation

Pour le choix du langage de programmation, le mémoire de master de Mattias Levlin « DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte » a servi de base de comparaison entre les différents langages.

Dans la programmation web, les langages utilisés le plus souvent sont le HTML, le CSS et le Javascript. Le HTML est utilisé pour créer les éléments de la page web comme les titres ou les paragraphes de texte. Le CSS est un langage de design. Il permet de modifier l'aspect d'éléments HTML et de les placer sur une page. Le Javascript est utilisé pour toute la partie dynamique de l'application. (Levlin, 2020, p. 2).

Il existe de nombreux framework javascript pour développer des sites web. Pour ce projet, nous allons comparer les plus importants, React, Angular et Vue.

React est un framework créé par Facebook en 2013. Il est spécialisé dans l'affichage frontend. Il utilise un Document Object Model [DOM] virtuel et réduit au minimum les modifications sur la page. En effet, React rechargera seulement les éléments qui ont été modifiés par rapport au stade précédent de la page. Ces deux points permettent de gagner de la vitesse sur les rafraîchissements. (Levlin, 2020, pp. 44-46).

Angular existe en deux versions. AngularJS n'est presque plus mise à jour. Angular 2+ créé en 2016, est l'un des plus gros framework javascript. Il intègre de nombreuses fonctionnalités comme l'utilisation de TypeScript, un langage de programmation facilitant son intégration et sa sécurité. Concernant le maniement du DOM, Angular 2+ utilise des fonctions pour agir directement sur les éléments. Contrairement à React, les composants sont souvent séparés en deux fichiers, un .HTML et un .TS. Il est également connu pour sa courbe d'apprentissage rude. (Levlin, 2020, pp. 46-49).

Vue a été créé par un employé de Google, Evan You, en 2014. Le succès de Vue est entre autres dû au fait qu'il est facile à apprendre, qu'il est léger et que ses performances sont bonnes. Comme React, un DOM virtuel est utilisé pour interagir avec la page web. (Levlin, 2020, pp. 49-52).

Mattias Levlin a testé les différents framework et a tiré des conclusions visibles dans la figure 15 ci-dessous.

TECHNICAL METRICS	React v16.12.0	Vue v2.6.11	Angular v8.2.14	Svelte v3.20.0
Add 10,000 (ms)	30.96	25.36	52.75	31.26
Edit one (ms)	16.58	22.23	6.08	0.11
Edit 10,000 (ms)	17.86	20.64	896.76	885.03
Remove one (ms)	16.54	24.51	0.09	0.53
Remove 10,000 (ms)	7.39	33.33	23.83	22.97
Compilation (s)	3.96	3.07	8.70	1.61
Minified size (kb)	6.4	63.5	187.6	3.5
POPULARITY	React v16.12.0	Vue v2.6.11	Angular v8.2.14	Svelte v3.20.0
Documentation in number of languages	16	8	4	2
Positive interest in StateOfJs 2019 (%)	83.7	74.7	31.6	51.7
Number of Github stars, January 2020	142,850	159,091	29,756	56,789
OTHER METRICS	React v16.12.0	Vue v2.6.11	Angular v8.2.14	Svelte v3.20.0
Virtual DOM	Yes	Yes	No	No
TypeScript	No	No	Yes	Yes
Release date	2013	2014	2016	2016
OVERALL RATING	React v16.12.0	Vue v2.6.11	Angular v8.2.14	Svelte v3.20.0
Placement	1	2	4	3

Figure 15 - Résultats de la comparaison des différents frameworks javascript (Levlin, 2020, p. 70)

React est le plus efficace d'après les analyses faites par Mattias Levlin, De plus, ce framework est très bien adapté pour le développement de petites applications et est simple d'utilisation. Pour ces raisons, c'est React qui sera choisi pour développer le prototype de ce projet.

3.2. Visionneuse d'image, OpenSeadragon

L'outil de visionnage d'image utilisé dans ce projet est OpenSeadragon car cet outil répond à tous les besoins nécessaires pour ce projet et est déjà connu de l'équipe de recherche. Il a été proposé par le professeur responsable lors du premier entretien.

3.3. Plugin d'annotation

Pour récupérer des zones d'images, un plugin a dû être ajouté à l'outil OpenSeadragon. Plusieurs plugins existent pour annoter des images. Deux d'entre eux ont été retenus. Il s'agit d'Annotorious³ et d'OpenSeadragon-Annotations⁴ [Osd-Annotations]. Les critères de sélection ont été définis comme suit :

- L'entretien du code
- La documentation
- La facilité d'utilisation
- La compatibilité entre les navigateurs web

Concernant l'entretien du code, Annotorious est mis à jour très régulièrement. A la date du 11 juin 2021, la dernière mise à jour datait du 6 juin 2021. Osd-Annotations, quant à lui, n'a plus été mis à jour depuis le 5 septembre 2017, soit il y a presque quatre ans. La mise à jour régulière permettant de combler d'éventuelles failles de sécurité, Annotorious marque donc des points sur ce critère.

La documentation est de bonne taille pour Annotorious. Elle est constituée du site du plugin ainsi que de liens pour expliquer comment l'implémenter. Ce plugin existant également pour toutes sortes d'images, la qualité de la documentation est meilleure que celle d'un plugin créé uniquement pour OpenSeadragon. Celle d'Osd-Annotation n'est constituée que de la partie « readme » du code source sur GitHub. La documentation est donc bien plus grande sur Annotorious.

Pour utiliser Annotorious, il faut sélectionner la zone voulue en maintenant la touche « Maj » enfoncée. On peut créer des zones rectangulaires ou des polygones. Pour Osd-Annotations, des icônes sont ajoutées en bas à gauche d'OpenSeadragon. Une fois que l'on a cliqué sur le bouton, on peut dessiner ce que l'on veut sur l'image. Pour le projet, dessiner tout type de forme n'est pas nécessaire, les rectangles suffisent. Les deux plugins sont relativement faciles à utiliser et aucun ne se démarque sur ce point.

Annotorious est compatible sur tous les navigateurs classiques, soit Firefox, Chrome, Safari, Opera, Internet Explorer 9 et supérieurs et Edge. Osd-Annotations est également supporté par Firefox, Chrome, Safari, Opera, Internet Explorer 9 et supérieurs et Edge. Cependant, des problèmes de performance ont été constatés pour Edge et Internet Explorer (Emigre, 2017). Annotorious est donc légèrement meilleur sur ce point.

³ Site d'Annotorious : <https://recogito.github.io/annotorious/>

⁴ Site d'Osd-Annotations : <https://github.com/Emigre/openseadragon-annotations>

Comparaison Plugin d'annotation d'image, notes sur 5		
	Annotorious	OpenSeadragon-annotations
Entretien du code	5	1
Documentation	5	2
Facilité d'utilisation	4	4
Compatibilité	5	4
Total	19	11

Figure 16 - Comparaison des plugins d'annotation - Image de l'auteur

La figure 16 montre les notes obtenues par les deux plugin. Avec un résultat de 19 contre 11, c'est donc Annotorious qui a été retenu et implémenté dans le prototype.

4. Implémentation

Dans ce chapitre, les différentes parties de l'application web développée sont expliquées en détail. La figure 17 montre l'application lors de la première visite. Les différentes parties y sont encadrées. Il s'agit de la zone de gestion d'image (zone 1), de la zone d'affichage des résultats (zone 2) et de la zone de sélection des paramètres (zone 3). Les trois zones sont expliquées en détail dans les sous-chapitres suivants. Le code source du projet est disponible sur GitHub (<https://github.com/nicolasconstantin/CADeval>).



Figure 17 - Parties de l'application CADeval - Image de l'auteur

4.1. Sélecteur de paramètre

Dans cette zone (zone 3 sur la figure 17), les paramètres désirés peuvent être sélectionnés, soit l'image sur laquelle travailler, le type de modèle et le type de XAI. Les trois paramètres se présentent sous la forme d'un menu déroulant s'affichant sur la droite lorsque l'on clique dessus.

4.1.1. Menu Image

Le menu « Image » permet de choisir l'image sur laquelle travailler. Elles sont réparties en cinq centres, numérotés de zéro à quatre. Dans chacun d'eux, une centaine d'images sont disponibles. En cliquant sur un centre (figure 18), on peut afficher toutes les images disponibles dans celui-ci (figure 19). La croix permet de fermer le menu.

