Projet Final

Maladie de Lynch

Groupe :
Laurent Dang-Vu
Homer Durand
Najwa Moursli
Solène Saulnier

Tuteur au sein de l'école :
Xavier Tannier
Thibault HILAIRE

Polytech Sorbonne

Spécialité
Mathématiques Appliquées et Informatique Numérique
Année 5
2020 – 2021

L'ensemble de l'équipe remercie
Thibault Hilaire pour son rôle de coordinateur,
Xavier Tannier pour son encadrement, son aide précieuse et pour ses conseils,

Table des matières

Table des figures

Introduction

La maladie de Lynch est associée à un fort risque de développer un cancer colorectal, *cancer colorectal* (*CCR*) [1]. Ce risque peut être réduit à condition de prendre en charge le patient dès la détection des précurseurs et/ou des caractéristiques propres au syndrome de Lynch. Cette maladie peut être détectée au moyen d'un test qui consiste à identifier des lésions des entérocytes grâce à des techniques immunohistochimiques sur des prélèvements de tissus du colon d'un patient. Une crypte MMR (Mismatch Repair) déficiente dites *aberrant crypt foci* chez un patient suffit à dire qu'il est atteint du syndrome de Lynch [3].

Différentes caractéristiques permettent de classer une crypte comme étant saine ou déficiente. Le critère le plus évident est la coloration des cryptes visualisées par microscopie, brune pour une crypte saine (myofibroblastes sains), bleu pour une crypte déficiente (*aberrant crypt foci*) [27]. Il existe d'autres critères qui sont développés au cours de ce rapport. Ces cryptes déficientes sont rares et leur recherche fastidieuse. De nombreuses lames sont à examiner au microscope par des spécialistes ce qui prend environ 15 minutes par lames. Chaque patient ayant une dizaine de lame, soit environ 2h30 pour avoir le résultat d'un seul patient.

Ce projet a donc pour but de mettre à contribution les méthodes de deep learning et machine learning aux services des spécialistes pour la détection des entérocytes et la distinction des différents types de cryptes, saines ou déficientes, afin de diminuer le temps de diagnostic et de permettre une prise en charge plus rapide du patient dans le processus de l'installation des symptômes de la maladie. Il s'inscrit dans la continuité du stage de *Clémence Lanfranchi* [25].

Dans un premier temps, on se penchera sur la définition médicale du Syndrome de Lynch [1] afin d'améliorer la compréhension de cette maladie et de comprendre la difficulté de son diagnostique.

Suivra alors, le pré traitement des données où nous apporterons des détails sur la segmentation des cryptes au travers du rapport de *Clémence Lanfranchi* [25] et la méthode de *ranking* qu'elle utilise pour la distinction entre les deux types de crypte.Nous présenterons également au sein de cette partie le jeu de données, et les modifications effectuées en amont pour préparer les données pour les algorithmes de classification. Ce traitement est d'autant plus important et spécifique du fait que nous ayons en notre possession peu de représentants de la classe déficiente.

Par la suite, nous développerons, la mise en place du protocole expérimental, en détaillant les algorithmes de classification que nous avons utilisés et leurs résultats pour les interpréter.

Nous proposerons une ouverture sur les perspectives d'amélioration du projet tant sur les techniques possible d'introduire dans le domaine du *machine learning* (*ML*) que les critères discriminants physio-histologiques autres que celui abordé dans le rapport de *Clémence Lafranchi* [25] afin d'apporter de nouvelles métriques à nos algorithmes pour finalement, réfléchir aux impacts que peut avoir ce projet sur l'environnement [5], la société[29] et l'éthique de celui-ci [45].

cryptes déficientes

1. Définition du Syndrome de Lynch

Le syndrome de Lynch est une affection héréditaire, par transmission La transmission autosomique dominante prédisposant à plusieurs cancers, lesquels constituent le spectre du syndrome de Lynch. Le côlon majoritairement, l'endomètre et l'ovaire sont les organes les plus touchés avec des risques cumulés de développer un cancer, atteignant les 75%. Cette famille de cancer étant répertoriée comme cancer colorectal ou les *CCR* . Comparé aux formes sporadiques du cancer du côlon, le syndrome de Lynch apparaît à un âge moins avancé (vers 45 ans) et les lésions ont tendance à être plus proximales par rapport à l'angle colique gauche. La lésion première est habituellement un adénome colique unique [1]. Cependant, comme dans la polypose adénomateuse familiale (autre forme hériditaire de cancer colorectal, *CCR*), de nombreuses manifestations extracoloniques se produisent. Parmi les pathologies bénignes, on compte un cancer de la peau de bas grade, le kératoacanthome. D'autres tumeurs malignes associées fréquentes sont les tumeurs de l'endomètre et les tumeurs de l'ovaire (39% respectivement et 9% de risque, avant l'âge de 70 ans). Les patients ont également un risque élevé d'autres cancers, dont des cancers de l'estomac, des voies urinaires, du pancréas, de l'arbre biliaire, vésicule biliaire, de l'intestin grêle et du cerveau.[40]

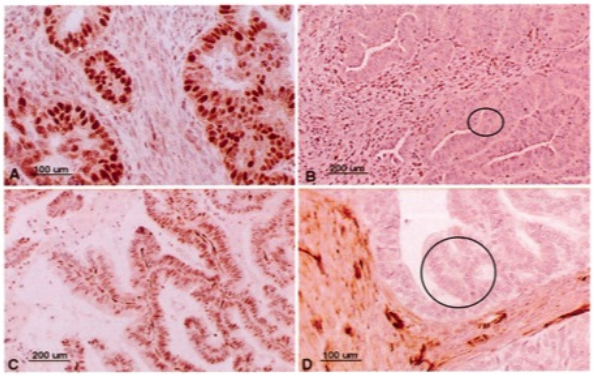


Figure 1.1 – Marquage immunohistochimique **A)** Marquage normal pour la protéine MSH2, les noyaux des cellules sont bruns marquant la présence de protéine MSH2. **B)** Marquage négatif pour la protéine MSH2. Les noyaux des cellules tumorales (entourées) ne sont pas bruns, démontrant l'absence de protéine MSH2. Pareil pour **C) et D)** avec la proteine MLH1.

Ce syndrome est caractérisé par des lésions sur les *aberrant crypt foci* (en français cryptes déficientes) qui sont causées des mutations sur les gènes *Instabilité des microsatellites* (*MMR*) (Mismatch repair), c'est-à-dire les gènes impliqués dans la réparation des erreurs de réplication de l'ADN. Ces gènes endommagés sont transcrits en ARNm puis traduits en protéines qui contrôlent la qualité de l'ADN, en particulier de l'ADN qui est synthétisée au cours des divisions cellulaires, et réparent les erreurs. Ces protéines *MMR* défectueuses, qui sont misses en évidence sur la figure **1.1** sont incapables d'exercer son travail de réparation de l'ADN [27].Le patient a ainsi une prédisposition à développer un cancer, qui survient si la seconde copie du gène subit elle-aussi une altération (altération limitée aux cellules tumorales). Les conséquences des altérations des deux copies d'un gène *MMR* sont visibles au niveau des cellules tumorales qui vont montrer comme caractéristiques une Instabilité des micro satellites ou phénotype MSI (MicroSatellite Instability) [3].

Le diagnostic spécifique du syndrome de Lynch est confirmé par des examens complémentaires génétiques et/ou immunohistochimiques. Cependant, il est difficile de déterminer quel patient relève de ces tests car, contrairement à la polypose adénomateuse familiale, il n'existe pas de tableau phénotypique caractéristique. Ainsi, afin d'évoquer le diagnostic du syndrome de Lynch, il faut pratiquer une anamnèse familiale détaillée chez tous les patients jeunes diagnostiqués avec un cancer colorectal, *CCR*.[40]

Pour correspondre aux critères d'Amsterdam II [35] permettant le diagnostic du syndrome de Lynch, il faut présenter les 3 éléments suivants de l'anamnèse :

— Trois apparentés du 1er degré ou plus qui ont un syndrome de Lynch ou un cancer colorectal, *CCR*
— Un cas de cancer colorectal, *CCR* impliquant au moins deux générations
— Au moins un cas de cancer colorectal, *CCR* avant l'âge de 50 ans

D'autres modèles prédictifs (p. ex., modèle Lynch Syndrome Prediction Model) et d'autres critères (p. ex., les critères de Bethesda [44]) sont utilisés par certains praticiens.

2. Pré-traitement

2.1 Travail en amont

Ce projet fait suite au stage de *Clémence Lanfranchi* [25] réalisé d'avril à août 2020. Toute cette partie est détaillée dans son rapport [25].
Rappelons que le problème se divise en deux parties

- détection des cryptes intestinales
- distinction des cryptes déficientes et des cryptes saines.

Dans ce projet, l'une des principales difficultés réside dans le manque d'annotations. En effet, sur une lame, il y a entre 0 et 30 cryptes déficientes environ pour entre 1 292 et 11 304 cryptes. Ce qui ne représente même pas 1 % des cryptes intestinales sur une lame. Dans la plupart des cas, il y a 1 à 2 crypte intestinale(s) déficiente(s) par lame. Sur l'ensemble du jeu de données, il y a donc :

- 30 cryptes déficientes (pour lesquelles nous avons les annotations sur les lames, et les annotations sur les images découpées)
- 202 cryptes déficientes (pour lesquelles nous avons les annotations sur les lames seulement)
- 20 307 cryptes saines (pour lesquelles nous avons les annotations sur les lames, et les annotations sur les images découpées)

Cette classe (de crypte déficiente) est non seulement, sous représentée par rapport à la classe des cryptes intestinales saines, mais elle est également, très peu présente dans le jeu de données. Ce manque d'exemples lors de l'entraînement du modèle, implique que ce dernier aura des difficultés à reconnaître une crypte déficiente. Or, il est primordial de détecter une crypte déficiente. La lame sera vérifiée par le spécialiste. Dans le cas où le patient a bel et bien le syndrome, cela signifie qu'il doit être traité au plus vite. C'est pour cette raison que *Clémence Lanfranchi* privilégie une méthode de **ranking** plutôt qu'une méthode de **classification** [25]. Ainsi, un classement permet d'avoir les cryptes intestinales les plus probables d'être déficientes, et donc les lames susceptibles de contenir une crypte déficiente au moins. Les spécialistes n'ont alors plus qu'à vérifier les lames en haut du classement. Cela réduit drastiquement le nombre de lames à examiner. Ce processus permet aux spécialistes de gagner du temps et de pouvoir prendre en charge leurs patients dans de meilleurs délais.

Nous pouvons récapituler brièvement la méthode qu'elle utilise par les étapes suivantes :

- utilisation de **Yolov3** [21] pour prédire les sorties
- utilisation de **k-means** permettant de savoir la couleur dominante de chacune des images des cryptes intestinales
- construction du graphe des voisins (des cryptes intestinales sur une lame)
- évaluation du score

La première étape utilisant **Yolov3** [21] permet de détecter les cryptes intestinales. **Yolov3** [21] est un réseau de neurones artificiels spécialisé dans la détection et l'analyse d'objets dans l'image. L'utilisation de **Yolov3** [21] permet d'avoir en amont un réseau de neurones artificiels pré-entrainé. Par la suite le ce réseau de neurones artificiels est ré-entrainé plus spécifiquement sur notre jeu de données. Notre jeu de données est divisé en jeu d'apprentissage (70.0 %), jeu de validation (10 %), jeu de test (20 %).

Lors de la troisième étape, les voisins correspondent à l'ensemble des cryptes intestinales présentes dans un rayon de trois fois le rayon de la crypte intestinale. Cette valeur arbitraire offre de meilleurs résultats que les autres valeurs de rayon. Pour calculer le score, la formule d'une sigmoïde est utilisée puisqu'elle donne de meilleurs résultats de prédiction. C'est à dire que le score à la forme suivante :

$$score = \frac{d}{1 + exp(-\alpha * (\beta - b)}$$

où d correspond à la différence de couleur entre la crypte intestinale et ses voisins, b correspond à la distance entre la crypte intestinale et la couleur dominante de l'image, et α et β sont des coefficients permettant d'améliorer la distinction du bleu. Une fois le score calculé, le classement peut être réalisé. Au cours de ce projet, nous étudierons les résultats obtenus avec des méthodes et des algorithmes d'apprentissage supervisé pour de la classification.

2.2 Présentation du jeu de données

Le jeu de donnée se compose en image des différentes lames (*slides*) avec leurs annotations des cryptes déficientes seulement. Chacune de ses lames est divisée en "images découpées" (*cropped-image*) avec des annotations sur les cryptes intestinales. Comme indiqué dans le rapport *Clémence Lanfranchi* [25] ces lames ont été préparées par des spécialistes de l'Assistance Publique - Hôpitaux de Paris (APHP).

Par la suite, elles sont scannées à l'aide d'un scanner numérique de référence *PathScan RCombi, Excilone*, pour obtenir des images numériques. Ces images sont alors dans un format *JPEG2000* (extension .j2). *TANNIER Xavier* les transforme à l'aide de l'outil **Kakadu** [43] au format *.tiff Tag(ged) Image File Format* (*TIFF*). Davantage d'informations sur cet outil sont disponibles sur le lien [43]. Ces images sont alors réduites de moitié au moyen d'outils classiques de conversion (**VIPS**) [23]. Une fois la réduction faite, il suffit de convertir au format *.jpg Joint Photographic Experts Group* (*JPEG*) pour obtenir les images présentes dans le jeu de données.