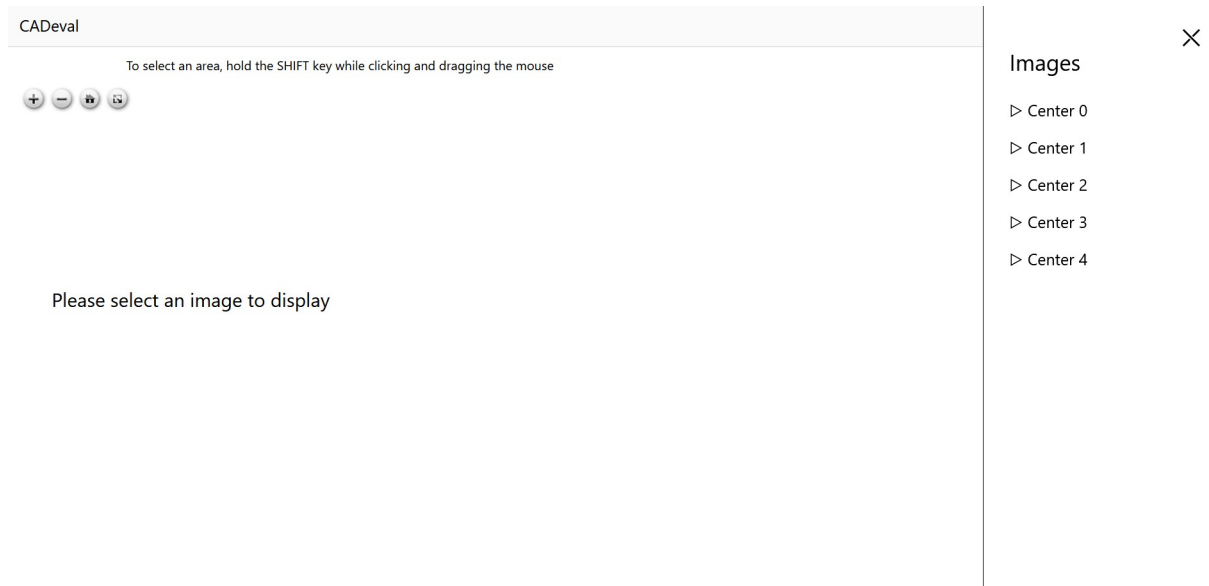


Figure 18 - Sélection des centres - Image de l'auteur

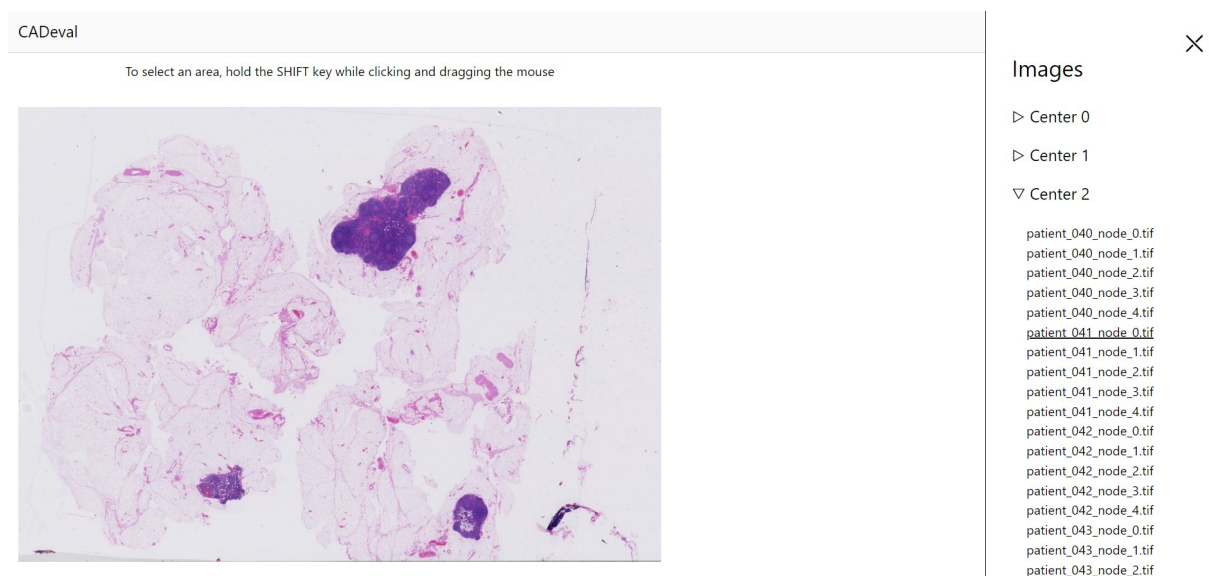


Figure 19 - Sélection des images - Image de l'auteur

4.1.2. Menu CNN

Le menu CNN (figure 20) permet de sélectionner le modèle que l'on souhaite utiliser. Trois choix sont possibles. Les spécificités de chaque modèle ne sont volontairement pas données dans l'interface. L'objectif est de ne pas influencer les utilisateurs lors des tests. Par défaut, le modèle 1 va être utilisé. Au moment du rendu de ce rapport, seul le modèle 1 est implémenté. Une fois que les deux autres modèles seront terminés par Mara Graziani, ils fonctionneront directement depuis l'interface.

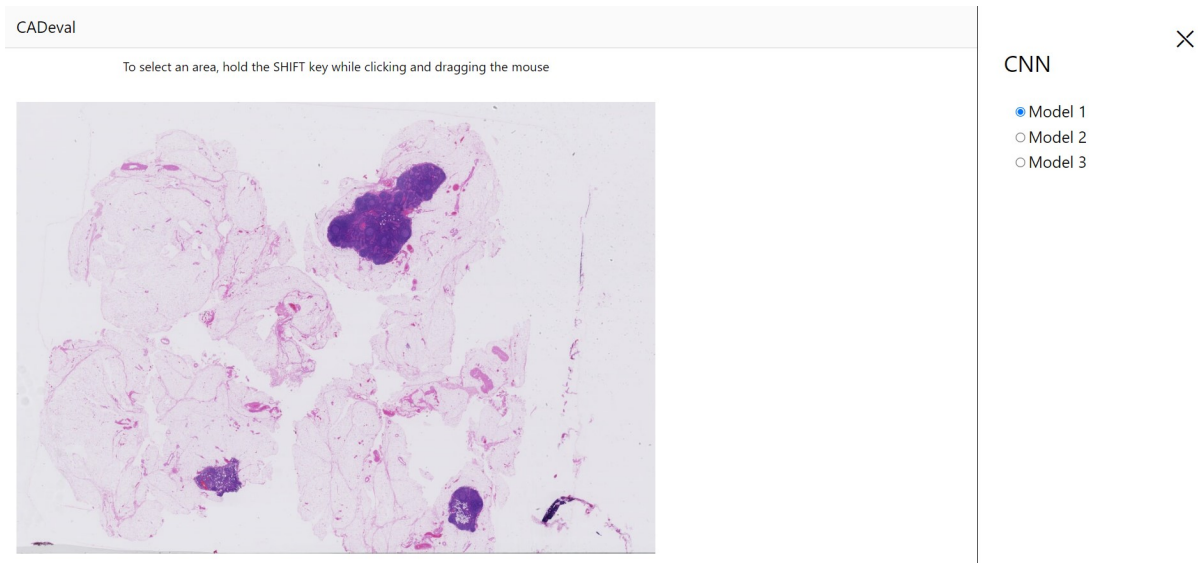


Figure 20 - Sélection du modèle CNN - Image de l'auteur

4.1.3. Menu XAI

Le menu XAI (figure 21) permet de sélectionner le type de XAI désiré. Trois choix sont possibles. Par défaut, aucun XAI n'est sélectionné. XAI 1 représente les XAI « Grad-CAM » et « LIME », XAI 2 représente les XAI « RCVs » et « Sharp-LIME ». L'explication des types de XAI sera donnée lors des tests utilisateurs. Lors du rendu de ce rapport, le XAI 2 n'est pas encore terminé. Il renvoie donc les données du XAI 1. Dès que Mara Graziani aura fini de gérer le XAI 2, l'interface sera adaptée par l'étudiant.

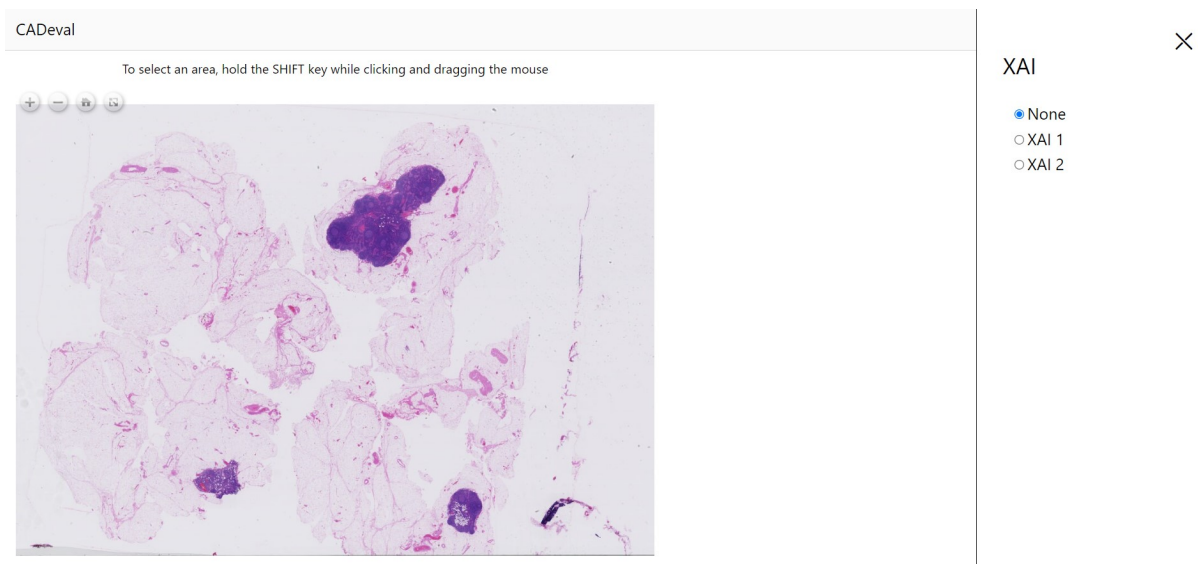


Figure 21 - Sélection du XAI - Image de l'auteur

4.2. Zone de gestion d'image

Une fois que l'on a choisi une image dans le sélecteur, elle s'affiche dans la zone de gestion d'image (zone 1 sur la figure 17). Si le site a déjà été utilisé, la dernière image sélectionnée est conservée dans le cache et s'affichera au démarrage.

4.2.1. Navigation

Il est possible de naviguer dans l'image grâce à la souris. La molette de celle-ci peut être utilisée pour zoomer ou dézoomer. Il est également possible de zoomer en cliquant n'importe où sur l'image. Les boutons en haut à gauche du lecteur permettent également de naviguer sur l'image. La figure 22 montre les boutons disponible dans le lecteur OpenSeadragon.



Figure 22 - Boutons du lecteur d'image - Image de l'auteur

Le bouton « + » tout à gauche sert à effectuer un zoom au centre de l'image. Le bouton « - » à sa droite sert à effectuer un dézoom. Le bouton avec une icône de maison sert à revenir à l'image dézoomée au maximum. Le dernier bouton, tout à droite, sert à passer le lecteur en plein écran. Il suffit de re cliquer dessus pour sortir du mode plein écran.

4.2.2. Sélectionner une zone

Comme le texte au-dessus de l'image nous le fait remarquer, il est possible de sélectionner une zone de l'image. Pour cela, il faut maintenir la touche « majuscule » du clavier, tout en cliquant et glissant avec la souris pour créer la zone voulue. Une fois la souris relâchée, la zone apparaît. Si une autre zone est créée, la première disparaît.

En cliquant sur une zone, la fenêtre de gestion de celle-ci apparaît (figure 23). Il est ensuite possible de la redimensionner, de la déplacer ou de la supprimer. Le bouton « Ok » valide les modifications de la zone, tandis que bouton « Cancel » les annule.

Pour déplacer la zone, il faut cliquer dessus et déplacer la souris en maintenant le clic.

Les cercles noirs aux quatre coins de la zone servent à la redimensionner.

Pour la supprimer, il suffit de cliquer sur l'icône de poubelle sur la gauche.

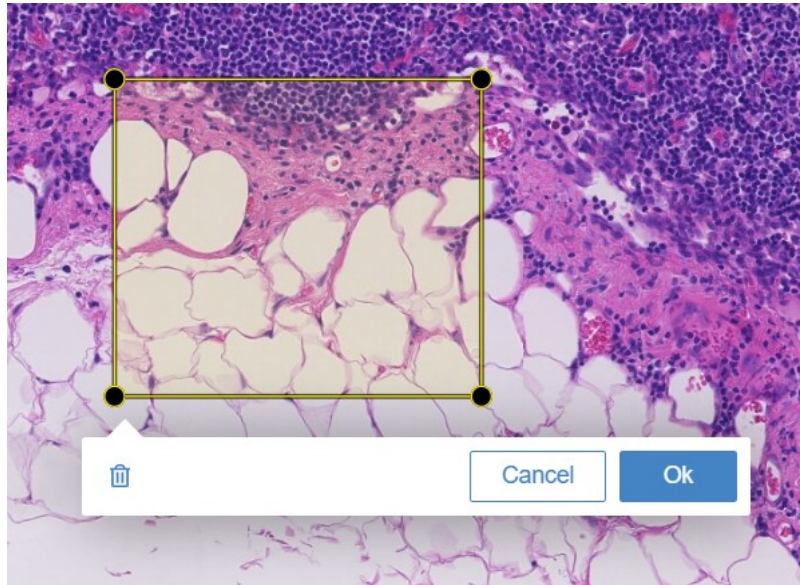


Figure 23 - Gestion d'une zone - Image de l'auteur

4.3. Zone de résultats

Au milieu de l'application (zone 2 sur la figure 17) se trouve la zone où vont s'afficher les caractéristiques de la zone choisie, les résultats du modèle CNN et le bouton pour lancer une requête vers ce dernier (figure 24).

4.3.1. Caractéristiques

En haut de la zone de résultats s'affichent les caractéristiques de la zone.

La localisation sur l'image globale s'affiche à l'aide de quatre coordonnées. Les deux premières représentent la localisation du point en haut à gauche de la zone, les deux dernières représentent celle du point en bas à droite. Pour chaque paire de nombres, le premier indique la distance en pixels le séparant du bord gauche de l'image et le deuxième indique la distance en pixels le séparant du haut de l'image.

La taille de la zone en pixels se trouve en dessous de sa localisation. La hauteur de la zone et sa largeur y sont indiquées. Cela permet notamment d'éviter de faire des zones trop petites ou trop grandes.

Des informations concernant l'image sélectionnée sont également affichées. On peut y voir le centre d'où provient l'image ainsi que son nom complet.

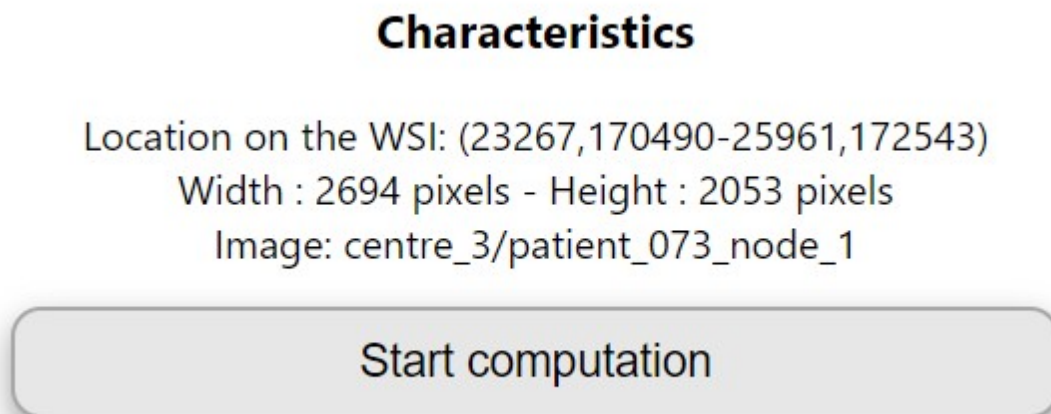


Figure 24 - Caractéristiques d'une image et bouton pour lancer le modèle - Image de l'auteur

4.3.2. Bouton pour lancer le modèle

Pour lancer le modèle sélectionné, il suffit de cliquer sur le bouton « Start computation » une fois que la zone choisie convient. Une requête https est alors transmise vers le modèle CNN pour lancer les calculs. Plusieurs informations lui sont envoyées :

- La localisation de la zone dans l'image
- Le centre sélectionné
- L'image sélectionnée
- Le modèle sélectionné
- Le type de XAI sélectionné

Les modèles de CNNs pouvant mettre longtemps à finir leurs calculs, un message s'affiche une fois le bouton cliqué (figure 25).

The computation has started, CNN model can take a moment to finish. Do not refresh the page please !

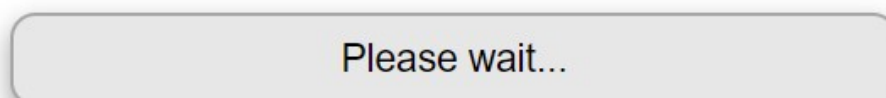


Figure 25 - Message indiquant que le modèle travaille - Image de l'auteur

Une fois que les résultats sont affichés, le bouton redevient cliquable pour lancer à nouveau les calculs pour une autre zone. Dans le cas où la page est rechargée, le bouton redevient cliquable et une nouvelle requête pourrait être envoyée alors que la précédente n'est pas encore terminée. Si tel est le cas, la nouvelle requête va être mise en attente par l'algorithme et résolue après celle en cours. Cela augmentera le temps d'attente.

4.3.3. Gestion des erreurs

Le modèle a besoin d'une image ni trop petite ni trop grande pour pouvoir travailler correctement. C'est pourquoi, si la zone fait moins que 200 pixels ou plus de 20 000 pixels en largeur ou en hauteur, le script ne va pas se lancer et un message va apparaître, demandant une zone de la bonne taille (figure 26).

Your area is too small or too big (min height and width 200 pixels and max height and width 20 000 pixels).

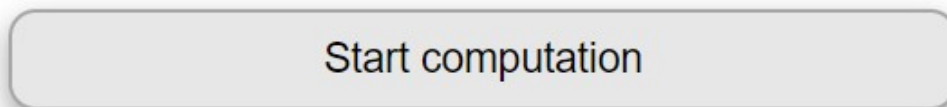


Figure 26 - Message d'erreur, la zone est trop grande ou trop petite - Image de l'auteur

Si le modèle n'a pas trouvé d'image valable pour faire ses calculs, le message suivant va s'afficher, demandant de sélectionner une autre zone (figure 27).

No processable images were found by the CNN model. Please, try to select another area.

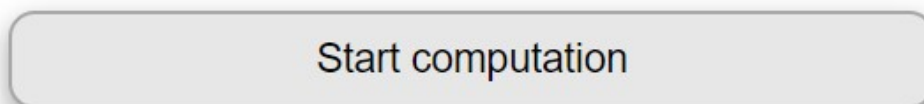


Figure 27 - Message d'erreur, la zone n'est pas valide pour le modèle - Image de l'auteur

Si une autre erreur survient durant l'exécution du modèle CNN, elle sera récupérée et affichée (figure 28).

An error occurs during the model processing: TypeError: Failed to fetch

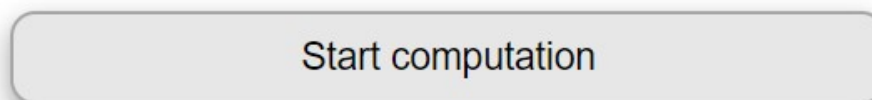


Figure 28 - Exemple d'autre erreur durant les calculs du CNN - Image de l'auteur

4.3.4. Résultats des calculs du CNN

Lorsque le modèle de CNN a terminé ses calculs, les images des résultats sont envoyées sur un serveur. Il renvoie ensuite en réponse à la requête le nom du dossier contenant les images, le nom des images et le résultat pour chacune d'elles. Les images et prédictions des résultats sont ensuite affichées en-dessous des caractéristiques.

Pour pouvoir communiquer avec le serveur hébergeant les scripts de DL, un tunnel sécurisé a été créé en utilisant ngrok⁵. Cet outil permet de créer facilement des tunnels sécurisés pour communiquer en Https sur le web.

Comme expliqué dans le chapitre 0 sur les CNNs, la zone sélectionnée par l'utilisateur est séparée aléatoirement en plusieurs morceaux par l'algorithme. Les morceaux les plus significatifs pour le résultat sont affichés dans l'ordre du plus impactant au moins impactant. À côté de ceux-ci, leurs XAI s'affichent s'il y en a. La prédiction donnée par le modèle est également indiquée tout à droite. La figure 29 montre les résultats sans XAI. La figure 30 quant à elle montre les résultats en demandant un XAI.

⁵ Site de ngrok : <https://ngrok.com/>

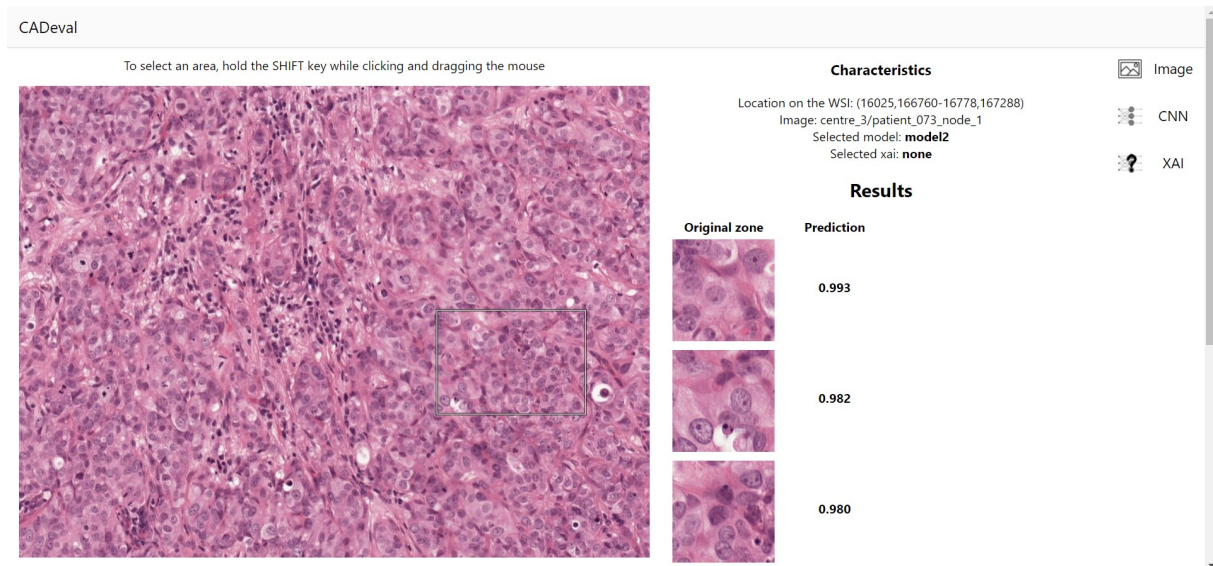


Figure 29 - Exemple de résultat sans XAI - Image de l'auteur

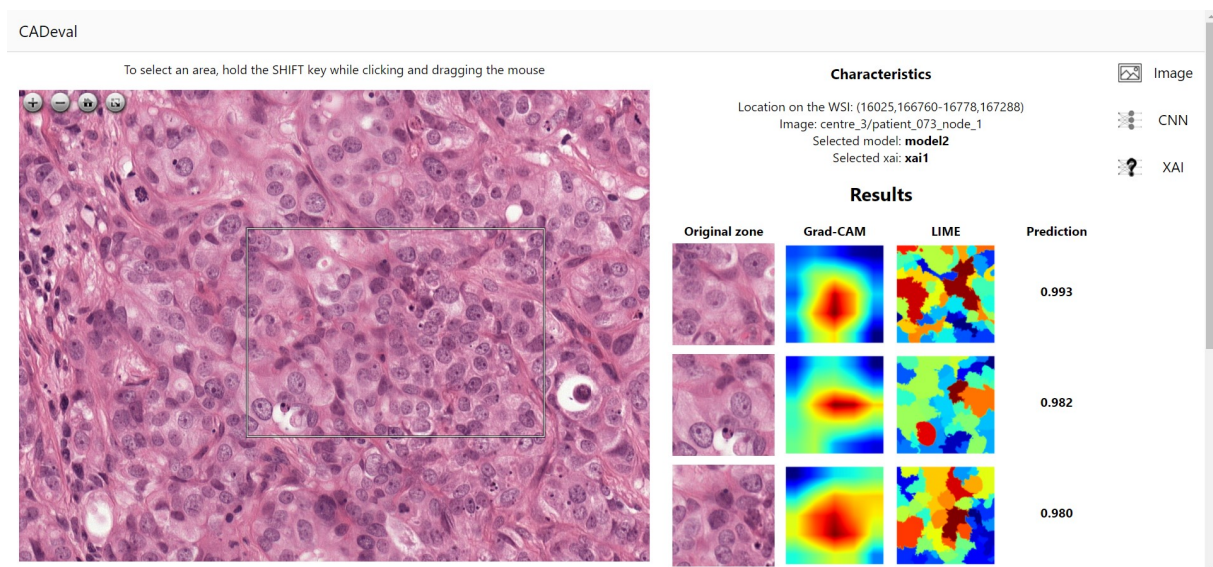


Figure 30 - Exemple de résultat avec XAI - Image de l'auteur

4.4. Déploiement

L'interface web a tout d'abord été développée localement. Une fois le stade permettant d'afficher les résultats atteint, les fichiers permettant de créer un container Docker ont été ajoutés. Pour les tests de déploiement, un serveur de la HES-SO a été mis à disposition de l'étudiant par Roger Schaer.