La taille des images des lames est particulièrement grande ce qui pose des problèmes de stockage et de temps d'exécution des algorithmes pour l'apprentissage et la prédiction. C'est pour cette raison que la qualité de l'image est réduite d'un facteur 2 lors de la conversion. Puis, ces images des lames au format *.jpg JPEG* sont alors découpées en *cropped-image* et la qualité des images est à nouveau réduite d'un certain facteur. Cette diminution de qualité provoque un léger changement dans les coordonnées entre les lames et les images découpées.

$$x_{lames} = 2.5 \times x_{croppedImage} \qquad y_{lames} = 2.5 \times y_{croppedImage}$$

Les annotations sont réalisées manuellement sur **supervisely** [9] puis enregistrées sous format *.json* (*JavaScript object notation* (*JSON*)). Il est important de préciser également que tous les patients ont signé un contrat de non opposition lors de leur opération. C'est la raison pour laquelle les images peuvent être utilisées pour ce projet. En revanche ces images doivent rester confidentielles et ne doivent pas apparaître sur des réseaux publics. Néanmoins, l'anonymat est conservé puisqu'aucun rapprochement entre la lame et le patient peut être effectué.

2.3 Réécriture des annotations

Ce projet a pour but d'appliquer une méthode d'apprentissage afin d'arriver à une classification des cryptes intestinales. Il existe des moyens pour pallier le manque de données avec par exemple de la *data augmentation* ou augmentation de données. Il existe également des modèles et des algorithmes implémentés pour des jeux de données avec peu d'annotation. Ces méthodes seront détaillées par la suite. Cependant, les données n'ont pas été écrites pour faire de la classification. Rappelons que les annotations des lames contiennent les cryptes déficientes, tandis que les images découpées contiennent les annotations des cryptes intestinales sans distinction entre déficientes ou saines. Nous avons donc écrit un programme permettant de reporter les annotations des cryptes déficientes des grandes lames sur les annotations des images découpées. Le principe est le suivant :

Algorithm 1 Ré-écriture des données

Require: annotations sur lames, annotations sur images découpées
Ensure: annotations sur les cryptes intestinales saines et déficientes (distinction)

```
for tous les fichiers annotations sur lames do
    for toutes les cryptes déficientes do
        récupération des coordonnées dans la lame
        Conversion dans l'image découpée (/2.5)
        for tous les fichiers annotations sur image découpée do
            for toutes les cryptes intestinales do
                récupération des coordonnées dans l'image découpée
                if coordonnées dans la lame et l'image découpée correspondent then
                    ajout de la mention "déficiente" dans les annotations de l'image découpée
                end if
            end for
        end for
    end for
end for
```

2.4 Data augmentation

L'apprentissage supervisé requiert un grand nombre de données avec des annotations associées pour avoir des meilleurs résultats. Le principal risque de ne pas avoir assez de données est l'*overfitting*, c'est à dire un modèle colle trop aux données et ne généralise pas assez. Le jeu de données contient une quantité finie d'annotation et dans la plupart des cas ce nombre est insuffisant.

La *data augmentation* a été développé pour combler ce manque de données et ainsi améliorer l'apprentissage et donc la fiabilité des résultats de prédiction. Elle est souvent utilisée pour des jeux de données sur des images en *apprentissage profond* ou en *machine learning*. Les grandes lignes de la *data augmentation* et de son fonctionnement sont disponibles dans la publication [8].

Il existe principalement 2 manières de générer des données.
— recopier les images sans modification
— réaliser des transformations sur les images

Les transformations sur les images peuvent être des rotations, des déformations, des recadrages, des changements de couleurs, l'ajout de bruits, l'ajout d'un flou, etc. Il suffit d'avoir le filtre pour pouvoir l'appliquer. Plusieurs transformations sont montrées en image dans la publication [4]. Il s'agit d'un tutoriel pour utiliser la *data augmentation* avec **Keras** [7]. Nous utiliserons **Opencv** [18] pour manipuler les filtres puisque nous utilisons **Yolov3** [21] par la suite mais les filtres et le principe restent les mêmes.

L'avantage des transformations, en plus d'augmenter les données, est de pouvoir donner au modèle d'autres situations possibles et probables. Dans notre cas, il y a des variations de luminosité. La source de lumière est fixe sur le microscope, et le microscope est différent d'une lame à l'autre. Ainsi, il peut y avoir des variations. De même, le produit utilisé pour colorer les cryptes déficientes peut avoir des teintes de bleu légèrement différentes. Cette variation est due à une concentration ou une propagation différente du produit sur les cryptes.

Par conséquent, une variation de la luminosité est intéressante puisque le modèle rencontrera différentes luminosités également. Ce principe peut être appliqué pour la netteté de l'image. Il peut arriver que l'image soit de moins bonne qualité que celle de l'apprentissage. Ainsi, appliquer des transformations permet d'ajouter des situations que le modèle rencontrera mais qui ne sont pas dans le jeu de données d'apprentissage, ce qui améliore l'entraînement.

Il est également important de préciser les limites et les dangers de cette méthode. En effet, la transformation effectuée sur l'image ne doit pas avoir d'impact sur les annotations de l'image d'origine. Par exemple, dans notre cas, pour détecter une crypte intestinale, nous avons besoin de ses coordonnées. En exerçant une distorsion de l'image contenant plusieurs cryptes intestinales, il y aura un impact direct sur la position des cryptes intestinales.
Donc, les annotations associées manuellement à la crypte intestinalene seront plus valides. Elles ne doivent pas apparaître lors de l'apprentissage puisqu'elles sont fausses. De même, tant que la couleur reste un des critères de la distinction entre les cryptes déficientes et cryptes intestinales saines, le filtre de changement de couleur ne peut être appliqué. Les rotations n'ont pas d'importance ni la translation.

Ainsi, les seuls filtres qui seront appliqués ici sont (on pourait imaginer les appliquer à une seule crypte déficiente) : le flou(**GaussianBlur(...)** sur la figure **2.2**), la luminosité (**subtract(...)** sur la figure **2.3**), ou **add(...)** sur la figure **2.4**). Toutes ces images ont été transformées à partir de l'image originale en figure **2.1**.

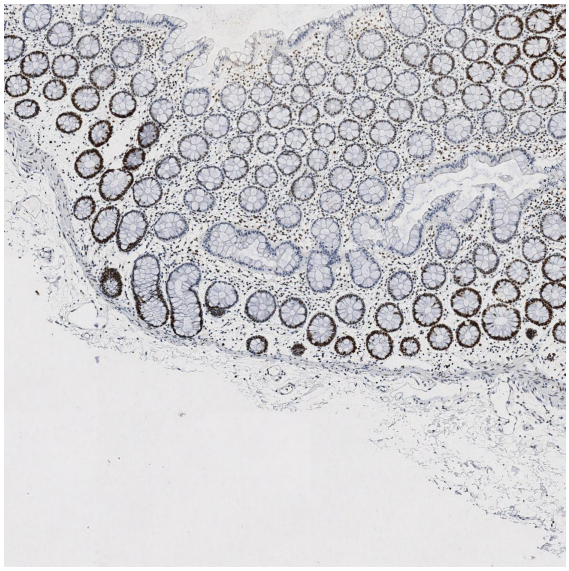


FIGURE 2.1 – Image originale

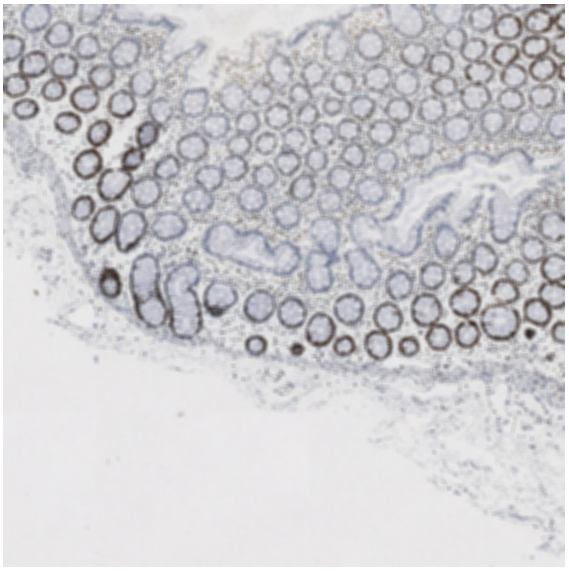


FIGURE 2.2 – Application flou gaussien

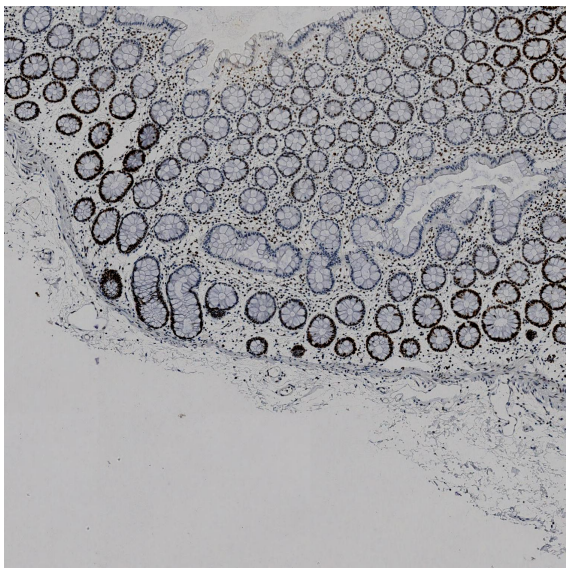


FIGURE 2.3 – Diminution de luminosité

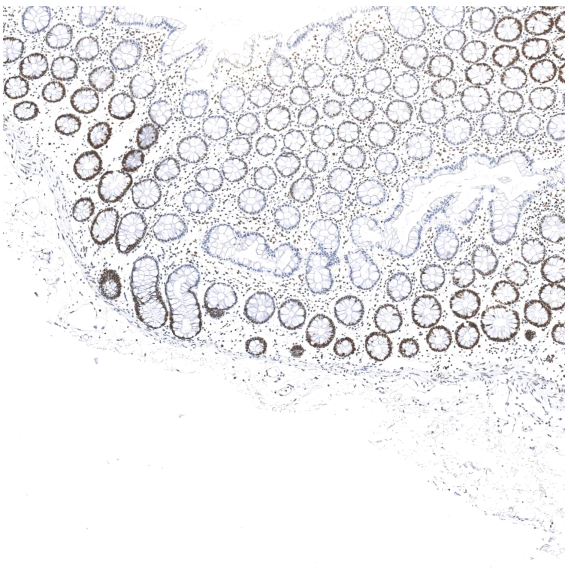


FIGURE 2.4 – Augmentation de luminosité

Un autre inconvénient de la *data augmentation* est la place que les données générées occupent en mémoire. Ainsi, il est préférable d'utiliser les générateurs. C'est un moyen de produire des images "à la volée" et de ne pas avoir à stocker l'ensemble des nouvelles données.

3. Méthode de classification

L'utilisation du modèle **YoloV3** [21] nous a désormais permis de localiser l'ensemble des cryptes intestinales sur les images des coupes. Nous disposons donc d'un jeu de données contenant des images de cryptes intestinales. L'objectif est alors de détecter les crypte déficiente parmi celles-ci. Nous disposons en pour cela d'un très grand nombre de cryptes intestinales saines labelisées (environ 20000) mais nous ne disposons que d'un petit nombre de crypte déficiente labelisées (232 dont 200 de plus faible résolution). C'est donc principalement ici que réside la difficulté de notre tâche. L'utilisation d'une trop grande quantité de crypte déficiente pour l'entraînement du modèle nous empêcherait de pouvoir le tester sur une crypte déficiente significatif. Nous proposons deux approches pour répondre à cette problématique. La première est de n'utiliser que les cryptes intestinales saines pour entraîner un modèle de classification à classe unique (One Class Classification *(*OCC*)*) qui définira une frontière autour de celles-ci de telle sorte à minimiser la probabilité qu'une crypte déficiente apparaisse à l'intérieur de la zone défini par cette frontière.

L'autre approche consiste à utiliser un petit nombre de cryptes déficientes et d'utiliser un modèle de (*pairwise ranking ou comparaison par paires*) pour déterminer si une une crypte du jeu de test est plus proche (en terme de distance euclidienne dans l'espace de ses features (caractéristiques)) de cryptes déficientes ou de cryptes intestinales non-déficientes. On obtiendra ainsi un score de déficience par vote d'ensemble.

Nous commençons cette section par expliquer l'état de nos recherches concernant les modèles de classification d'images et d'extraction de features puis sur les modèles de *One Class Classification* et de *pairwise ranking (comparaison par paires)* qui nous semblent être les plus pertinent pour répondre à notre problème. Nous détaillerons ensuite l'approche que nous avons choisit et les raisons de ce choix. Enfin nous justifierons nos choix concernant les mesures de performance que nous utiliserons dans la phase de test du modèle.

3.1 Modèle d'extraction d'images

3.1.1 Modèles de classification d'image et d'extraction de features

Les performances des réseau de neurones artificiels, et principalement celle des réseau neuronal convolutif, pour la classification d'images et la détection d'objets à fait évoluer le paradigme de la recherche dans ce domaine. Celle-ci s'appuie aujourd'hui principalement sur la recherche des architectures de réseau neuronal convolutif les plus performantes [38]. Pendant très longtemps la classification d'image se différenciait en deux étapes.