Une fois ce travail terminé, l'interface sera déployée dans un serveur interne de l'unité eSanté de l'IIG par Roger Schaer.

4.5. Autres services

Cette section décrit les autres services qui ont été utilisés durant ce travail. La figure 31 illustre comment chaque composant interagit avec l'interface web.

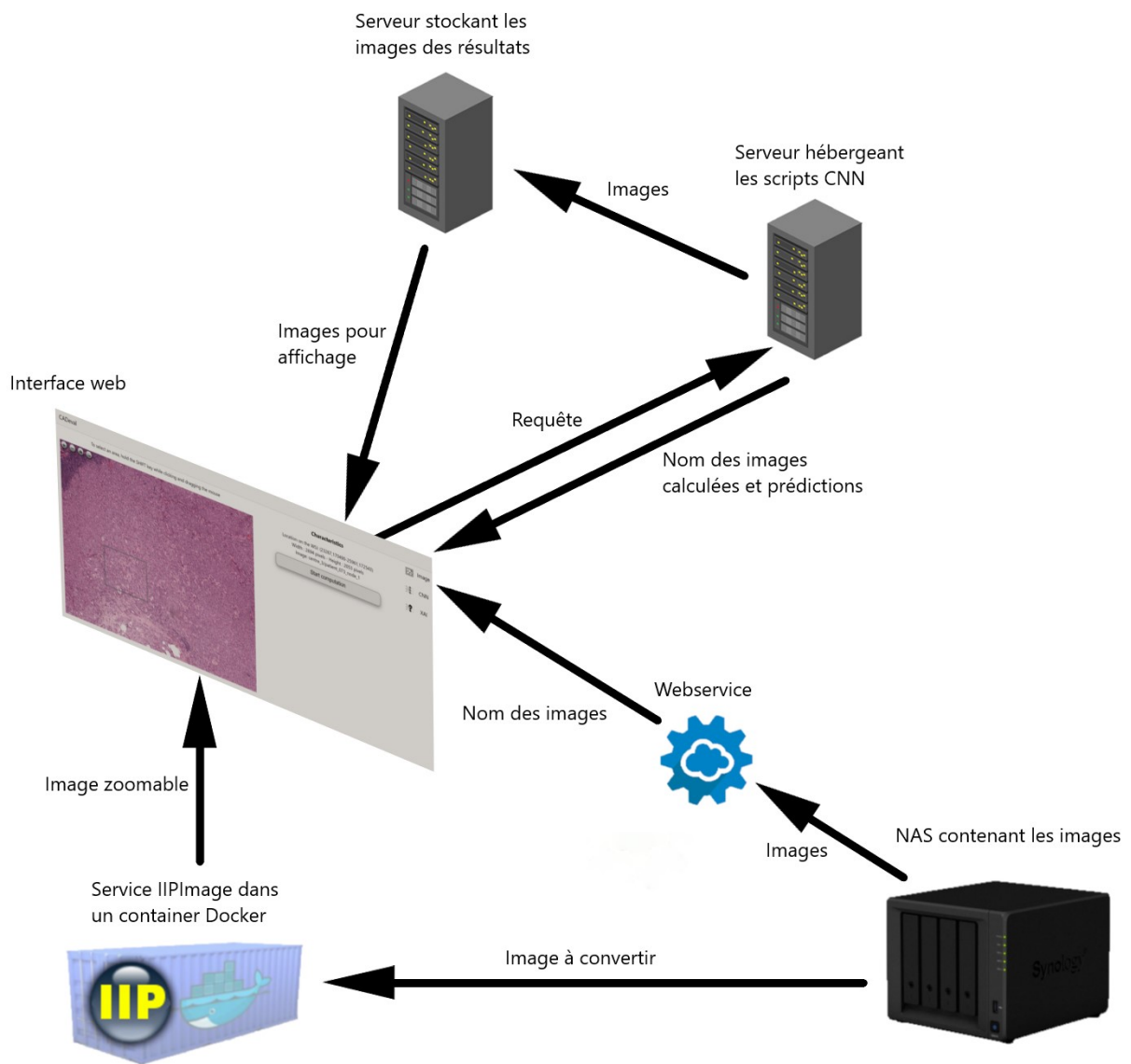


Figure 31 - Interaction entre les composants - Image de l'auteur

4.5.1. Récupération du chemin des images

Au début du projet, quelques images ont été téléchargées en local pour les premiers tests. Une copie de la totalité des images de la banque d'images de Camelyon17 (voir 2.2.1, banque d'images) était déjà présente sur un NAS de l'IIG. Une fois l'application déployée sur le serveur de test, ces

images ont pu être utilisées. Pour cela, Roger Schaer a développé un petit webservice⁶ qui renvoie le nom de toutes les images contenues sur le serveur pour les afficher dans le menu « image ». Comme pour les scripts de CNN, il est disponible sur le web grâce à un pont ngrok.

4.5.2. IIPImage

Pour transformer les images d'un format BigTIFF à un format d'image zoomable, le système de serveur IIPImage a été utilisé⁷. Ce système est fait pour visualiser en continu des images à haute résolution sur le web (Pillay, 2021).

Un container Docker contenant le système IIPImage a tout d'abord été installé en local pour les tests et les premiers affichages d'images. Par la suite, une version de ce container a été installée sur un serveur de l'IIG afin de pouvoir l'atteindre en tout temps et de n'importe où de manière sécurisée. Les noms des images sont renvoyés en format JSON et traités dans l'application pour pouvoir être choisis par l'utilisateur.

4.5.3. Scripts python CNN

Mara Graziani a développé des scripts en python qui prédisent les zones cancéreuses en utilisant un modèle de CNN. Dès qu'il reçoit une requête, l'algorithme va prendre la zone d'image sélectionnée par l'utilisateur et la découper en plusieurs morceaux. Ceux-ci vont ensuite être analysés par le CNN. Une fois ce travail terminé, les images originales seront ajoutées dans un dossier créé sur un serveur de l'IIG. Ensuite, les éventuels XAI sélectionnés par l'utilisateur vont être calculés. Ces derniers seront alors ajoutés au même dossier que les résultats.

Une fois l'exécution terminée, le script renvoie une réponse à la requête Https. Celle-ci est composée d'un objet JSON contenant le nom du dossier sur le serveur ainsi que deux tableaux. Le premier contient les noms des images les plus pertinentes à afficher, le deuxième les prédictions de l'algorithme concernant les images du premier tableau. L'objet est ensuite traité par l'application et les résultats y sont affichés.

⁶ L'URL du webservice est le suivant : <https://cb95959001d2.ngrok.io>

⁷ Lien GitHub de l'IIPImage : <https://github.com/ruven/iipsrv>

5. Résultats

Ce chapitre regroupe les résultats de tous les tests pratiqués sur l'application. Des tests de latence ont été effectués afin d'optimiser la vitesse des algorithmes de DL. Des tests utilisateurs seront réalisés dès que les XAI 2 seront gérés par les scripts.

5.1. Tests de latence

Les modèles de CNNs prennent du temps pour calculer tout ce dont ils ont besoin pour faire une prédiction. Des tests ont été effectués afin de réduire au minimum le temps d'attente de la réponse d'un modèle. Les résultats de ces tests sont expliqués dans cette partie.

L'algorithme a été lancé avec des zones de différentes tailles pour observer la variation des temps d'attente. Les mêmes zones ont été testées avec et sans XAI ainsi que sur le CPU et le GPU. Selon la taille de la zone, le modèle de CNN va la découper en un nombre d'un à 20 morceaux. La limite de 20 morceaux a été introduite manuellement dans les scripts. Les tests ont été effectués de manière à avoir au minimum un test par groupe de morceaux. L'image se fait découper en 20 morceaux à partir d'une taille d'environ 1 000 pixels par 1 250 pixels. La zone pouvant aller jusqu'à 20 000 pixels par 20 000 pixels, des tests ont également été effectués pour des zones carrées avec un côté de 2 000 pixels, 3 000 pixels, 4 000 pixels, 5 000 pixels, 10 000 pixels, 15 000 pixels et 20 000 pixels, pour voir si la taille de la zone influence beaucoup le temps de latence. Les tests ont été effectués sur un serveur de l'IIG. Celui-ci possède un CPU de 56 cœurs, 1TB de RAM et un GPU de type NVIDIA v100.

Pour chaque test, cinq temps ont été calculés :

- Le temps d'extraction, correspondant à la durée nécessaire pour extraire les fragments d'images à partir de la zone sélectionnée.
- Le temps de prétraitement, correspondant à la durée nécessaire pour normaliser les couleurs des morceaux d'images. C'est une étape indispensable pour le bon fonctionnement du CNN.
- Le temps d'inférence, correspondant à la durée nécessaire pour calculer le résultat à partir des morceaux d'images.
- Le temps XAI, correspondant à la durée nécessaire pour calculer les XAI choisis.
- Le temps total, étant l'addition des quatre temps précédents, c'est le temps total d'exécution du modèle.