La première, l'extraction des caractéristiques des images qui jouait un rôle primordial consistait à plonger les images dans un espace de plus petite dimension et à ainsi les représenter par des vecteurs. On utilisait pour cela un ensemble de filtre qu'il fallait choisir avec attention afin d'extraire des caractéristiques discriminantes pour la tâche que l'on désirait accomplir.

La seconde consistait à utiliser un modèle de classification (*Régression Logistique ou Bayésienne*, perceptron, perceptron multi-couches, support vector machine, etc...) [28] afin de de classer les vecteurs. L'utilisation de couches de convolution (et de pooling, sous échantillonnage de l'image) dans les réseau neuronal convolutif permet désormais d'effectuer ces deux étapes en utilisant un unique modèle. Les couches de convolution permettant de réduire la dimension des images en un vecteur dont les valeurs sont optimisées pour l'étape de classification.

Par ailleurs il a été remarqué que les réseau neuronal convolutif entraînés sur de très large jeux de données (tel qu' **ImageNet**) [17] montrent de très bonnes propriétés de généralisation y compris pour des images sur lesquelles ils ne se sont pas entraînés. Principalement, les premières couches de convolution permettent d'extraire des caractéristiques qui s'avèrent souvent pertinentes pour la classification de nouveaux jeux de données. Cela ouvre donc le champ de ce qu'on appelle l'apprentissage par transfert qui consiste à utiliser un modèle dont les poids ont été pré-entraîné sur un très large jeu de données (dit jeu de donnée *sources*) et à les ré-entraîner sur un nouveau jeu de données (dit *cible*) en utilisant ainsi la capacité de généralisation des poids pré-entraînés. Ces méthodes montrent deux points d'intérêt : elles permettent de faire converger les modèles bien plus rapidement et les modèles ainsi entraînés sont moins sujet au sur-apprentissage et ont donc en général de meilleures performances.

Comme nous l'avons expliqué, les modèles de classification prennent en entrée des vecteurs, la classification d'image nécessite donc une étape d'extraction de features. Or les réseau neuronal convolutif sont particulièrement performant pour cette tâche comme cela a été décrit dans [20]. Cela se fait en entraînant un réseau neuronal convolutif pour une tâche de classification. En retirant la couche de prédiction, le modèle renverra lors de la prédiction

une couche dense de features qui sera alors discriminante pour la tâche de classification.

3.2 One Class Classification

3.2.1 Motivations

Le problème de classification à une classe (One Class Classification) diffère des modèles de classification supervisée (cf apprentissage supervisé) classique car pour ce type de modèles on dispose d'une classe statistiquement bien défini par un jeu d'entraînement (dite classe *cible* ou *positive*) et d'une classe très minoritaire voir inexistante ou ne formant pas un échantillon statistique représentatif (dite classe *aberrante* ou *négative*). Cela rend le problème d'*OCC* plus difficile que les modèles de classification supervisée classiques. Les modèles *OCC* permettent en revanche de résoudre plusieurs problèmes où les méthodes de classification usuelles échouent. Ils montrent par exemple des bonne propriétés pour la détection d'anomalies, la classification de jeux de données fortement déséquilibrés ou comportant une classe dont l'échantillon d'entraînement est statistiquement mal défini.

L'objectif d'un modèle d'*OCC* est donc de trouver une frontière permettant de maximiser le taux d'observation de classe cible à l'intérieur de la zone définie par cette frontière tout en minimisant la probabilité qu'une observation aberrante soit à l'intérieur de cette zone.

3.2.2 Modèle état de l'art

Jusqu'à récemment et avant l'explosion des performances des réseau de neurones artificiels [22] dans un très grand nombre de tâches de classification et de Régression, les modèles représentatifs du One Class Classification étaient les *Isolation Forest*[26] (utilisé pour la détection d'anomalie) et les *One-Class Support Vector Machine*[37]. Certains modèles de classification basé sur des réseau de neurones artificiels montrent aujourd'hui, de meilleures performances que ces premiers[34]. Récemment, l'article [30] a fournit une étude détaillée d'un grand nombre de méthode d'*OCC* basé sur des réseau de neurones artificiels (Deep Auto-Encoder, One Class Convolutional Neural Networks, One class Generative Adversarial Networks, etc..) [28] mais également de modèle basé sur des méthodes statistiques (Support Vector Data Descriptor, One Class Mini-max Probability Machine, etc...) [39].

Une approche intéressante et nous semblant assez inédite est proposé dans [15] pour les problèmes d'imagerie médicale avec jeux de données fortement déséquilibrés. L'idée étant d'appliquer un certain nombre de filtre sur les images du jeu d'apprentissage (composé uniquement de la classe positive), ces filtres représenterons les labels des images du jeu d'apprentissage . On entraîne ensuite un réseau neuronal convolutif à classer les images en fonction du filtre qui leur a été appliqué. On espère ainsi que le réseau neuronal convolutif entraîné sera capable, dans le jeu de test, de bien détecter les filtres appliqués aux images de la classe positive mais sera moins performant pour détecter ceux appliqués à la classe négative. L'article proposant de définir un score inspiré de l'entropie croisée qui permettrait d'attribuer une probabilité d'appartenance à la classe positive à chacune des images du jeu de test.

3.3 Learning To Rank

Dans le domaine de la Recherhce d'Information (*Information Retrieval* ou *IR*) le *Learning To Rank* [6] à pour objectif de classer les documents d'un corpus par ordre de pertinence, étant donnée la requête d'un utilisateur. Ces méthodes sont particulièrement étudiées pour analyser la pertinence des moteurs de recherche mais apparaissent plus généralement dès lors qu'un problème d'ordonnancement [1] se pose.

On différencie en général trois type de modèle de Learning To Rank [6]. L'approche par point (*pointwise learning* s'apparente à un problème de régression. Chaque feature possède dans ce cas un score numérique ou ordinal (dans lejeu d'apprentissage) que l'on cherche à prédire avec un modèle de Régression. Ensuite, l'approche par paires (*pairwise ranking*, celle qui nous intéresse) s'apparente à un problème de classification ou l'on cherche à savoir si deux features du jeu d'entraînement appartiennent à la même classe. Enfin il existe une troisième méthode, dîte par liste (*listwise ranking*) on le modèle cherche directement à optimiser une mesure de performance [2] durant la phase d'entraînement.

3.3.1 Pairwise Ranking

Nous allons dans cette section décrire la méthode de *pairwise ranking* dans un cadre général en se référençant au formalisme présenté dans l'article [13].

1. A différencier des problèmes de classification pour lesquels on cherche à attribuer une classe aux features. Le classement (*ranking*) cherche lui à ordonner les features de manière pertinente
2. Par exemple avec la loss *top-k probability* pour le modèle ListNet

On considère donc le problème d'apprentissage suivant : étant donnés un ensemble de labels $L = \{\lambda_i | \forall i \in \{1, ..., c\}\}$ un ensemble d'exemples (ou de features) $E = \{e_k | \forall k \in \{1, ..., n\}\}$ et un ensemble de préférences $P_k \subset L \times L$ où $(\lambda_i, \lambda_j) \in P_k$ indique que le label λ_i est préféré au label λ_j pour l'exemple e_k, nous cherchons une fontion $f : E \longrightarrow L^n$ qui ordonne les labels $\lambda_i, \forall i = 1, ..., n$ pour tout exemple e_k.

Diverses méthodes peuvent être utilisées pour inférer cette fonction $f(.)$ depuis le jeu d'apprentissage. Les auteurs de [13] proposent d'entraîner un modèle M_{ij} différent pour chacune des $\frac{c(c-1)}{2}$ paires possibles $(\lambda_i, \lambda_j), 1 \leq i < j \leq c$ qui décide pour tout exemple $e_k \in E$ si λ_i est préféré à λ_j ou inversement. On procède, pour la phase de test, à un vote de chacun des modèles M_{ij} pour déterminer un ordre sur les $\lambda_i, i = 1, ..., c$ pour chaque exemple $e_k \in E$. Cette méthode est souvent référencé comme la méthode *one-against-one classification.*

3.3.1.1 Few Shot Learning

Des méthodes plus récentes basé sur des réseaux de neurones artificiels semble aujourd'hui montrer de très bonne performance dans des situations ou le nombre d'exemple de chaque labels est très réduit. Les réseau de neurones artificiels siamois [24] en sont un très bon exemple et notre modèle s'inspire très fortement de leur architecture. Ils proposent une solution intéressante pour l'application de modèles de type *pairwise ranking* à des images en permettant d'extraire les features des images afin que ceux-ci soient optimaux pour résoudre un problème de *ranking*.

Le modèle d'extraction de features utilisé dans [24] est donc un réseau neuronal convolutif siamois prenant en entré deux images qui passe chacune par un réseau de neurones artificiels dont les poids sont similaires (voire figure 3.1) extrayant ainsi un vecteur de features pour chaque image qui passe ensuite par un perceptron multi-couches qui cherche à déterminer si les deux images prises en entré appartiennent à la même classe. Les labels que le modèle cherche à prédire sont donc 1 pour les images appartenant à la même classe et 0 pour les images de classe différente.

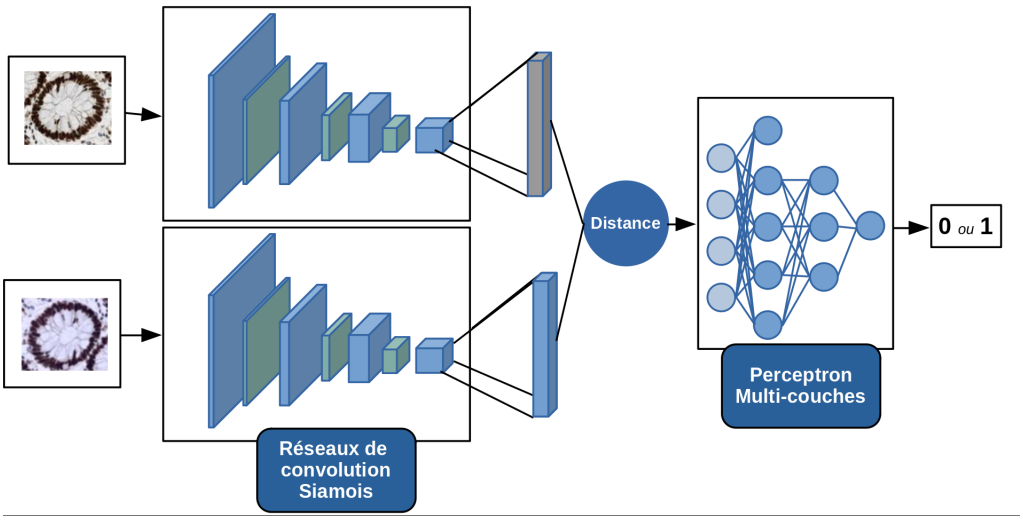


FIGURE 3.1 – Architecture générale d'un réseau de convolution siamois

Les auteurs de [24] propose ensuite une méthode de prédiction des labels des images par une méthode dite de *one-shot learning*. Formellement, on cherche à prédire la classe $\lambda_i, i \in \{1, ..., c\}$ d'une image x et on dispose pour cela d'un ensemble d'images de référence $X_ref = \{x_i | \forall i = 1, ..., c\}$ (une pour chaque classe). Pour ce faire il suffit de prédire, avec le modèle siamois entraîné précédemment, les probabilité $p_i, i = 1, ..., c$ correspondant aux probabilités que l'image x soit de la même catégorie que l'image de référence $x_i, i = 1, ..., c$. On peut dès lors attribuer la classe $max_{p_i}\{\lambda_i | i = 1, ..., c\}$ à l'image x ou classer les labels du plus probable au moins probable $((\lambda_1, ..., \lambda_c) : p_1 > p_2 > ... > p_c)$. On peut généraliser cette méthode en considérant k représentants de chaque classe.

3.4 Notre approche

L'objectif que nous nous fixons pour ce modèle de classification est d'attribuer à chaque crypte intestinale un score de déficience compris entre 0 et 1 (1 correspondant au plus haut risque de déficience et 0 au risque minimal). Il

faudra alors choisir un seuil pertinent pour séparer les cryptes intestinales que nous classerons alors comme potentiellement déficientes et les cryptes intestinales que nous classerons comme saine. Étant données le peu de cryptes déficientes dont nous disposons pour entraîner et tester nos modèles, nous avons choisit d'utiliser un modèle de *pairwise ranking* en utilisant pour la prédiction une méthode de *Few Shot Learning*. Ce type de modèles nous semblent pertinent pour plusieurs raison. Tout d'abord, ils permettent d'obtenir un score de risque de déficience en comparant l'image de la crypte intestinale testée à un ensemble de cryptes déficientes et saines. Ensuite, contrairement aux modèles de *One Class Classification*, ces modèles tirent de l'information de la classe *aberrante* (ici la classe des crypte déficiente) et même s'ils sur-apprennent cette classe – dû au petit nombre d'observations la représentant – cette information semble pertinente pour obtenir des modèles ayant des performances suffisantes pour les rendre utilisables en pratique.

Le choix de différencier la phase d'extraction de features et de *pairwise ranking* s'est pour faciliter les expérimentations mais nous estimons qu'une fois que nous aurons déterminer l'ensemble des hyperparamètres et le modèle de régression utilisé pour le *ranking* il serait préférable d'envisager un modèle unique effectuant ces deux tâches (cela faciliterai son utilisation et pourrait améliorer les performances).