5.1.1. Tests sur le GPU

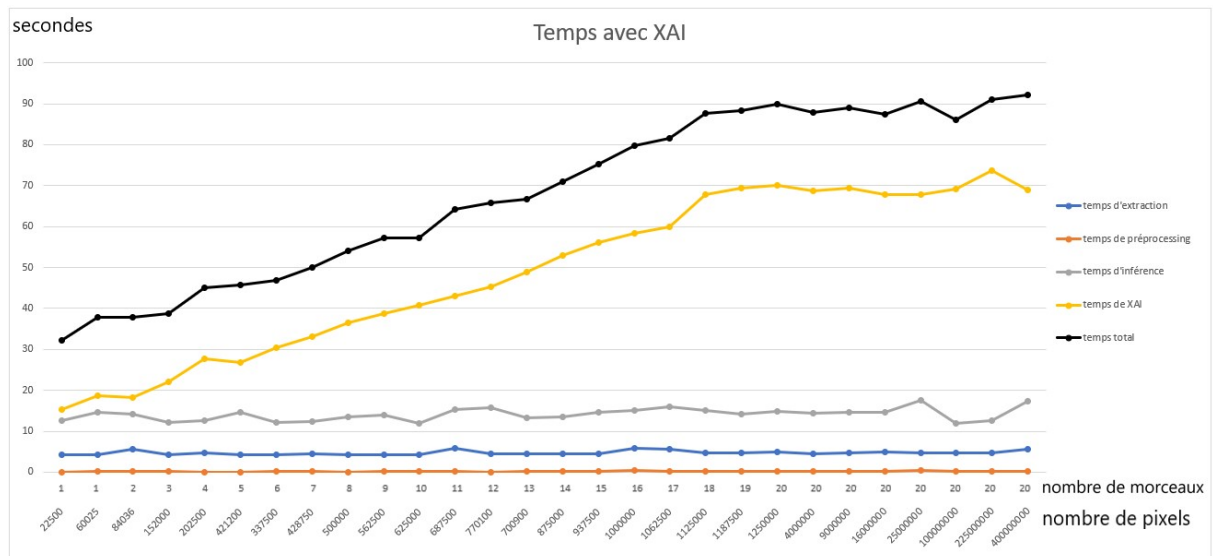


Figure 32 - Synthèse des tests de latence sur GPU avec XAI - Image de l'auteur

La figure 32 montre l'évolution des différents temps en fonction du nombre de morceaux d'images extraits par le modèle ainsi que du nombre de pixels de la zone en demandant un XAI. On constate que peu importe la taille de la zone ou son nombre total de pixels, les temps d'extraction, de prétraitement et d'inférence ne varient presque pas. Le temps de calcul de XAI, en revanche, augmente clairement en rapport avec le nombre d'images. On constate qu'une fois à 20 images, il se stabilise. Le nombre de pixels n'affectant pas significativement le temps de XAI, on constate que seul le nombre de morceaux d'image calculés influe sur le temps d'exécution du modèle. Le temps de XAI étant le seul à ne pas être régulier, le temps total suit donc bien son évolution. Pour obtenir un résultat avec cette configuration, il faut attendre environ entre 30 et 90 secondes. Cette durée dépend du nombre d'images extraites par le modèle.

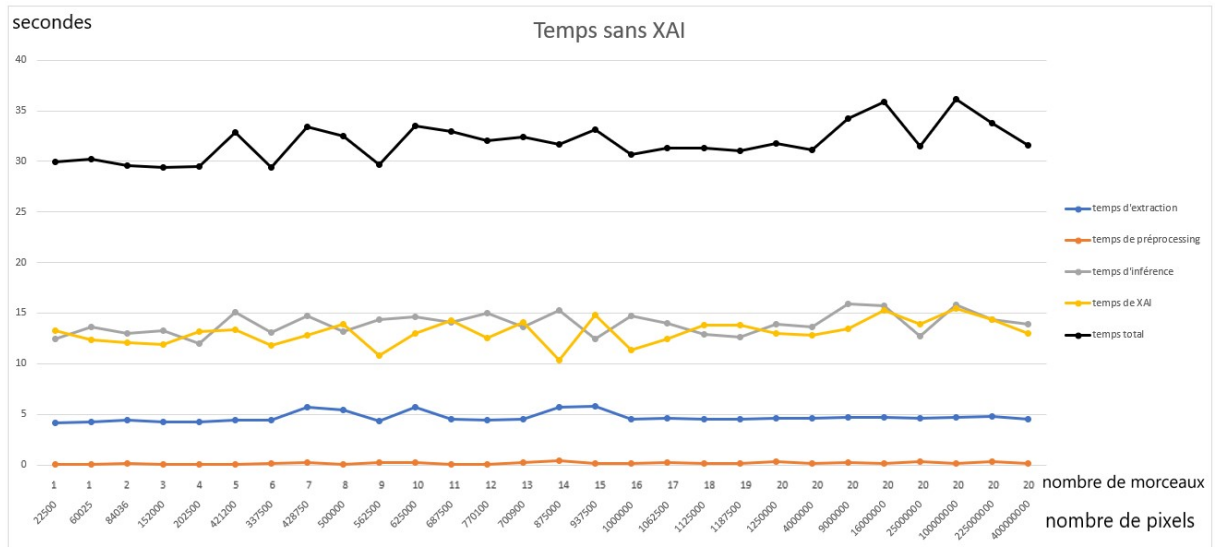


Figure 33 - Synthèse des tests de latence sur GPU sans XAI - Image de l'auteur

La figure 33 montre l'évolution des différents temps en fonction du nombre de morceaux d'images extraits par le modèle ainsi que du nombre de pixels de la zone en ne demandant pas de XAI. Contrairement aux tests précédents, l'algorithme ne fait que vérifier qu'aucun XAI n'a été sélectionné, c'est pourquoi ce temps n'augmente pas avec le nombre d'images. Tous les temps étant réguliers, peu importe la taille de la zone ou le nombre de morceaux d'images à analyser, le temps total est, lui aussi, régulier. Sans XAI demandé, le temps nécessaire à l'exécution totale du script est d'environ 30 à 36 secondes dans cette configuration.

5.1.2. Tests sur le CPU

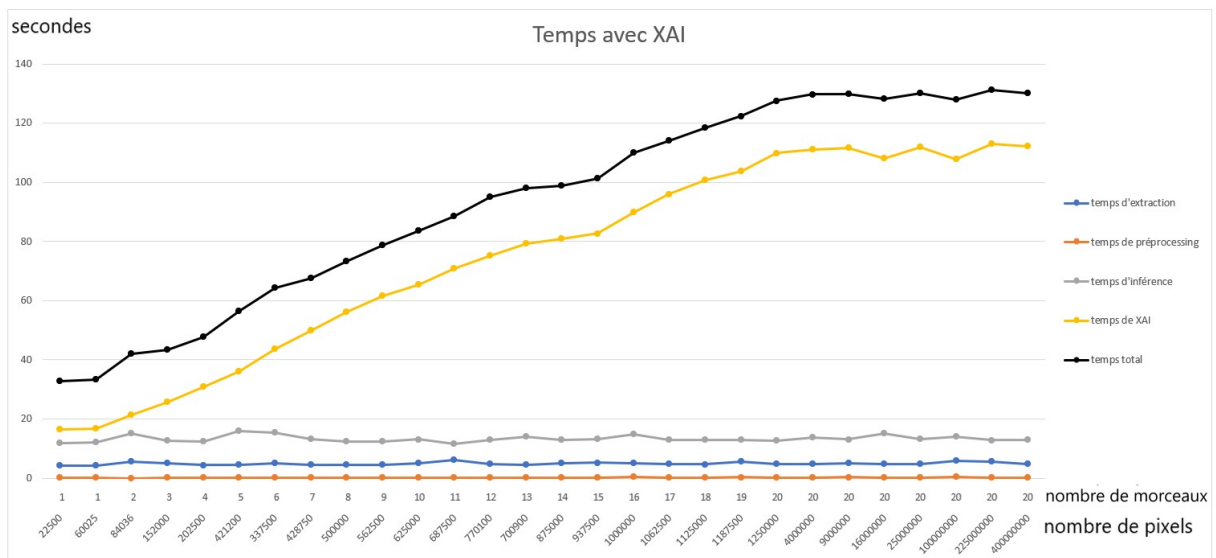


Figure 34 - Synthèse des tests de latence sur CPU avec XAI - Image de l'auteur

La figure 34 montre l'évolution des différents temps en fonction du nombre de morceaux d'images extraits par le modèle ainsi que du nombre de pixels de la zone en demandant un XAI. On constate que l'aspect de la courbe est le même qu'avec le GPU. Les temps d'extraction, de prétraitement et d'inférence sont réguliers. Le temps XAI, quant à lui, augmente en fonction du nombre de morceaux d'image défini par l'algorithme. Le temps total suit donc la courbe du temps XAI. On constate que le modèle est plus lent pour calculer les XAI avec le CPU qu'avec le GPU. Le CPU met jusqu'à presque 50% de temps en plus pour les calculer (figure 35). Dans cette configuration, le temps d'exécution varie entre environ 38 et 132 secondes, dépendant du nombre de zones extraites par le modèle.

type de tests	CPU	GPU
largeur de la zone	2000	2000
longueur de la zone	2000	2000
nbre d'image	20	20
temps d'extraction	4,74	4,5
temps de préprocessing	0,24	0,14
temps d'inférence	13,7	14,5
temps de XAI	110,99	68,63
temps total	129,67	87,77
pourcentage	147,74	100

Figure 35 - Comparaison directe entre un test sur CPU et sur GPU - Image de l'auteur

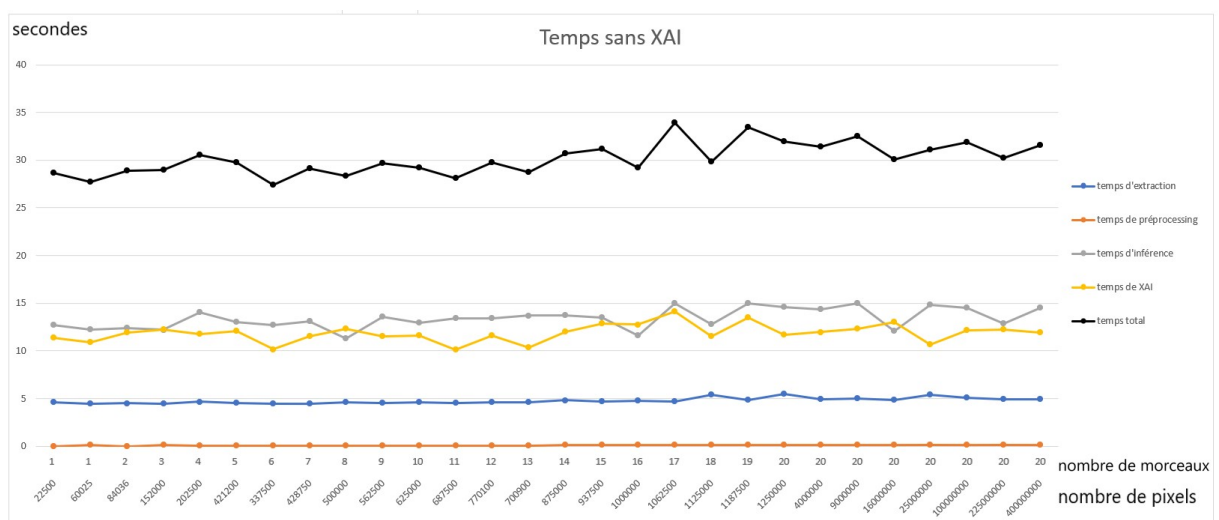


Figure 36 - Synthèse des tests de latence sur CPU sans XAI - Image de l'auteur

La figure 36 montre l'évolution des différents temps en fonction du nombre de morceaux d'images extraits par le modèle ainsi que du nombre de pixels de la zone en ne demandant pas de XAI. Comme pour les tests sans XAI sur le GPU, tous les temps sont réguliers. On constate également

que le temps d'exécution est très similaire sur le CPU et sur le GPU. Il varie entre environ 26 et 35 secondes dans cette configuration.

5.1.3. Synthèse

Deux types de tests ont été effectués, les tests avec et sans XAI. Ils ont les deux été réalisés sur GPU et sur CPU. La taille des zones testées est similaire pour tous les types de tests. Les résultats complets des tests sont disponibles dans l'annexe I.

Concernant les tests sans XAI, aucun changement significatif n'ayant été remarqué, l'utilisation d'un CPU ou un GPU est donc équivalente.

En revanche, pour les tests demandant un XAI, le GPU est nettement plus performant. Les temps d'extraction, de prétraitement et d'inférence ne changent pas entre le GPU et le CPU. Le temps XAI, quant à lui, prend jusqu'à 50% de temps supplémentaire pour être calculé. Pour une exécution de petite zone, soit une image séparée en un ou deux morceaux, le temps est similaire. Cependant, dès que le nombre d'images évaluées par le modèle augmente, le temps XAI augmente beaucoup plus rapidement si le CPU est utilisé. Pour réaliser cette tâche, il est donc plus optimisé d'utiliser la puissance de calcul du GPU. Ce résultat est logique car même si le script tourne sur le CPU, le GPU est utilisé pour calculer certaines choses concernant les XAI, par exemple, pour calculer à nouveau certains paramètres comme l'inférence dans le Grad-CAM, ou encore pour calculer le modèle de substitution dans le LIME.

5.2. Évaluations utilisateurs

Il était initialement prévu que les résultats des tests utilisateurs figurent dans ce rapport, mais les scripts ont pris plus de temps que prévu à être terminés. Ces évaluations seront donc réalisées dès qu'il sera possible de faire des requêtes avec le deuxième type de XAI.

Les évaluations vont être envoyées à des pathologistes. Tout d'abord, une vidéo explicative de l'outil leur sera transmise. Ils vont ensuite devoir réaliser un scénario prévu à l'avance. Dans celui-ci, ils testeront toutes les parties de l'interface. Après les tests, un questionnaire leur sera remis et ils donneront leur avis sur l'outil, leurs éventuelles propositions d'amélioration et leur ressenti durant l'expérience. Ces résultats seront récupérés et analysés dans le but de perfectionner le prototype développé par l'étudiant.

6. Discussion

Ce chapitre comprend des discussions sur ce qui a été fait dans ce projet et sur ce qui pourrait être amélioré dans le futur.

6.1. Améliorations possibles

Dans cette partie, plusieurs idées d'amélioration de l'outil vont être proposées. À la fin d'un développement, des idées de ce qui aurait pu être amélioré ou optimisé apparaissent généralement. Après les tests utilisateurs, d'autres idées d'améliorations vont certainement être suggérées.

6.1.1. Explications plus intuitives

Une idée pour rendre les explications plus intuitives serait de fournir la possibilité d'afficher les XAI directement sur l'image originale dans les résultats. Actuellement, les morceaux d'images sont affichés les uns en dessous des autres avec leurs XAI affichés à côté d'eux. L'idée serait d'afficher la totalité de la zone dans les résultats avec un XAI en transparence au-dessus, par exemple la carte thermique. Les prédictions pourraient également être affichées sur ces dernières. Les résultats actuels ne seraient pas forcément retirés. La figure 37 montre un exemple de rendu pour cette amélioration. Il est également possible de mettre un lien à disposition renvoyant l'image avec la carte thermique au-dessus au lieu de l'afficher directement dans les résultats.

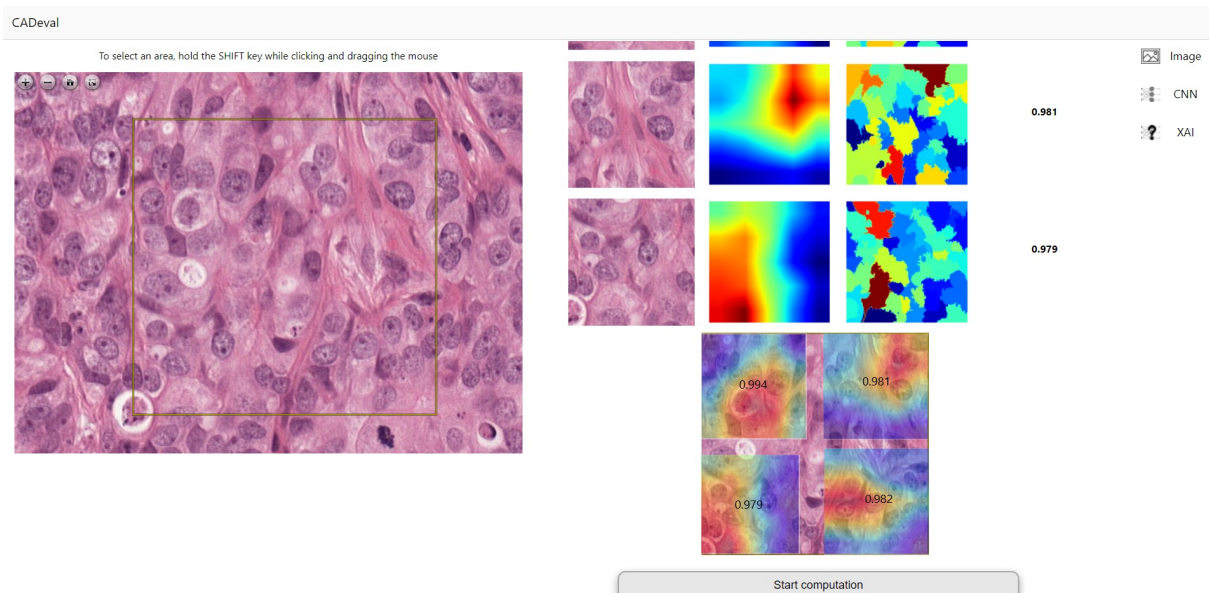


Figure 37 - Prototype de l'affichage de la carte thermique en dessous des résultats actuels - Image de l'auteur

Pour fabriquer l'image avec la carte thermique et la prédiction, une API devrait être créée, récupérant la zone de base, les cartes thermiques et les prédictions, et renvoyant l'image à afficher.

6.1.2. Explication des modèles

Il pourrait être intéressant de créer une page pour expliquer la différence entre les modèles. Actuellement, la possibilité de sélectionner différents modèles existe, mais aucune explication supplémentaire n'est donnée.

6.2. Rétrospective personnelle sur le travail

Le développement de l'interface s'est très bien déroulé. Toutefois, si je devais recommencer le travail maintenant, je gagnerais du temps sur deux points en particulier.

Le premier point est la transformation d'image du format BigTIFF en format zoomable. Cette étape m'a pris beaucoup plus de temps que ce que j'avais prévu initialement. J'ai fait de nombreuses recherches et installé plusieurs logiciels qui n'ont pas fonctionné. De plus, je développe sur un système Windows et la majorité des logiciels et des solutions de transformation d'images n'étaient compatibles qu'avec un système Mac ou Linux, ou disponibles uniquement dans des langages que je ne connais pas, tel que python.

Le second point est l'utilisation du plugin d'image. J'ai passé beaucoup de temps à réussir à le faire fonctionner malgré une documentation bien faite. Le plus problématique a été l'intégration du plugin à une application en React. La documentation propose des implémentations dans une page html classique avec du Javascript. Cependant, React a une structure un peu différente. J'ai donc mis du temps avant d'arriver à faire fonctionner correctement le plugin.

Un autre aspect qui aurait pu être amélioré est la réalisation des scripts. Mara Graziani a commencé à les créer à partir d'environ la moitié du travail, car au début, nous manquions d'informations. Avec le recul, nous aurions pu commencer les scripts en même temps que le développement de l'interface web. Nous les aurions donc probablement terminés avant le rendu de ce rapport.

7. Conclusion

Une interface web a été développée en React. Elle offre presque toutes les possibilités désirées au début de ce travail :

- Il est possible de sélectionner une image depuis un NAS, de zoomer et de naviguer dans celle-ci.
- Il est également possible de sélectionner une zone dans l'image.
- La zone peut être modifiée ou déplacée.
- Il est possible de choisir si la requête sera lancée avec ou sans XAI.
- Pour l'instant, un seul type de XAI est implémenté. L'interface est prête à intégrer le deuxième dès que Mara Graziani l'aura terminé.
- Il est possible de choisir trois modèles différents pour lancer une requête. Cependant, cette fonction renvoie toujours le même type de modèle pour l'instant. Les deux autres modèles n'ont pas encore été développés. Mara Graziani terminera leur développement après le rendu de ce rapport.
- Les caractéristiques de la zone sont affichées.
- Une requête peut être facilement lancée grâce à un bouton.
- Le résultat de la requête est affiché dès qu'il est disponible.
- Plusieurs requêtes peuvent être lancées à la suite.

Concernant les tests qui ont été réalisés, il a été constaté que dès qu'un XAI est choisi, il est préférable de faire tourner l'algorithme de DL sur le GPU de l'IIG. Si aucun XAI n'est souhaité, le CPU fonctionne autant rapidement que le GPU.

Les scripts pour les algorithmes de DL n'ont pas pu être terminés assez vite pour avoir le temps de faire les tests utilisateurs avant le rendu de ce rapport. C'est pourquoi leurs résultats ne sont pas présents dans ce document. Cependant, ils seront réalisés dès que possible.

7.1. Conclusion personnelle

J'ai pu apprendre beaucoup de choses que j'ignorais sur le domaine de la médecine. J'ai eu accès à des documents intéressants et j'ai apprécié pouvoir poser des questions à plusieurs personnes de l'unité eSanté de l'IIG.

Je pense m'orienter dans le futur vers des domaines touchant à l'IA. Ce travail m'a permis de découvrir ce monde plus précisément. Je n'en ai pas directement développé, mais j'ai pu utiliser des algorithmes de DL dans ce projet. J'ai également pu obtenir de bonnes sources pour me documenter sur le domaine.

Ce projet m'a également permis de mieux connaître certains aspects vus en cours de manière théorique. Il m'a permis de faire du déploiement sur serveur ainsi que d'utiliser plus en profondeur des outils tels que Docker. A cause du covid, j'ai surtout travaillé depuis chez moi. Malgré cela, j'ai pu me rendre plusieurs fois au Technopôle, ce qui m'a permis d'expérimenter le fait de travailler avec des professionnels, ce que j'ai particulièrement apprécié.

Pour toutes ces raisons, j'ai trouvé ce projet passionnant.