3.4.1 Extraction des features

La première étape du modèle consiste à extraire des *features* discriminants pour différencier les cryptes déficientes et les cryptes intestinales saines. Pour cela nous entraînons un réseaux neuronaux convolutifs sur un ensemble d'images de cryptes intestinales $\mathcal{I}_{train}$. Ce jeu d'entraînement étant déséquilibré, nous décidons de sur-échantillonner les cryptes déficientes afin qu'elles apparaissent en même proportion que les cryptes intestinalessaines. Nous appliquerons plusieurs filtres de *data augmentation* – rotation et contraste – afin que notre modèle généralise mieux à de nouvelles images.

Nous utiliserons pour cette tâche deux modèles de base, le modèle *MobileNetV2* [36] et le modèle *InceptionresNetV2 IRNV2*) [41] tout deux pré-entraînés sur le large jeu de données *ImageNet* [17] . Nous avons choisit ces deux modèles car ils présentent des structures très différentes. Le modèle *InceptionResNetV2* [41] est un modèle relativement lourd (55,873,736 poids) est très profond (780 couches) mais est précis (0.803 de top-1 accuracy). Le modèle *MobileNetV2* [36] est un modèle plus léger (3,538,984 poids pour une profondeur de 88) mais moins performant (0.713 de top-1 accuracy sur *ImageNet* [17]). La couche dense n'aura pas de fonction d'activation afin que les features extraites soient ainsi linéairement séparable. La figure 3.2 présente l'architecture de notre modèle d'extraction de features.

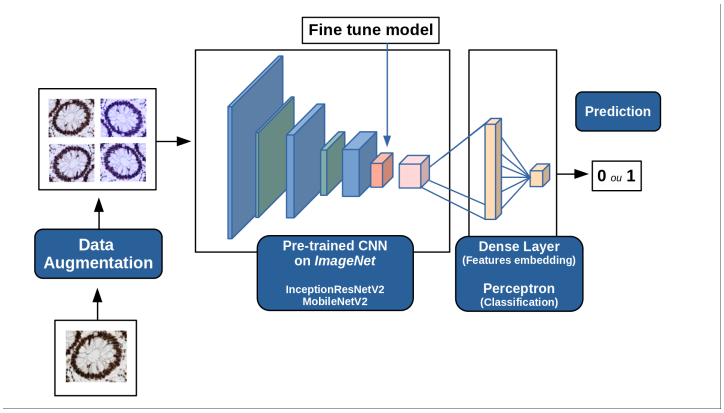


FIGURE 3.2 – Architecture du modèle d'extraction de features

Nous supprimons la couche supérieure du modèle pré-entraîné et la remplaçons par une couche dense de 1280 neurones (qui correspondra à aux features des images) puis un neurone avec une Fonction d'Activation logistique pour la classification. Nous ré-entraînons les dernières couches du modèle et les deux couches denses que nous avons ajouté sur notre jeu d'apprentissage. On peut désormais retirer la dernière couche pour obtenir un extracteur de *features* qui seront alors discriminants pour différencier les images de cryptes déficientes et saines. On peut maintenant extraire les features des jeux d'entraînement et de test en utilisant le modèle pour prédire les features des images en entrées.

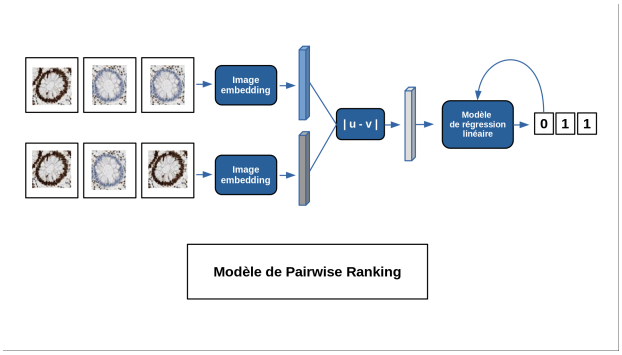


FIGURE 3.3 – Architecture du modèle de pairwise ranking utilisé

3.4.2 Pairwise ranking

Nous avons donc désormais un ensemble de vecteurs représentant notre ensemble d'images. Nous cherchons ici à entraîner un modèle de classification qui nous permettra de différencier les vecteurs extraits d'images de cryptes déficientes V_{def} et de cryptes intestinales non déficientes V_{norm}. Nous utilisons pour cela un modèle du type *pairwise ranking*.

Nous constituons, à partir des vecteurs d'entraînement, deux jeux de données V_{dif} et V_{sim}. Le jeux de données V_{dif} est l'ensemble des différences absolues de chacun des éléments de V_{def} et V_{norm}. Le jeu de données V_{sim} est lui formé par les différences absolues de chacun des éléments de V_{def} différents et V_{norm} différents.

$$
\begin{aligned}
V_{simDef} &= \cup_{\{(v_1,v_2)\in V^2_{def}:v_1\neq v_2\}}|v_1 - v_2| \\
V_{simNorm} &= \cup_{\{(v_1,v_2)\in V^2_{norm}:v_1\neq v_2\}}|v_1 - v_2| \\
V_{sim} &= V_{simNorm} \cup V_{simDef} \\
V_{dif} &= \cup_{(v_1,v_2)\in V_{def}\times V_{norm}}|v_1 - v_2|
\end{aligned}
$$

On associera au jeu de données V_{dif} une classe de valeur -1 et au jeu de données V_{sim} une classe de valeur 1. L'union de ces deux jeux créé donc un jeu d'entraînement V_{train} avec les classes associées y_{train}.

Nous pouvons entraîner un modèle de classification usuel sur le jeu de données créer en utilisant un modèle de *Régression Linéaire*. Par la suite nous noterons $f : \mathcal{R}^p x R^p \longrightarrow \mathcal{D} \subset \mathcal{R}$ où $\mathcal{D}$ = [-1,1], le modèle de prédiction qui associe un score de similarité pour toute paire de vecteurs de dimension p. Ce score de similarité sera proche de 1 si les vecteurs seront considérés similaires l'un de l'autre ou proche de -1 s'ils sont considérés différents. L'architecture de ce modèle est disponible sur la figure 3.3.

3.4.3 Few shot learning

Nous avons maintenant un modèle qui nous permet de comparer deux vecteurs représentant deux images de cryptes intestinales. Plus particulièrement ce modèle retournera 1 si les images sont de la même classe et -1 sinon. Nous allons à partir de cela créer un algorithme associant à chaque crypte intestinale un score compris entre 1 et 0 où 1 correspond au risque maximal de déficience et 0 au risque minimal de déficience.

Nous choisissons aléatoirement $d << |V_{def}|$ représentants de l'ensemble V_{def} et $n << |V_{norm}|$ représentant de l'ensemble V_{norm} que nous noterons respectivement V_{repDef} et $V_{repNorm}$. Ces vecteurs représentent les cryptes de référence auxquelles nous allons comparer les images de test. Plus concrètement, nous voulons affecter un score a un vecteur de features v_{test}. Pour cela nous nous calculons sa différence absolue avec chacun des vecteurs de V_{repDef} et $V_{repNorm}$ que vous stockons respectivement dans $\mathcal{U}_{def}$ et $\mathcal{U}_{norm}$.

$$
\begin{aligned}
\mathcal{U}_{def} &= \{|v_d - v_{test}| : v_d \in V_{repDef}\} \\
\mathcal{U}_{norm} &= \{|v_n - v_{test}| : v_n \in V_{repNorm}\}
\end{aligned}
$$

Puis nous calculons le score associé à chacun des vecteurs de ces ensembles par notre modèle entraîné à l'étape précédente.

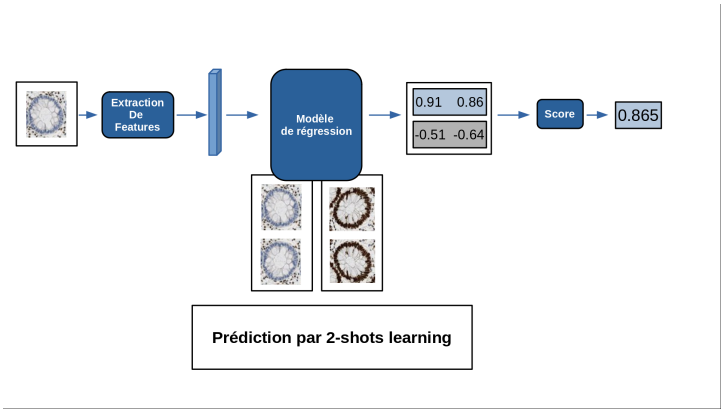


FIGURE 3.4 – Architecture de la méthode de few-shot learning utilisée

$$\mathcal{S}_{def} = \{f(v) : v \in \mathcal{U}_{def}\}$$
$$\mathcal{S}_{norm} = \{f(v) : v \in \mathcal{U}_{norm}\}$$

Nous pouvons désormais calculer le score de déficience du vecteur v_test comme la somme des scores $\mathcal{S}_{def}$ à laquelle on retranche la somme des scores $\mathcal{S}_{norm}$.

$$\mathrm{S}(v) = \frac{1}{2} + \frac{1}{2}\frac{\sum_{s_d \in \mathcal{S}_{def}} s_d - \sum_{s_n \in \mathcal{S}_{norm}} s_n}{|\mathrm{S}_{norm}| + |\mathrm{S}_{def}|}$$

On obtient ainsi un score compris entre 0 et 1. Un score de 1 correspond au fait le vecteur v_test a été considéré similaire à toutes les cryptes déficientes et différent de toutes les cryptes intestinales non déficientes. A l'inverse, un score de 0 correspond au fait le vecteur v_test a été considéré similaire à toutes les cryptes intestinales non déficientes et différent de toutes les cryptes déficientes.

L'architecture de cette méthode est disponible sur la figure 3.4.

3.5 Mesures de performances

Nous allons dans cette section expliquer les différentes mesures de performances que nous avons choisit afin d'évaluer les résultats de nos modèles. Nous disposons pour cela d'un ensemble de 232 image de cryptes déficientes et de 20307 images de cryptes non-déficientes formant le jeu de données $\mathcal{I}$. Afin que les résultats obtenu soient robustes nous utiliserons la *Validation Croisée à k-blocs* – cela sera précisé si ce n'est pas le cas. Afin de pouvoir évaluer la significativité de nos tests nous indiquerons toujours le nombre de cryptes déficientes et non déficientes formant les jeux d'apprentissage et de test. Par ailleurs l'objectif de nos modèles n'est pas de se substituer totalement à l'humain mais de faciliter sa tâche et de lui faire gagner du temps. Nous parlerons donc régulièrement dans cette section de nombre d'observations à effectuer pour observer toute les cryptes déficientes car cela nous semble être un indicateur de l'apport de notre modèle par rapport à une approche naïve consistant à observer toutes les cryptes intestinales afin de déterminer lesquelles sont déficientes. Pour cela, nous traitons notre problème de classification comme un problème de classement (*Learning To Rank*) [6]. Ainsi en associant à chaque image un score de déficience, le médecin ou le laborantin en charge de la détection n'aura à observer qu'un nombre réduit de cryptes intestinales (celles qui ont le score le plus élevé). Nous cherchons donc à associer à chaque image $i \in \mathcal{I}_{test}$ un score de déficience compris entre 0 et 1, où 1 correspond au risque maximal de déficience et 0 correspond au risque minimal de déficience. Il existe un grand nombre de mesure de performances pour évaluer la qualité de la prédiction d'un modèle de type *Learning To Rank* [6]. Nous utiliserons la mesure Average Precision *(*AP*)* qui est une mesure usuelles pour l'évaluation de modèle de *Learning To Rank* [6] et d'extraction d'information. Celle-ci nous donne donc une idée globale la qualité du modèle évalué. Nous introduisons également deux mesure de performance, la Précision à Rappel Maximal *(*PRM*)* et le Taux de Faux Négatif à Rappel Maximal *(*TFPRM*)* qui nous semble particulièrement approprié à l'évaluation de modèle dans notre cadre. Par la suite nous noterons donc $\mathcal{I}_{test}$ le jeu de test contenant $\mathcal{I}_{def}$ et $\mathcal{I}_{norm}$ les images du jeux de test, respectivement déficientes et non déficientes. Nous noterons également $\mathrm{S}(i), \forall i \in \mathcal{I}_{test}$ le score associé à l'image i pour le modèle évalué et $\mathrm{S}(\mathcal{I})$ l'ensemble des scores associés aux images de $\mathcal{I}$.

3.5.1 Average Precision

Avant de définir la mesure *Average Precision* nous rappelons les notions de rappel et de précision, qui nous seront d'ailleurs utiles pour les autres mesures de performance. La précision correspond dans le cadre de l'apprentissage automatique (pour problème binaire, i.e $y_{test} \in \{0, 1\}$) au nombre d'observations correctement attribuées à la classe positive rapportée au nombre de nombre total d'observation attribuées à la classe positive. Plus particulièrement dans notre cas, où la classe positive représente les cryptes déficientes et la classe négative les cryptes intestinales non déficientes. Notre modèle retourne un score compris entre 0 et 1 pour chaque image de crypte intestinale, on définit donc un seuil τ qui distinguera les cryptes intestinales potentiellement déficientes des cryptes intestinales non-déficientes (celles ayant un score supérieur à τ sont potentiellement déficientes et celles qui ont un score supérieur sont potentiellement non-déficiente). La précision du modèle est alors

$$precision_\tau = \frac{|\{i \in \mathcal{I}_{def} : \mathrm{S}(i) \geq \tau\}|}{|\{i \in \mathcal{I}_{test} : \mathrm{S}(i) \geq \tau\}|}$$

Le rappel correspond lui au nombre d'observations correctement attribuées à la classe positive rapporté au nombre d'observation appartenant réellement à la classe positive. De la même manière que pour le calcul de la précision, il nous faut définir un seuil de décision à partir duquel on classe les images comme déficientes ou comme non déficientes. Le rappel se calcule donc comme

$$rappel_\tau = \frac{|\{i \in \mathcal{I}_{def} : \mathrm{S}(i) \geq \tau\}|}{|\mathcal{I}_{def}|}$$

Ainsi pour chaque seuil $\tau \in [0, 1]$ on a un score de précision et un score de rappel. On pourra alors tracer la courbe $p(r)$ qui correspond à la précision en fonction du rappel. Le score *Average Precision* correspond à l'aire sous cette courbe, i.e

$$AP = \int_0^1 p(r)\,\mathrm{d}r$$

On pourra calculer cette intégrale comme une somme fini en prenant pour seuil tout les valeurs de score dans $\mathrm{S}(\mathcal{I}_{test})$.

$$AP = \sum_{i \in \mathcal{I}_{test}} precision_{\mathrm{S}(i)} \times \Delta recall_{\mathrm{S}(i)}$$

où $recall_{\mathrm{S}(i)}$ représente la différence de rappel entre les images i et $i-1$.