RÉFÉRENCES

- Amini, A. (2021a, Février 19). *MIT 6.S191: Convolutional Neural Networks*. Consulté le Juin 15, 2021, sur MIT 6.S191 Introduction to deep learning: https://www.youtube.com/watch?v=AjtX1N_VT9E&ab_channel=AlexanderAmini
- Amini, A. (2021b, Février 5). *MIT Introduction to Deep Learning | 6.S191*. Consulté le Juin 15, 2021, sur MIT 6.S191 Introduction to Deep Learning: https://www.youtube.com/watch?v=5tvmMX8r_OM&list=PLtBw6njQRU-rwp5__7C0oIVt26ZgjG9NI&index=2
- Bashari Rad, B., Bhatti, H. J., & Ahmadi, M. (2017, Mars). An Introduction to Docker and Analysis of its Performance. *International Journal of Computer Science and Network Security*, 17(3), 228-235.
- Beck, A. (2019, Juin 19). Artificial intelligence for pathology: From discovery to AI-powered companion diagnostics. Cambridge, Massachusetts, Etats-Unis. Consulté le Mai 10, 2021, sur https://www.youtube.com/watch?v=3NiFjm_SiFs&t=497s
- Cian, D., van Gemert, J., & Lengyel, A. (2020, Août 4). *Evaluating the performance of the LIME and Grad-CAM explanation methods on a LEGO multi-label image classification task*. Récupéré sur arxiv: <https://arxiv.org/abs/2008.01584>
- Emigre. (2017). *openseadragon-annotations*. Consulté le mai 31, 2021, sur github: <https://github.com/Emigre/openseadragon-annotations>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press. Récupéré sur <http://www.deeplearningbook.org>
- Graziani, M., Andrearczyk, V., Marchand-Maillet, S., & Müller, H. (2020, Août). Concept attribution: Explaining CNN decisions to physicians. *Computers in Biology and Medicine*, 123(103865). Récupéré sur <https://www.sciencedirect.com/science/article/pii/S0010482520302225>
- Graziani, M., de Sousa, I., Vellasco, M. M., Costa da Silva, E., Müller, H., & Andrearczyk, V. (2021, Octobre). Sharpening Local Interpretable Model-agnostic Explanations for Histopathology: Improved Understandability and Reliability. *To appear in the Proceedings of the 24th International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI 2021*. Strasbourg, France. Consulté le Juillet 6, 2021
- Graziani, M., Eggel, I., Deligand, F., Bobák, M., Andrearczyk, V., & Müller, H. (2020). *Breast Histopathology with High-Performance Computing and Deep Learning*. Récupéré sur

Computing and Informatics:
http://www.cai2.sk/ojs/index.php/cai/article/download/2020_4_780/1048

Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2019, Février 1). *Metrics for Explainable AI: Challenges and Prospects*. Récupéré sur arxiv: <https://arxiv.org/abs/1812.04608>

Levlin, M. (2020, Mai 28). *DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue, and Svelte*. Retrieved Juin 16, 2021, from Åbo Akademi University: <https://www.doria.fi/handle/10024/177433>

Markonis, D., Holzer, M., Baroz, F., De Castaneda, R. L., Boyer, C., Langs, G., & Müller, H. (2015, Octobre). User-oriented evaluation of a medical image retrieval system for radiologists. *International journal of medical informatics*, 84(10), 774-783.

Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill. Récupéré sur <http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/mlbook.html>

Mozilla Developer Network. (2021, Juin 15). *Référence du DOM*. Consulté le Juin 15, 2021, sur Mozilla Developer Network (MDN): https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model

OpenSeadragon. (2015). *OpenSeadragon*. Consulté le Mai 10, 2021, sur OpenSeadragon: <https://openseadragon.github.io/>

Parlement européen. (2016, Avril 27). *Règlement Général sur la Protection des Données*. Consulté le Juin 30, 2021, sur EUR-LEX: <https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX%3A32016R0679>

Petrack, N., Sahiner, B., Armato 3rd, S. G., Bert, A., Correale, L., Delsanto, S., . . . Chan, H.-P. (2013, Août). Evaluation of computer-aided detection and diagnosis systems. *Medical Physics*, 40(8). doi:10.1118/1.4816310

Pillay, R. (2021). *About*. Consulté le Juillet 19, 2021, sur IIPImage: <https://iipimage.sourceforge.io/contact/>

Schaer, R., Otàlora, S., Jimenez-del-Toro, O., Atzori, M., & Müller, H. (2019, July 1). Deep Learning-Based Retrieval System for Gigapixel Histopathology Cases and the Open Access Literature. *Journal of Pathology Informatics*. doi:10.4103/jpi.jpi_88_18

Seif, G. (2018, Avril 4). *Deep Learning vs Classical Machine Learning*. Consulté le Juillet 14, 2021, sur towards data science: <https://towardsdatascience.com/deep-learning-vs-classical-machine-learning-9a42c6d48aa>

Tonekaboni, S., Joshi, S., McCradden, M. D., & Goldenberg, A. (2019, Août 7). *What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use*. Récupéré sur arxiv: <https://arxiv.org/abs/1905.05134>

van Ginneken, B., Kerkstra, S., & Meakin, J. (2012a). *Camelyon17 - Background*. Consulté le Mai 19, 2021, sur Grand Challenge: <https://camelyon17.grand-challenge.org/Background/>

van Ginneken, B., Kerkstra, S., & Meakin, J. (2012b). *Camelyon17 - Home*. Consulté le Mai 19, 2021, sur Grand Challenge: <https://camelyon17.grand-challenge.org/Home/>

ANNEXES I : Résultats complets des tests de latence

Tests sur le GPU, centre 3, patient 73, nœud 1 :

no du test	1	2	3	4	5	6	7	8	9	10
XAI Sélectionné										
largeur de la zone	150	150	245	245	282	282	200	200	450	450
longueur de la zone	150	150	245	245	298	298	760	760	450	450
total de pixels	22500	22500	60025	60025	84036	84036	152000	152000	202500	202500
nbre d'images	1	1	1	1	2	2	3	3	4	4
temps d'extraction	4,23	4,13	4,2	4,22	5,5	4,44	4,2	4,2	4,8	4,23
temps de preprocessing	0,01	0,01	0,12	0,01	0,14	0,12	0,1	0,04	0,03	0,03
temps d'inférence	12,65	12,48	14,69	13,63	14,17	12,95	12,2	13,23	12,63	12,03
temps de XAI	15,41	13,29	18,73	12,35	18,12	12,1	22,14	11,89	27,62	13,19
temps total	32,3	29,91	37,74	30,21	37,93	29,61	38,64	29,36	45,08	29,48

no du test	11	12	13	14	15	16	17	18	19	20
XAI Sélectionné										
largeur de la zone	270	270	450	450	245	245	500	500	750	750
longueur de la zone	1560	1560	750	750	1750	1750	1000	1000	750	750
total de pixels	421200	421200	337500	337500	428750	428750	500000	500000	562500	562500
nbre d'images	5	5	6	6	7	7	8	8	9	9
temps d'extraction	4,23	4,4	4,2	4,4	4,4	5,7	4,3	5,45	4,3	4,3
temps de preprocessing	0,03	0,04	0,16	0,17	0,17	0,19	0,06	0,07	0,18	0,19
temps d'inférence	14,7	15,11	12,2	13,07	12,3	14,76	13,4	13,13	13,91	14,37
temps de XAI	26,74	13,33	30,3	11,8	33,04	12,8	36,4	13,89	38,8	10,8
temps total	45,7	32,88	46,86	29,44	49,91	33,45	54,16	32,54	57,19	29,66

no du test	21	22	23	24	25	26	27	28	29	30
XAI Sélectionné										
largeur de la zone	500	500	250	250	510	510	215	215	500	500
longueur de la zone	1250	1250	2750	2750	1510	1510	3260	3260	1750	1750
total de pixels	625000	625000	687500	687500	770100	770100	700900	700900	875000	875000
nbre d'images	10	10	11	11	12	12	13	13	14	14
temps d'extraction	4,34	5,67	5,8	4,5	4,5	4,44	4,48	4,53	4,4	5,69
temps de preprocessing	0,22	0,24	0,09	0,08	0,08	0,08	0,22	0,23	0,22	0,4
temps d'inférence	11,87	14,65	15,23	14,13	15,77	14,98	13,19	13,61	13,53	15,28
temps de XAI	40,77	12,96	43,02	14,25	45,35	12,57	48,9	14,07	52,93	10,35
temps total	57,2	33,52	64,14	32,96	65,7	32,07	66,79	32,44	71,08	31,72

no du test	31	32	33	34	35	36	37	38	39	40
XAI Sélectionné										
largeur de la zone	750	750	1000	1000	250	250	750	750	250	250
longueur de la zone	1250	1250	1000	1000	4250	4250	1500	1500	4750	4750
total de pixels	937500	937500	1000000	1000000	1062500	1062500	1125000	1125000	1187500	1187500
nbre d'images	15	15	16	16	17	17	18	18	19	19
temps d'extraction	4,5	5,77	5,9	4,52	5,6	4,61	4,6	4,5	4,6	4,5
temps de preprocessing	0,22	0,1	0,32	0,11	0,3	0,23	0,13	0,13	0,29	0,14
temps d'inférence	14,51	12,45	15,02	14,72	15,87	14,02	15,17	12,86	14,17	12,59
temps de XAI	56,08	14,81	58,42	11,34	59,9	12,45	67,85	13,79	69,34	13,83
temps total	75,31	33,13	79,66	30,69	81,67	31,31	87,75	31,28	88,4	31,06

no du test	41	42	43	44	45	46	47	48	49	50
XAI Sélectionné										
largeur de la zone	1000	1000	2000	2000	3000	3000	4000	4000	5000	5000
longueur de la zone	1250	1250	2000	2000	3000	3000	4000	4000	5000	5000
total de pixels	1250000	1250000	4000000	4000000	9000000	9000000	16000000	16000000	25000000	25000000
nbre d'images	20	20	20	20	20	20	20	20	20	20
temps d'extraction	4,85	4,58	4,5	4,56	4,74	4,68	4,87	4,66	4,8	4,6
temps de preprocessing	0,3	0,29	0,14	0,13	0,26	0,27	0,14	0,14	0,47	0,29
temps d'inférence	14,74	13,93	14,5	13,59	14,52	15,87	14,64	15,73	17,54	12,72
temps de XAI	70,09	12,99	68,63	12,81	69,46	13,44	67,81	15,31	67,73	13,88
temps total	89,98	31,79	87,77	31,09	88,98	34,26	87,46	35,84	90,54	31,49

no du test	51	52	53	54	55	56				
XAI Sélectionné										
largeur de la zone	10000	10000	15000	15000	20000	20000				
longueur de la zone	10000	10000	15000	15000	20000	20000				
total de pixels	100000000	100000000	225000000	225000000	400000000	400000000				
nbre d'images	20	20	20	20	20	20				
temps d'extraction	4,73	4,71	4,6	4,8	5,5	4,54				
temps de préprocessing	0,14	0,14	0,25	0,3	0,19	0,14				
temps d'inférence	11,88	15,81	12,5	14,36	17,4	13,92				
temps de XAI	69,26	15,48	73,75	14,35	69	13				
temps total	86,01	36,14	91,1	33,81	92,09	31,6				