3.5.2 Précision à Rappel Maximal

Comme il serait très indésirable de ne pas détecter toutes les cryptes déficientes nous choisissons un seuil tel que toutes les cryptes déficientes soient considérées comme potentiellement déficiente c'est donc une borne inférieur des scores des cryptes déficientes . Plus particulièrement nous choisirons τ tel que

$$\tau = min(\mathcal{I}_{def})$$

Nous avons donc un rappel maximal

$$rappel_\tau = \frac{|\{i \in \mathcal{I}_{def} : \mathrm{S}(i) \geq \tau\}|}{|\mathcal{I}_{def}|} = 1$$

Nous désirons également diminuer autant que possible le travail de l'observateur, nous voulons donc qu'il ait un minimum de cryptes intestinales non déficientes à observer avant d'avoir observer toutes les cryptes déficientes . Nous proposons donc pour cela d'observer la précision du modèle lorsque le rappel est maximal, i.e. lorsque que l'on choisit un seuil de décision comme une borne inférieur des scores des images de cryptes déficientes .

$$PRM = \frac{|\mathrm{I}_{def}|}{|\{i \in \mathcal{I}_{test} : \mathrm{S}(i) \geq min(\mathcal{I}_{def})\}|}$$

Cela nous indiquera donc la proportion de cryptes déficientes parmi les cryptes classées comme potentiellement déficientes.

3.5.3 Taux de Faux Positifs à Rappel Maximal

Comme nous l'avons expliqué précédemment pour la Précision à Rappel Maximal, il est très indésirable de classer une crypte déficiente comme non déficiente. Nous fixons donc encore le taux de cryptes intestinales non déficientes classées comme déficientes. En revanche nos jeux de données de test étant déséquilibré, il ne nous donne pas d'indication sur la réduction du nombre d'observations à effectuer en moins sur l'ensemble du jeu de données.

Nous introduisons pour cela, le *Taux de Faux Négatif à Rappel Maximal* qui correspond au taux de cryptes intestinales non-déficientes ayant un score supérieur à la crypte déficiente ayant le score le plus bas. Plus formellement, On considère un ensemble d'images de cryptes déficientes $\mathcal{I}_{def} \in \mathcal{I}_{test}$ et un ensemble d'images de cryptes non-déficiente $\mathcal{I}_{norm} \in \mathcal{I}_{test}$, où $\mathcal{I}_{test}$ est le jeu de test tel que $\mathcal{I}_{def} \bigcup \mathcal{I}_{norm} = \mathcal{I}_{test}$. On alors

$$TFPRM(\mathcal{I}_{test}) = \frac{|\{i \in \mathcal{I}_{norm} : \mathrm{S}(i) \geq min(\{\mathrm{S}(j) : \forall j \in \mathcal{I}_{def}\})\}|}{|\mathcal{I}_{norm}|}$$

L'intérêt de cette mesure est qu'elle nous donne une indication relativement intuitive de la diminution du travail à fournir pour observer toutes les cryptes déficientes. Un taux d'erreur de 0.2 signifiera, par exemple, qu'il faudra parcourir 20% des cryptes intestinales non-déficientes avant d'avoir observé toutes les cryptes déficientes. On divise donc en théorie le travail de l'observateur par 5 (le nombre de cryptes déficientes étant négligeable devant le nombre de crypte intestinale non-déficientes).

3.5.4 Mesures de performance à k

Nous souhaitons ici décrire un type de métrique qui nous semble très intéressant même si nous n'avons pas eu le temps de l'utiliser dans nos résultats car nous l'avons envisagé trop tard. Nous pensons que celle-ci devrait faire parti des métriques utilisées par la suite pour l'évaluation du modèle de *ranking*.

L'intérêt de ces mesures est qu'elles sont plus robustes aux valeurs extrêmes (voir [19] et [16]). En effet si le modèle utilisé est capable de bien ranger 10 des 12 *aberrant crypt foci* présente sur la lame analysé nous voulons considérer que celui à était performant ce qui ne sera pas le cas si on regarde la Précision à Rappel Maximal et qu'une crypte est très mal rangée. Il suffit donc pour cela de n'observer que les k images les mieux classé et d'analyser les différentes métriques proposées précédemment.

4. Résultats

Nous discutons dans cette section des performances qualitatives et quantitatives que nous avons effectué afin d'évaluer notre modèle de *ranking* décrit dans la section 3.4. Nous discuterons dans un premier temps des résultats du modèle d'extraction de features puis du modèle de ranking dans sa globalité.

4.1 Performances du modèle d'extraction de features

Notre processus de classification étant scindé en deux grandes étapes, il nous semble important que chacune d'elle soit évalué individuellement dans un premier temps et ceux principalement pour l'étape d'extraction de features. Nous analyserons les performances de ce modèle en comparant les résultats obtenus pour les deux modèles de base (**InceptionResNetV2** [41] et **MobileNetV2** [36]) en fonction de plusieurs hyperparamètres que nous avons considérés comme potentiellement décisifs pour optimiser les performances du modèle. L'optimisation des hyperparamètres d'un modèle de *machine learning* est un tâche longue et fastidieuse et même s'il existe aujourd'hui des méthodes se basant sur des outils statistiques pour la rendre la plus rigoureuse possible, cette étape fait souvent appel à l'intuition du développeur concernant les modèles utilisés.

Nous estimons que l'un des paramètre critique pour notre problème est le nombre de cryptes déficientes présentes dans jeu d'apprentissage. La structure de notre modèle a été pensé de telle sorte qu'elle nous permette d'avoir des performances correctes en utilisant un minimum d'images de cryptes déficientes lors de la phase d'apprentissage du modèle aussi nous nous attendons à ce que les performances du modèle converge relativement vite avec le nombre de cryptes intestinales utilisées dans le jeu d'apprentissage. Lors de l'évaluation de modèle de *machine learning*, il est souvent porté une grande attention au nombre d'epochs pour la phase d'entraînement afin de ne pas rentrer en phase de sur-apprentissage. Enfin, du fait que nous utilisons un modèle pré-entraîné, il nous faut déterminer les couches du modèle de base que nous voulons ré-entraîner, nous évaluerons donc également l'impact de cet hyperparamètre sur les performances du modèle d'extraction de features.

4.1.1 Convergence du modèle d'extraction de features

L'un des facteurs critique de notre modèle est le nombre d'image déficientes utilisées pour entraîner le modèle.

Comme on peut le voir (figure 4.1), le modèle InceptionResNetV2 converge très lentement et demande un grand nombre d'exemple d'*aberrant crypt foci* pour obtenir des résultats moins intéressant que ceux que permettent un même nombre d'*aberrant crypt foci* avec le modèle MobileNetV2 (voir figure 4.2).

Pour cette raison nous avons préféré utiliser le modèle MobileNetV2 (celui présentant également l'avantage d'être plus rapide à entraîner) pour la suite de nos expérimentation.

4.2 Performance du modèle de pairwise ranking

Nous détaillons dans cette section les performances du modèle de *pairwise ranking*. Nous comparons tout d'abord ce modèle avec celui d'extraction de features afin de déterminer si celui montre un intérêt. Nous évaluons ensuite les performances de ce modèle en fonction de différents paramètres.

4.2.1 Intérêt du modèle proposé

Nous comparons dans cette section les performances de notre modèle avec le modèle d'extraction de features pour lequel nous laissons le perceptron de prédiction. Si notre modèle de *pairwise ranking* n'est pas en mesure de montrer des performances significativement supérieures à celle du modèle d'extraction de features alors son utilisation ne sera plus justifier et nous considérerons que notre modèle n'est pas intéressant.

Comme on peut le voir sur la figure 4.2, notre modèle montre des performance supérieur au modèle d'extraction de features tant que le nombre de cryptes déficientes faisant parti du jeu d'apprentissage est inférieur à 20. Notre

modèle (nb *ACF* in train)	*AP*	*PRM*	*TFPRM*
IRNV2 (5)	0.91504	0.081187	0.956333
IRNV2 (20)	0.921336	0.126961	0.856333
IRNV2 (100)	0.936747	0.305882	0.518333

FIGURE 4.1 – Comparaison du modèle d'extraction de features avec notre modèle de *pairwise ranking*

modèle (nb *ACF* in train)	*AP*	*PRM*	*TFPRM*
MNV2 (1)	0.917932	0.071495	1.0
MNV2 + *pairwise ranking* (*PWR*) (1)	**0.947119**	**0.117081**	**0.580667**
MNV2 (5)	0.913865	0.073797	0.949667
MNV2 + *PWR* (5)	**0.938719**	**0.076508**	**0.913333**
MNV2 (10)	0.975043	0.126785	0.509667
MNV2 + *PWR* (10)	**0.986873**	**0.223790**	**0.256667**
MNV2 (20)	**0.997231**	**0.762590**	**0.022**
MNV2 + *PWR* (20)	0.996133	0.619883	0.0433333
MNV2 (100)	**0.990271**	**0.963504**	**0.001667**
MNV2 + *PWR* (100)	0.997561	0.956522	0.002

FIGURE 4.2 – Comparaison du modèle d'extraction de features avec notre modèle de *pairwise ranking*

modèle montre donc des performances intéressantes tant que le nombre d'image de cryptes déficientes est très faible. En revanche, dès lors que l'on dispose d'un nombre suffisamment grand l'intérêt du modèle décrit dans ce rapport décroît fortement. Ainsi nous testerons particulièrement notre modèle dans les conditions où il est le plus performant car nous considérons que les performances d'un modèle de convolution plus classique (comme notre modèle d'extraction de features) sont mieux documentées et donc moins intéressantes.

4.2.2 Importance du modèle de régression utilisé

Une des raisons pour lesquelles nous avons choisit de décomposer notre modèle en plusieurs étapes, à savoir une étape d'extraction de features et une étape pour le *pairwise ranking*, est qu'une fois les features des images extraites, il nous est plus facile de tester différent modèle pour la phase de *pairwise ranking*. Plus particulièrement nous voulions être en mesure de tester plusieurs modèle de régression pour notre modèle. Cette réflexion vient du fait que notre jeu de données est très déséquilibré et il nous semble alors qu'un modèle utilisant une méthode de régularisation sera plus performant. Nous évaluons donc par la suite les performances de notre modèle de *pairwise ranking* en utilisant différents modèles de *Régression Linéaire*.

Nous comparons donc quatre modèles de *Régression Linéaire* avec différentes régularisations, un modèle de *Régression Linéaire* classique, un modèle de *Régression Linéaire* avec régularisation L_1 dit modèle Lasso, un modèle Elastic-Net (avec un ratio de régularisation L_1 de $\frac{1}{2}$) et un modèle de *Régression Logistique ou Bayésienne*. Les modèles utilisés sont ceux proposés par la librairie *Python scikit-learn* avec les hyperparamètres par défaut.

Modèle de régression	AP	PRM	TFPRM
Linear	0.922554	0.245304	0.227667
Elastic Net	0.942145	0.121378	0.535667
Lasso	0.940942	0.110945	0.593000
Bayesian	0.955613	0.188936	0.317667

FIGURE 4.3 – Performance modèle *pairwise ranking*. Modèle de base : MobileNetV2 (MNV2) [36], epochs : 40, 1ère couche de ré-entraînement : 100, nombre d'*aberrant crypt foci* dans le jeu de train : 10

Nous avons observé (figure 4.3) que le modèle de régression bayesien montre de très bonnes performances globales. Nous expliquons cela par le fait que notre jeu de donné étant très fortement déséquilibré, le modèle de régression bayesien semble moins susceptible au sur-apprentissage. Cela reste à formaliser et demanderai de nouveaux tests pour déterminer si cette observation se confirme. Les méthode de régularisation Elastic Net et Lasso sont intéressantes en terme d'Average Precision mais semblent moins robustes car les scores qu'ils entraînent semblent plus dispersé comme on le voit avec la métrique Taux de Faux Négatif à Rappel Maximal.

Nous considérons donc par la suite le modèle de régression comme modèle par défaut. Nous pouvons voir (figure 4.4), les performances du modèle semble bien augmenter avec le nombre d'*aberrant crypt foci* dans le jeu d'apprentissage mais restent intéressante y compris pour un nombre d'*aberrant crypt foci* très faible (inférieur à 10).

Modèle - nbr déficientes train/ mesures	*AP*	*PRM*	*TFPRM*
PWR bayesian - 5	0.941394	0.353033	0.138667
PWR bayesian - 20	0.951915	0.963014	0.124
PWR bayesian - 100	**0.998934**	**0.985075**	**0.000667**

FIGURE 4.4 – Performance modèle de régression bayesien. Modèle de base : MobileNetV2 (MNV2) [36], epochs : 40, 1ère couche de ré-entraînement : 100

4.2.3 Distribution des cryptes

Cette section propose une analyse plus qualitative de notre modèle. Définir les mesures de performances les plus adaptées à un problème donné est rarement une tâche facile, celles-ci étant rarement représentative de la qualité globale d'un modèle. Il faut donc en général observer un ensemble de métriques afin de se faire une idée de la qualité du modèle obtenue. C'est ce que nous avons cherché à faire en proposant deux mesures (la Précision à Rappel Maximal et le Taux de Faux Négatif à Rappel Maximal) qui nous semblaient adaptées à notre problème de *ranking*. Il nous a tout de même semblé pertinent d'évaluer nos modèle de manière plus qualitative en observant les histogrammes des scores obtenus durant la phase de test. Ceux-ci nous semblent particulièrement pertinent pour l'évaluation du modèle car ils condensent une grande quantité d'informations à la fois globales (ce que font bien nos métriques) mais également locales. L'histogramme 4.5 présente selon nous une distribution des scores très intéressante car la zone ou les scores des *aberrant crypt foci* et des cryptes saine et relativement réduite. Par ailleurs les deux distributions semblent suivre une loi de poisson dont la moyenne se trouvant pour la première aux alentours de 0.7 pour les *aberrant crypt foci* et de 0.2 pour les cryptes saines.

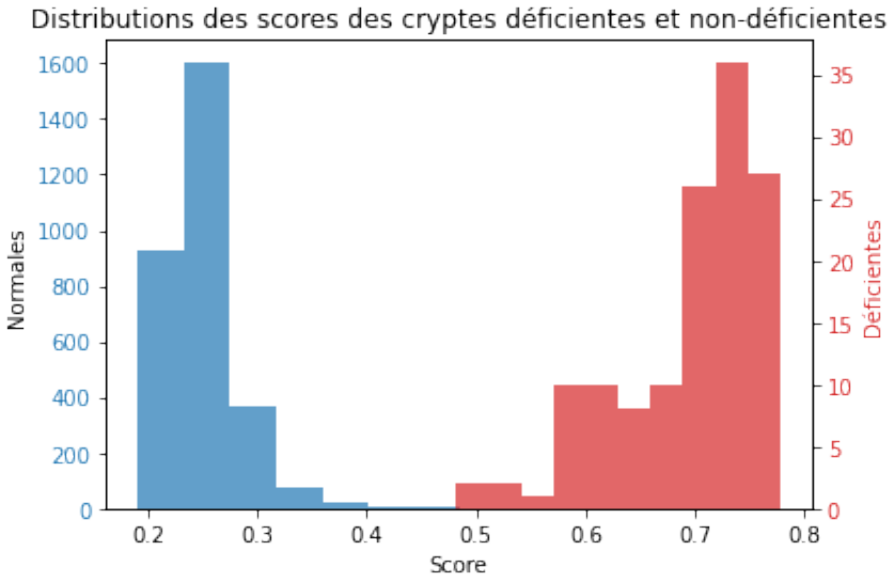


FIGURE 4.5 – Distribution des scores de déficiences

Après avoir entraîné notre modèle de Few Shot Learning sur le jeu d'apprentissage, on évalue ses performances sur le jeu de test avec les métriques présentées précédemment. Le jeu de test est composé ici de cryptes intestinales provenant de la même lame que celles du jeu d'apprentissage. On obtient alors pour chaque crypte du jeu de test sa probabilité d'être déficiente. On peut alors représenter la distribution des cryptes intestinales en fonction de leur probabilité. On obtient donc l'histogramme ci-dessus qui montre le nombre de cryptes intestinales qui ont une même probabilité pour chaque probabilité.

On voit qu'on peut séparer les cryptes intestinales en deux groupes en fonction de leur probabilité : d'un côté les cryptes intestinales saines sur la gauche et les cryptes déficientes sur la droite. En particulier, les cryptes intestinales saines se concentrent sur une probabilité de 0.15 alors que les déficientes se regroupent autour d'une probabilité de 0.7. On observe aussi un seuil de probabilité environ égal à 0.38 à partir duquel une crypte intestinale est considérée comme déficiente par notre modèle.

Comme notre modèle parvient bien à séparer les cryptes intestinales saines des déficientes, ces résultats montrent

que notre modèle est efficace dans ce cas là

4.2.3.1 Tests sur des cryptes d'une autre lame

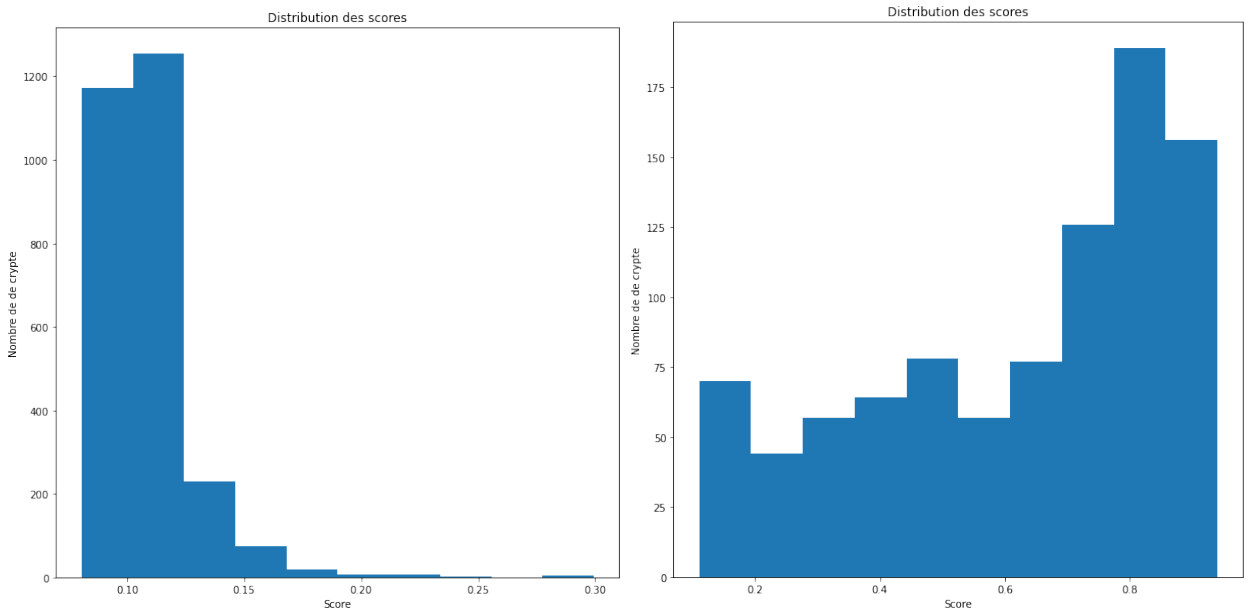


FIGURE 4.6 – Distribution des scores pour deux lames distinctes

Désormais on teste les performances du modèles sur un jeu de test issu d'une lame différente de celle qu'on a utilisé lors de l'apprentissage. Celui-ci n'est pas annotée. On remarque cette fois une distribution avec un seul groupe de cryptes intestinales et non plus deux comme avant : toutes les cryptes intestinales ont été classées comme déficientes et aucune comme saine. Le seuil à partir duquel on considère une crypte déficiente change aussi.

Cela s'explique par le fait que les cryptes intestinales diffèrent considérablement d'une lame à une autre. Par exemple leur coloration (plus ou moins bleue) peut changer pour des raisons techniques comme à cause de la configuration du microscope.

Donc si on entraîne le modèle seulement sur des cryptes intestinales d'une seule lame, il aura du mal à généraliser sur les autres qui sont très différentes. On en déduit donc que notre modèle sur-apprend.

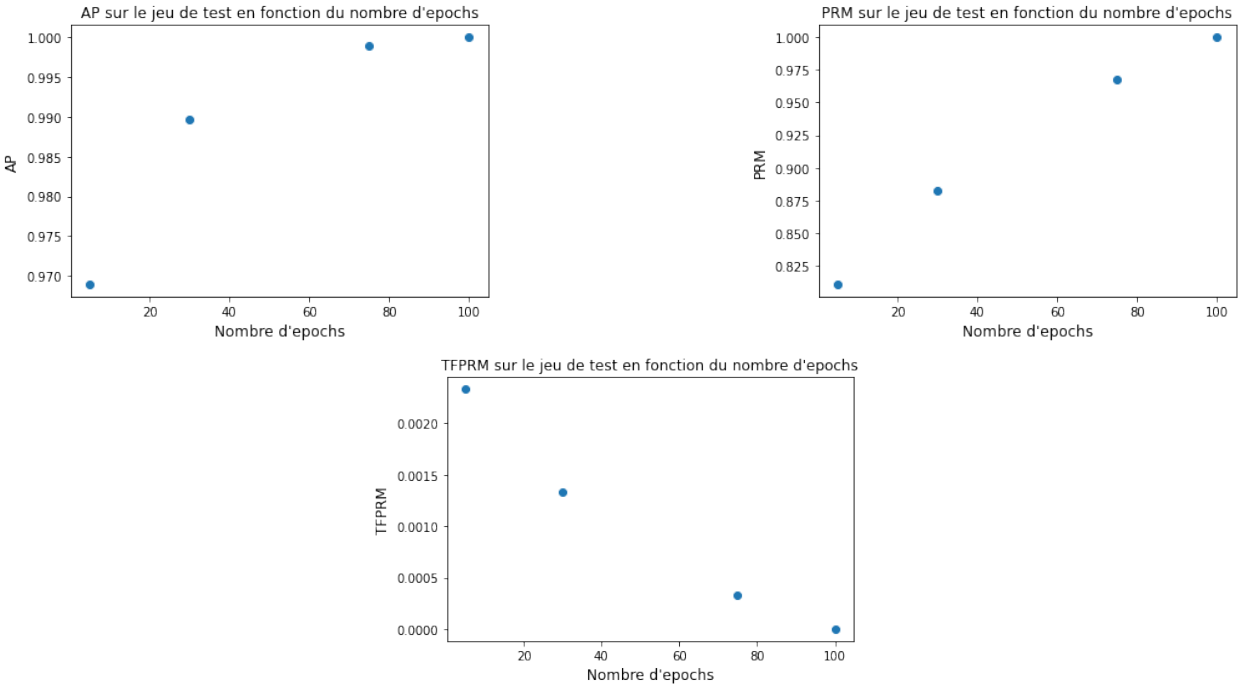
Cela soulève un problème important qu'il faudra chercher à résoudre : trouver un seuil fixe à partir duquel on considère une crypte intestinale comme étant déficiente qui ne varie pas d'une lame à l'autre et qui ne diminue pas les performances du modèle. Un seuil fixe est crucial pour que notre modèle soit exploitable.

4.2.4 Tests sur les hyperparamètres

Nous faisons maintenant varier différentes hyperparamètres pour étudier leur influence sur les performances du modèle. A chaque fois, nous faisons varier un seul hyperparamètre et fixons tous les autres à une valeur constante. Nous effectuons les tests sur des cryptes intestinales appartenant à la même lame que celle qu'on a utilisée pendant l'apprentissage. En effet, nous disposons de très peu de lames annotées.

4.2.4.1 Test sur le nombre d'epochs

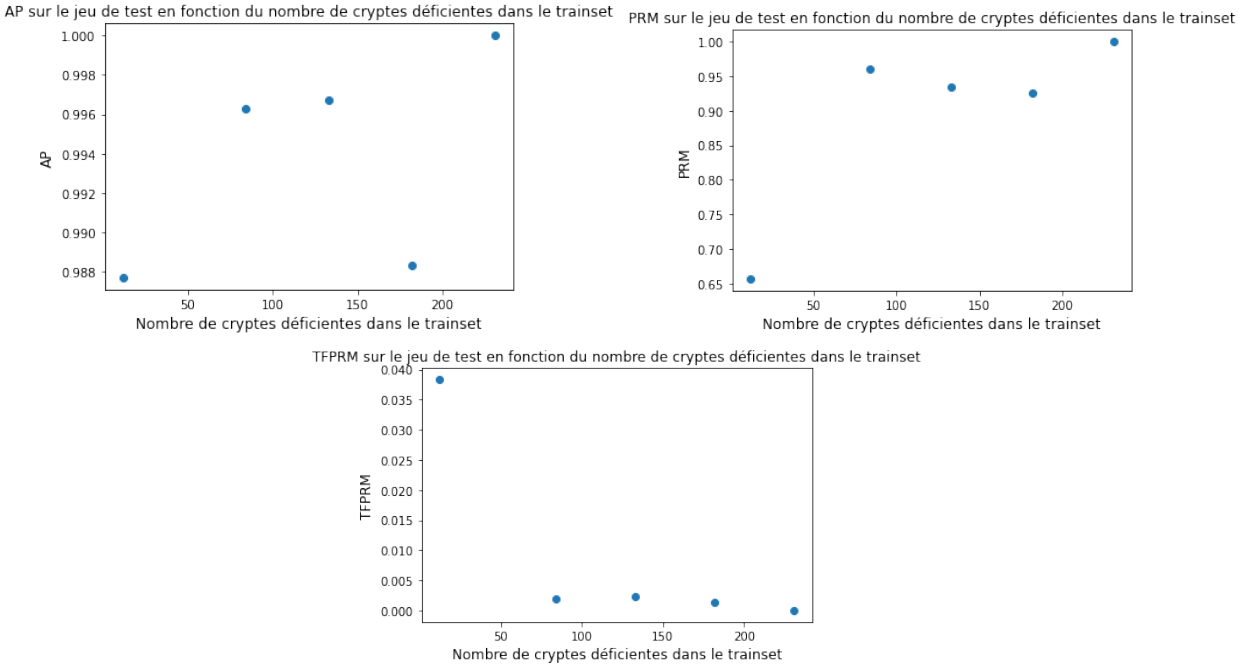
Nous étudions l'influence du nombre d'epochs sur les performances du modèle. Le nombre d'epochs étant le nombre de fois où on présente le jeu d'apprentissage à apprendre au modèle, autrement dit le temps dont dispose le modèle pour apprendre. On s'attend à ce que les performances du modèle augmentent avec le nombre d'epochs.