Tests sur le CPU, centre 3, patient 73, nœud 1 :

no du test	1	2	3	4	5	6	7	8	9	10
XAI Sélectionné										
largeur de la zone	150	150	245	245	282	282	200	200	450	450
longueur de la zone	150	150	245	245	298	298	760	760	450	450
total de pixels	22500	22500	60025	60025	84036	84036	152000	152000	202500	202500
nbre d'images	1	1	1	1	2	2	3	3	4	4
temps d'extraction	4,28	4,59	4,32	4,48	5,62	4,5	5,15	4,45	4,35	4,66
temps de préprocessing	0,1	0,01	0,1	0,12	0,02	0,02	0,15	0,11	0,03	0,04
temps d'inférence	11,89	12,69	12,1	12,22	15,1	12,42	12,57	12,2	12,47	14,08
temps de XAI	16,43	11,4	16,82	10,92	21,38	11,94	25,6	12,22	30,75	11,74
temps total	32,7	28,69	33,34	27,74	42,12	28,88	43,47	28,98	47,6	30,52

no du test	11	12	13	14	15	16	17	18	19	20
XAI Sélectionné										
largeur de la zone	270	270	450	450	245	245	500	500	750	750
longueur de la zone	1560	1560	750	750	1750	1750	1000	1000	750	750
total de pixels	421200	421200	337500	337500	428750	428750	500000	500000	562500	562500
nbre d'images	5	5	6	6	7	7	8	8	9	9
temps d'extraction	4,48	4,55	5,03	4,44	4,4	4,48	4,5	4,63	4,43	4,55
temps de préprocessing	0,17	0,04	0,22	0,04	0,15	0,05	0,18	0,06	0,16	0,07
temps d'inférence	15,79	13,04	15,44	12,73	13,23	13,12	12,46	11,33	12,45	13,55
temps de XAI	36,02	12,1	43,63	10,16	49,8	11,51	56,06	12,3	61,63	11,54
temps total	56,46	29,73	64,32	27,37	67,58	29,16	73,2	28,32	78,67	29,71

no du test	21	22	23	24	25	26	27	28	29	30
XAI Sélectionné										
largeur de la zone	500	500	250	250	510	510	215	215	500	500
longueur de la zone	1250	1250	2750	2750	1510	1510	3260	3260	1750	1750
total de pixels	625000	625000	687500	687500	770100	770100	700900	700900	875000	875000
nbre d'images	10	10	11	11	12	12	13	13	14	14
temps d'extraction	4,94	4,61	6,08	4,53	4,78	4,62	4,59	4,61	4,97	4,82
temps de préprocessing	0,19	0,07	0,2	0,08	0,2	0,09	0,19	0,09	0,23	0,11
temps d'inférence	13,05	12,95	11,59	13,42	12,92	13,41	13,91	13,69	12,92	13,73
temps de XAI	65,44	11,61	70,74	10,1	75,22	11,62	79,31	10,35	80,81	12,03
temps total	83,62	29,24	88,61	28,13	93,12	29,74	98	28,74	98,93	30,69

no du test	31	32	33	34	35	36	37	38	39	40
XAI Sélectionné										
largeur de la zone	750	750	1000	1000	250	250	750	750	250	250
longueur de la zone	1250	1250	1000	1000	4250	4250	1500	1500	4750	4750
total de pixels	937500	937500	1000000	1000000	1062500	1062500	1125000	1125000	1187500	1187500
nbre d'images	15	15	16	16	17	17	18	18	19	19
temps d'extraction	5,18	4,67	5,09	4,77	4,88	4,68	4,63	5,39	5,47	4,85
temps de préprocessing	0,24	0,11	0,32	0,12	0,26	0,12	0,22	0,14	0,29	0,13
temps d'inférence	13,1	13,48	14,8	11,59	12,9	15	12,95	12,82	12,97	14,96
temps de XAI	82,65	12,89	89,8	12,75	95,96	14,12	100,7	11,5	103,6	13,5
temps total	101,17	31,15	110,01	29,23	114	33,92	118,5	29,85	122,33	33,44

no du test	41	42	43	44	45	46	47	48	49	50
XAI Sélectionné										
largeur de la zone	1000	1000	2000	2000	3000	3000	4000	4000	5000	5000
longueur de la zone	1250	1250	2000	2000	3000	3000	4000	4000	5000	5000
total de pixels	1250000	1250000	4000000	4000000	9000000	9000000	16000000	16000000	25000000	25000000
nbre d'images	20	20	20	20	20	20	20	20	20	20
temps d'extraction	4,7	5,52	4,74	4,95	4,92	4,98	4,78	4,88	4,66	5,4
temps de préprocessing	0,25	0,15	0,24	0,14	0,27	0,14	0,24	0,15	0,24	0,16
temps d'inférence	12,72	14,57	13,7	14,38	13,04	15,02	15,19	12,05	13,25	14,84
temps de XAI	109,83	11,7	110,99	11,96	111,53	12,33	107,98	13	111,95	10,68
temps total	127,5	31,94	129,67	31,43	129,76	32,47	128,19	30,08	130,1	31,08

no du test	51	52	53	54	55	56				
XAI Sélectionné										
largeur de la zone	10000	10000	15000	15000	20000	20000				
longueur de la zone	10000	10000	15000	15000	20000	20000				
total de pixels	100000000	100000000	225000000	225000000	400000000	400000000				
nbre d'images	20	20	20	20	20	20				
temps d'extraction	5,86	5,09	5,5	4,95	4,8	4,95				
temps de préprocessing	0,32	0,15	0,1	0,16	0,25	0,15				
temps d'inférence	13,97	14,51	12,79	12,9	12,84	14,49				
temps de XAI	107,77	12,16	112,85	12,21	112,17	11,95				
temps total	127,92	31,91	131,24	30,22	130,06	31,54				

Source : images de l'auteur.

ANNEXE II : Product backlog du projet

Product Backlog									
Number	To Do							Done	Priority
1	Integration of the dataset into the web based slide viewer								Must
	- Read informations about OpenSeaDragon								
	- understand dataset sizes, formats								
	- Integrate Images by URL in the prototype								
3	Extends the OpenSeaDragon viewer with plugins for annotations								Must
	- Choose a plugin to annotate images								
	- Motivate my choice								
	- Install the plugin								
	- Do a demo of the plugin								
4	Integration of scripts in python wrote by Mara								Must
	- Add radio buttons for the parameters selected by the user (cnn and xai)								
	- Integrate the option to trigger execution of python scripts on an external server								
5	Display outcomes of the scripts								Must
	- Retrieve the outcome of the scripts								
	- Display it into the viewer								
6	Write report								
	- Document about the state of art and write it								
	- Explain the implementation and the results								
7	Integration of the React plugin								Should
	- Integrate OpenSeaDragon with React								

Source : image de l'auteur.

ANNEXE III : Journal de bord du projet

Date	Content	H
10.mai	Réunion avec Mara	1
	Transports	1
	Documentation sur les cancers	2
	Documentation sur OpenSeaDragon	4,5
11.mai	Documents administratifs (planning, journal de bord, organisation...)	2
	Documentation sur le Deep Learning	5
12.mai	Réunion avec Mara et Roger	1
	Documentation sur le Deep Learning	6
17.mai	Organisation du rapport (style, forme, titres)	1,5
	Réunion avec Mara	0,5
	Création d'une app React avec un titre	1
	Implémentation du viewer OpenSeadragon dans l'app	2
	Recherche d'une solution pour implémenter les images de camelyon17	3
18.mai	Recherche de comment créer une image DZI	6
19.mai	Réunion avec Henning	0,5
24.mai	Essais pour utiliser OpenSlide pour transformer des images en DZI	6
25.mai	Travail sur le product backlog et autres administratifs	1,5
	Réunion avec Mara	0,5
	Réunion avec Roger pour les images DZI	2
	Configuration d'un container Docker pour transformer les images	2
	Implémentation des images avec le prototype	3
27.mai	Réunion avec Henning	0,5
28.mai	Travailler sur le menu de droite pour le choix des images	6
31.mai	Réunion avec Mara	0,5
	Essai d'implémentation du plugin d'annotation	6
01.juin	Essai d'implémentation du plugin d'annotation	7
07.juin	Réunion avec Mara	0,5
	Essai d'implémentation du plugin d'annotation	3,5
	Réunion avec Roger pour le plugin	1
	Finitions d'implémentation du plugin	3,5
08.juin	Implémentation de la partie résultats (données en brut)	3,5
	Amélioration du visuel du site	3,5
09.juin	Début de l'écriture du choix de la technologie	6
14.juin	Lecture des articles et écriture du choix de la technologie	6,5
15.juin	Lecture des articles et écriture du choix de la technologie	6,5
16.juin	Lecture des articles pour l'état de l'art	6
21.juin	Affichage des images depuis le serveur	2
	Résolution du problème d'import des données	5
22.juin	Préparation du code pour recevoir la réponse Https du serveur	3
	Lecture des documents pour l'état de l'art	4
23.juin	Lecture des documents pour l'état de l'art	2,5
	Écriture de la partie état de l'art	5

28.juin	Écriture de l'avant-propos, les remerciements et l'introduction	6
29.juin	Lecture des articles pour l'état de l'art	6
30.juin	Lecture des articles pour l'état de l'art	7
05.juil	Écriture de l'état de l'art	7
06.juil	Écriture de l'état de l'art	7
07.juil	Relecture de l'état de l'art et corrections des références	3
	Préparation de mon code pour l'implémenter dans un container Docker	4
08.juil	Réunion avec Henning	0,5
	Correction des problèmes du déploiement	2
	Récupération des images du NAS plutôt que celles en local	3
	Correction de l'état de l'art avec les remarques de Mara et d'Henning	2
09.juil	Affichage des images proprement depuis le NAS	3
	Correction de l'état de l'art avec les remarques de Mara et d'Henning	4
12.juil	Préparation de l'interface pour correspondre aux réponses du script	4
	Lecture du document des normes APA	3
13.juil	Réunion avec Mara	0,5
	Travail sur l'ajout du script au projet	6,5
14.juil	Affichage correct des réponses du script	5
	Correction de l'état de l'art avec les remarques de Mara et d'Henning	2
15.juil	Gestion des erreurs de réponses du script	3
	Correction de l'état de l'art avec les remarques de Mara et d'Henning	2
	Correction de toutes les citations pour les mettre en norme APA	3
16.juil	Gestion et affichage des erreurs	2
	Gestion de la modification des zones du plugin sans erreurs	1,5
	Début de l'écriture de la partie implémentations	4
19.juil	Réajustement de l'affichage des résultats (changement de taille des images)	1
	Écriture d'une section Docker de l'état de l'art	2,5
	Suite de l'écriture de l'implémentation	3,5
20.juil	Réalisation des tests sur GPU et écriture de la partie résultats	8
21.juil	Réalisation des tests sur CPU et écriture de la partie résultats	8
22.juil	Écriture du résumé, des discussions et de la conclusion, finition de la partie résultat	8
23.juil	Relecture des parties résultats, discussion et conclusion	8
26.juil	Modification du rapport suite aux commentaires de Henning et Mara	2
	Création des figures pour illustrer certains points	3
	Relecture et vérification des références, du sens etc.	3
27.juil	Relecture et vérification des références, du sens etc.	8
28.juil	Relecture et vérification des références, du sens etc.	8
29.juil	Relecture et correction de l'orthographe et de la syntaxe	8
30.juil	Corrections de la syntaxe et de l'orthographe avec le feedback des relecteurs	8
02.août	Création de l'affiche et préparation des clés USB, finition et envoi à l'impression	8
Total		312

Source : données de l'auteur.

DÉCLARATION DE L'AUTEUR

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du responsable de filière et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- Dr Henning Müller
- Mme Mara Graziani
- M. Roger Schaer

Sierre, le 2 août 2021

Nicolas Constantin