On observe que plus le nombre d'epochs augmente, plus notre modèle devient efficace : les précisions augmentent et le taux de faux positif diminue. C'est un résultat prévisible et cohérent. Cependant, les précisions convergent vers 1 et le taux de faux positif vers 0. Cela montre que notre modèle sur-apprend quand le nombre d'epochs est trop grand. En effet, si ses performances sont très élevées quand il est évalué sur la lame qu'il a apprise, elles seront faibles avec d'autres lames qu'il n'a jamais rencontré.

4.2.4.2 Tests sur la composition du jeu d'apprentissage

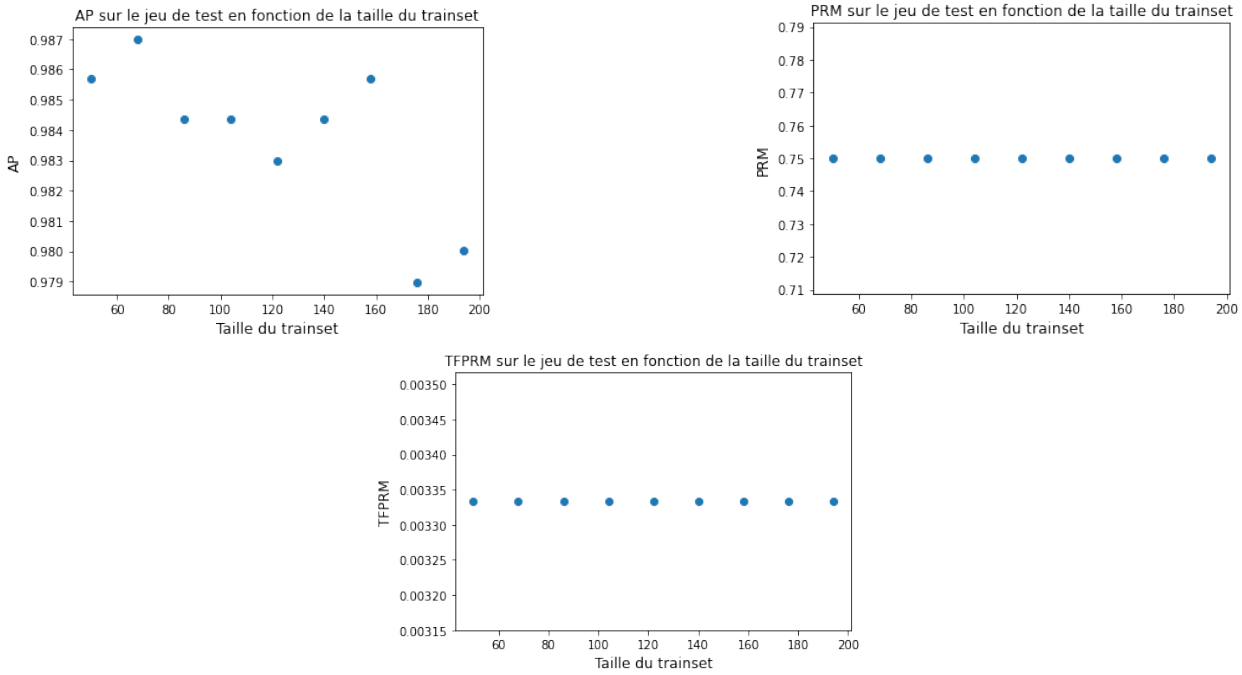
Nous faisons maintenant varier la composition du jeu d'apprentissage. Celui-ci est constitué à la fois de cryptes saines et de cryptes déficientes. On peut donc jouer sur le nombre de cryptes déficientes dans le jeu d'apprentissage. On s'attend à ce que les performances du modèle augmente avec le nombre de cryptes déficientes qu'il apprend.



Malgré quelques valeurs étranges, globalement, on voit que les performances du modèle augmente plus il apprend de cryptes déficientes.

4.2.4.3 Tests sur la taille du jeu d'apprentissage

Enfin on étudie l'influence du nombre de données apprises par le modèle sur ses performances. On s'attend encore une fois à ce que le modèle s'améliore plus on lui fournit de données à apprendre.



On obtient des résultats étranges et problématiques : des fluctuations aléatoires et des valeurs constantes. Cela peut s'expliquer par le fait que nous n'avons pas réalisé des Validation Croisée à k-blocs. C'est un phénomène qu'il faudrait approfondir par la suite.

4.3 Discussion des résultats

Malgré le fait que la phase de test du modèle n'est pas été achevée, nous constatons tout de même deux choses intéressantes après cette évaluation de notre algorithme de *pairwise ranking*.

Tout d'abord il semble qu'un modèle d'apprentissage statistique est tout à fait pertinent pour résoudre le problème de *ranking* des cryptes intestinales pour la détection d'cryptes déficientes. Le modèle proposé montre en effet des performances plus que correcte sur l'ensemble des mesures de performance que nous avons regardé.

De manière plus qualitative, la distribution des scores de déficience qu'il permet semble très prometteuse. Cela étant d'autant plus remarquable que même avec un nombre d'cryptes déficientes très réduit (inférieur à 10) le modèle garde sa pertinence en affichant des résultats très corrects. Ce modèle a d'ailleurs été conçu avec cet objectif, le nombre d'cryptes déficientes à notre disposition au début de l'étude était encore plus réduit (seulement 32) et il était alors inenvisageable de s'orienter vers une méthode classique de *ranking* demandant en général un grand nombre de représentants de chaque classe. Nous sommes donc satisfait de ces résultats.

D'autre part, il semblerait (figure 4.2) que notre modèle de *pairwise ranking* devient moins intéressant à mesure que le nombre d'cryptes déficientes disponible pour l'entraînement augmente. Cette approche était donc pertinente dans un premier temps mais cela ne semble plus être les cas et nous pensons qu'il serait préférable de s'orienter vers des méthodes de *ranking* plus classique en envisageant par exemple de créer de nouvelles cryptes déficientes en utilisant des Generative Adversarial Networks comme cela a été proposé dans [12].

5. Discussion

5.1 Travaux futurs et Perspectives d'amélioration

5.1.1 Extraction des zones de fort contraste

Un des critères particulièrement important utilisés pour la distinction entre les cryptes déficientes et les cryptes intestinales saines est la couleur. En effet, les cryptes déficientes sont colorées dans des nuances de bleu alors que les cryptes intestinales saines sont colorées dans des tons bruns. De même, le contexte autour d'une crypte intestinale (voisinage de trois fois le rayon de la crypte intestinale) est également important pour déterminer si une crypte intestinale est belle et bien déficiente ou saine. Il peut être intéressant d'extraire les zones de fort contraste au sein d'une image pour différentes raisons :

- diminuer le temps de recherche d'une crypte déficiente
- amélioration des résultats de prédiction d'une crypte déficiente
- prendre en compte le contexte pour certains modèles de classification

Les zones de fort contraste correspondent à des parties de l'image où se trouvent des cryptes déficientes. La recherche des cryptes déficientes est alors simplifiée. Le modèle s'appliquera non plus sur l'ensemble de l'image mais plutôt sur la zone sélectionnée et donc sur le problème cryptes intestinales appartenant à cette zone.

Ce qui permet de diminuer le temps de calcul puisqu'on parcourt seulement une partie de l'image et une partie des annotations. De même, avec cette méthode, nous pouvons supposer que le modèle sera plus performant pour prédire si une crypte intestinale est déficiente ou saine. Le problème d'un jeu de données non représentatif de la population cible, les cryptes déficientes n'en n'est plus un étant donné que nous nous intéressons, ici, à des régions où il y aura nécessairement au moins une crypte déficiente. Nous considérons un voisinage de trois fois le rayon de la crypte.

Certains modèles de classification utilisent des images centrées sur l'objet à classifier en l'occurrence une crypte intestinale dans notre cas. Le contexte et en particulier les cryptes intestinales voisines ne sont donc pas pris en compte (c'est à dire le voisinage de l'objet sur lequel on concentre notre étude). Réaliser cette étape en amont de l'application du modèle permet donc de prendre en compte, par la suite, le contexte et donc la couleur des cryptes intestinales voisines et le contraste d'une zone. Cependant, cette méthode est particulièrement gourmande en temps d'exécution. Sur la plupart des lames l'algorithme a un temps d'exécution relativement élevé. C'est donc une piste qui reste encore à améliorer puisque l'intérêt est d'aller plus vite qu'un spécialiste pour lui faire gagner du temps.

Cette étape d'extraction de zone de contraste survient après avoir détecté les cryptes intestinales avec **Yolov3** [21] ou un autre modèle (segmentation des cryptes intestinales). En effet, cet algorithme construit le graphe à partir d'un fichier d'annotation sur des cryptes intestinales (déficientes ou saines). Il doit avoir en entrée les annotations avec les coordonnées des cryptes intestinales de l'image, et les images elles même en couleur. Une partie du code de *Clémence Lanfranchi* a été réadapté pour implémenter l'extraction de la zone de contraste et donc analyser le contexte.

5.1.2 Critères d'évaluation des Cryptes Déficientes

Dans cette section [10], nous allons plutôt se concentrer sur les caractéristiques histologique des cryptes de Lieberkühn (ou cryptes intestinales) qui sont sujettes à des malformations visibles sur les images obtenues par microscopie avec teinture des cellules intestinales au bleu de méthylène qui sont soumises à étude, dépeinte dans ce rapport de projet.

Ces cryptes déficientes ou dites aberrantes sont appelées les *aberrant crypt foci* ou *ACF* (en français cryptes déficientes) . Le nombre d'*ACF* chez les patients atteints d'adénomes était significativement corrélé au nombre d'adénomes.

Afin d'affiner la classification des cryptes et de trouver d'autres critères que la coloration bleu des cryptes pour les considérées comme déficientes, on présente des caractéristiques histologiques plus précises et visibles et/ou quantifiables sur nos images.

1. **Taille et nombre de noyau** : Le noyau sain est ovoïde dans les myofibroblastes sub-épithéliaux .Son diamètre est de quelques micromètres (5 à 6 µm), et il représente en général moins de 10 % du volume cellulaire. Le rapport nucléo cytoplasmique est le rapport ($\frac{volume\_du\_noyau}{volume\_du\_cytoplasme}$). Par conséquent, dans l'analyse cytologique de l'altération nucléaire des cellules, **si ce rapport est $\geq$ à $\frac{1}{9}$ comme dans la figure 5.1 et que le noyau occupent plus de 35 % de l'espace cellulaire, on a une *ACF* => Syndrome de Lynch** [33].

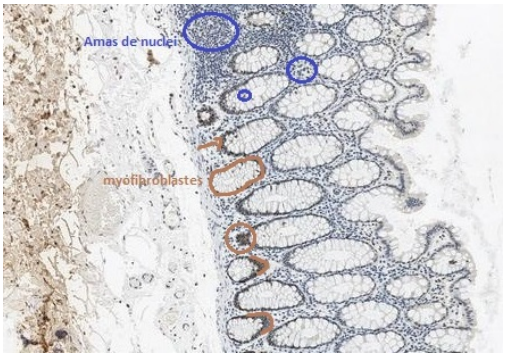
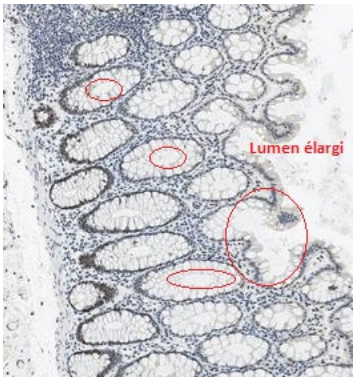
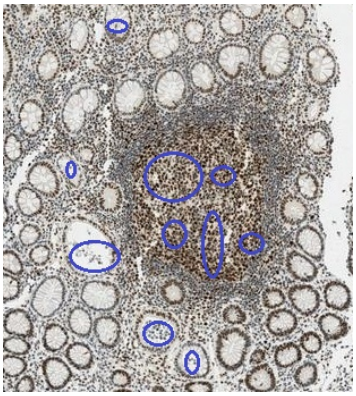


FIGURE 5.1 – Image obtenu par microscopie avec des myofibroblastes ayant des noyaux élargis, les ellispses entourant les amas de nuclei

2. **Taille et forme du *lumen*** : Pour évaluer l'altération du *lumen* des cryptes déficientes, chaque *ACF* a été classée comme *lumen* modérément altérée si le contour est **elliptique**, figure **5.2a** ou gravement altérée s'il présente des **fentes irrégulières et est cribriforme** comme on peut le constaté sur la figure **5.2b** (trou dans le *lumen* comme si était criblé de balles) [42].



(a) Lumen en forme d'ellipse ou arrondis



(b) Lumen cribiformes

FIGURE 5.2 – Altération du Lum sur des crypte déficiente

3. **Nombre des cellules caliciformes ou cellules de Gobelet** :Pour un intestin sain, ces cellules occupent 9.5 % de la surface des entérocytes. Dans l'analyse de la réduction des cellules caliciformes ou cellules de Gobelet chaque *ACF* a été classée légèrement, modérément ou sévèrement altéré selon, si la réduction du nombre de cellules caliciformes ou cellules de Gobelet est respectivement de plus **10 % par rapport à celles qui entourent les cryptes normales** [14].

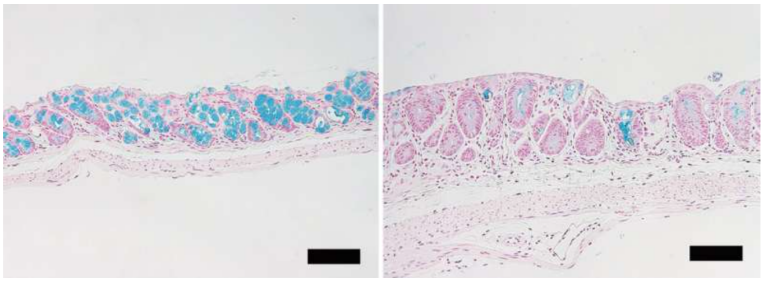


FIGURE 5.3 – A droite une crypte saine avec de nombreuses cellules caliciformes ou cellules de Gobelet et à gauche une *ACF* avec des cellules caliciformes ou cellules de Gobelet clairsemées

5.2 Ethique de l'IA appliqué à la Médecine

Selon une récente enquête menée au Royaume-Uni, 63 % de la population adulte n'est pas à l'aise avec l'utilisation de données personnelles pour améliorer les soins de santé [11]. Une autre étude, menée en Allemagne, a révélé que les étudiants en médecine – les médecins de demain – adhèrent massivement à la promesse de l'intelligence artificielle ou *Artificial Intelligence* (*IA*) d'améliorer la médecine (83 %) mais sont plus sceptiques quant à l'établissement de diagnostics concluants, par exemple lors d'examens d'imagerie (56 % en désaccord) [31]. Ces données d'enquête font écho avec les enjeux éthiques et réglementaires qui entourent l'intelligence artificielle dans les soins de santé, en particulier **la confidentialité, l'équité, la responsabilité et la transparence**.

5.2.1 La collecte de données doit respecter les exigences en matière de protection des données et de confidentialité

Les algorithmes *machine learning*, *ML* utilisent des données soumises à des mesures de protection de la vie privée notamment en France par le Commission nationale de l'informatique et des libertés ou *Commission nationale de l'informatique et des libertés* (*CNIL*) ou bien le Règlement général sur la protection des données ou *règlement général sur la protection des données* (*RGPD*), ce qui oblige les développeurs à prêter une attention particulière aux restrictions éthiques et réglementaires à chaque étape du traitement des données. La provenance des données et le consentement à l'utilisation et à la réutilisation revêtent une importance particulière, en particulier en *machine learning* qui exige des quantités considérables et une grande variété de données. En effet, dans le cadre de notre projet l'ensemble des images recueillies par microscopie sont relativement nombreuses (dans la moyenne pour un projet de cette taille et pour du traitement de l'image étudiant un phénomène rare) dans le but d'avoir de meilleurs résultats mais derrière chaque image se cache une personne.

Il est très probable que ces données disparates auront des conditions d'utilisation différentes et/ou seront assujetties à des protections juridiques différentes. Un exemple frappant est le *RGPD*, récemment adopté, qui fixe des exigences spécifiques en matière de consentement éclairé pour l'utilisation des données et accorde aux personnes concernées plusieurs droits qui doivent être respectés par les personnes qui traitent leurs données. Ce consentement a été récolté à l'hôpital, dans le cadre de notre projet. Les données utilisées pour former des algorithmes doivent avoir les autorisations d'utilisation nécessaires, mais il n'est pas aisé de déterminer quelles utilisations sont permises pour une fin donnée. Cela dépendra également du type de données, de la compétence, de l'objet de l'utilisation et des modèles de surveillance [45].

5.2.2 Le développement des méthodes de Machine Learning doivent faire preuve d'équité pour le traitement des données

Les bases de données sur lesquelles les modèles de *machine learning* (en français apprentissage automatique) sont entraînés et validés sont essentiels pour garantir l'utilisation éthique des algorithmes prédictifs. Des jeux de données d'entraînement peu représentatifs peuvent introduire des biais dans ceux-ci. Le biais possède au moins deux archétypes communs dans les données médicales. Tout d'abord, les sources de données elles-mêmes ne reflètent pas la véritable épidémiologie au sein d'un groupe démographique donné, par exemple les données démographiques biaisées par le sur-diagnostic de la schizophrénie pour les personnes issues d'Afrique du Nord. Pour le Syndrome de Lynch il y a un sur diagnostique pour les femmes caucasienne de plus de 70 ans (prévalence du syndrome chez celles-ci). Deuxièmement, un algorithme est formé à partir d'un ensemble de données qui ne contient pas suffisamment de membres d'un groupe démographique donné – par exemple, un algorithme formé principalement à partir de données provenant de femmes de plus de 70 ans. Un tel algorithme ferait de mauvaises prédictions, par exemple, chez les jeunes hommes qui pourraient être atteints [45].

5.2.3 Le traitement de données se doit de satisfaire le critère de transparence

Les techniques de *machine learning* posent les questions éthiques et juridiques les plus difficiles et sont représentées par des algorithmes dits «boîtes noires» non interprétables, dont la logique interne reste cachée même à leurs développeurs. Ce manque de transparence peut empêcher l'interprétation des résultats fondés sur l' apprentissage automatique et, par conséquent, réduire la fiabilité du diagnostic.[32] De plus, la divulgation de détails élémentaires mais significatifs sur le traitement médical aux patients, un principe fondamental de l'éthique médicale, exige que les médecins eux-mêmes saisissent au moins le fonctionnement interne fondamental des dispositifs qu'ils utilisent. Par conséquent, pour que le traitement d'image soit éthique, les développeurs doivent communiquer à leurs utilisateurs finaux, les médecins, la logique générale qui sous-tend les décisions fondées sur l'intelligence artificielle. Un certain degré d'explicabilité peut également être requis pour justifier la validation clinique de la méthode de *machine learning*, *ML* dans les études prospectives et les essais cliniques randomisés. [45]

5.3 Impact Social

Tout d'abord, l'impact négatif sur les patients créé par le dépistage Lynch devient plus évident lorsqu'on considère la population testée et étant donné qu'un projet comme le nôtre est réalisé dans le but d'accélérer et de démocratiser le dépistage de ce syndrome. En effet, environ 50 % à 60 % de tous les patients atteints d'un cancer colorectal, *CCR* nouvellement diagnostiqué, notamment par technique utilisant le *traitement de l'image* (*IP*) (de l'anglais *image processing*) suivront une chimiothérapie multiagent, soit à court terme, soit à vie (6 mois pour le stade III [≈30 %] et tous les stades IV [≈20 %]), et environ 30 % mourront d'un cancer colorectal, *CCR* d'ici 2 à 3 ans[29].
Pour éviter cela, les sociétés scientifiques et les organismes de réglementation doivent élaborer des pratiques exemplaires pour reconnaître et réduire au minimum les effets en aval des ensembles de données biaisés sur l'apprentissage. Afin d'éviter les préjudices engendrés par des ensembles de données obtenu par des apprentissages biaisés, les critères de qualité de la **FDA** (Food and Drug Administration) pourraient être élargis pour couvrir le risque de biais, tels que la *Vérification et Validation*, i.e, les éléments liés à la compréhension de la criticité et incidence sur la sécurité des patients en fournissant une assurance de la conformité aux exigences, une confiance raisonnable dans le fait que le logiciel réponde aux besoins de l'utilisation ou des utilisateurs. Dans les cas de **Yolov3** [21], cet réseau de neurones artificiels est rapide et donne des prédictions avec une excellente précision mais il n'est pas adapté à la détection de petits objets, alors que dans ce projet on étudie des cellules à l'échelle microscopique, par conséquent le critère d'assurance de la conformité des exigences n'est pas respecté.

Dans le cas de décisions médicales entièrement automatisées, le niveau de risque associé à l'intervention peut déterminer s'il faut informer les patients de la présence de technologies fondées sur la intelligence artificielle utilisées pour orienter leurs soins et comment le faire. Le fait de communiquer avec les patients au sujet de l'utilisation des technologies peut accroître leur confiance et leur acceptation, ce qui, selon les données de l'enquête [32], est dû au nombre croissant d'interventions diagnostiques et thérapeutiques s'appuient sur l'*image processing*, *IP*. Néanmoins, l'autonomie des patients dans les processus décisionnels concernant leur santé pourrait être compromise [29]. En effet, dans aucunes étapes du projet les patients ont été acteurs de leur santé. On sera amené à utiliser des méthodes similaires à celles utilisées dans notre projet dans les années avenir ce qui changera la pratique du diagnostic notamment de maladie rare qui aujourd'hui n'est pas réalisable avec la technologie disponible mais le sera peut être avec les progrès de détection et de traitement de l'image comme combinaison de **Yolov4** [2] avec Opencv [18].

5.4 Impact Environnemental

Pour réaliser ce projet, nous avons utilisé des processeur graphique ou *processeur graphique* (*GPU*), ceux-ci permettent d'accélérer considérablement l'exécution de code. Lorsqu'on analyse des données ou effectue des simulations, la plupart des chercheurs s'inquiète du temps nécessaire pour trouver une solution plutôt que de son impact sur l'environnement.

Heureusement, un temps d'exécution réduit en raison d'un matériel plus rapide ou d'optimisations du logiciel en général conduit également à une empreinte carbone plus faible. Ce n'est pas le cas lorsque le "wall-clock" réduit est atteint en sur-cadencement (overclocking) le processeur, ou lors de l'utilisation de super-ordinateurs. L'augmentation de la popularité des langages de script/interprétés comme Python (langage utilisé pour notre projet) et la disponibilité générale des postes de travail à haute performance constituent une menace considérable pour l'environnement [5].

Lorsque vous exécutez un million de noyaux (kernel en Calcul Haute Performance), l'émission de pour faire fonctionner un super-ordinateur dépasse de loin le transport aérien et se rapproche de l'empreinte carbone du lancement une fusée dans l'espace [5]. Évidemment, pour le projet, notre utilisation de *GPU* reste très faible, cependant l'empreinte carbone est considérable sachant que notre projet est considéré comme minime et est non utilisable pour un réel diagnostique donc d'un point de vue écologique, c'est un gaspillage d'énergie.

Conclusion

Dans ce rapport, nous avons présenté les résultats nous paraissant les plus intéressants pour la détection d'*aberrant crypt foci* dites crypte déficiente. Reprenant le travail proposé par Clémence Lafranchi, nous avons décidé de conserver son modèle de détection de cryptes intestinales, à savoir le modèle de détection **YoloV3** [21], car celui-ci semble très performant pour effectuer cette tâche.

Nous proposons en revanche une approche différente pour l'ordonnancement des cryptes intestinales par niveau de risque de déficience. Alors que Clémence Lafranchi proposait dans son rapport une méthode basée sur l'agrégation de quelques critères médicaux pour la distinction des *aberrant crypt foci* et des cryptes saines, nous étudions la possibilité d'appliquer des méthodes d'apprentissage statistique pour faire cette distinction. Nous proposons pour cela un modèle de *pairwise ranking* qui montre des propriétés intéressantes lorsque le nombre de cryptes déficientes est très faible lors de la phase d'apprentissage. Même si l'évaluation de notre modèle n'est pas terminé, il nous semble à ce stade, et étant donné que le nombre de cryptes déficientes labelisées à notre disposition est croissant, que ce modèle n'est plus le plus pertinent pour notre tâche de *ranking* car un modèle du type réseau neuronal convolutif entraîné pour la classification montre de meilleures performances lorsque le nombre d'cryptes déficientes dans le jeu d'apprentissage est supérieur à 20. Nous estimons, néanmoins, que la piste de l'apprentissage statistique pour l'ordonnancement des cryptes déficientes est très prometteuse puisque les réseau neuronal convolutif montre de manière générale de très bonnes performances.

Nous avons par ailleurs exploré plusieurs autres pistes qui pourraient potentiellement améliorer les performances du modèle de *ranking*. Nous proposons dans la section 5.1.2 de nouveaux critères (autre que la coloration au bleu de méthylène) permettant de distinguer les cryptes déficientes des cryptes saines. Il semblerait que la taille et le nombre de noyaux des cellules pourrait être un critère tout à fait pertinent pour déterminer si une crypte intestinale est une cryptes déficientes. Même s'il semble difficile de forcer un réseau neuronal convolutif à regarder ce critère en particulier il pourrait être très intéressant de savoir si ce type de modèle est capable d'inférer ce critère du jeu d'apprentissage – en testant le modèle sur des images en noir et blanc par exemple. Le critère de la couleur semblant le plus adapté dû aux techniques immunohistochimiques de coloration actuelles qui ont fait leur preuve, nous avons proposé dans la section 5.1.1 de détecter les zones contenant potentiellement des cryptes déficientes en utilisant un algorithme repérant les zones de fort contraste de couleur sur les images traitées. Cela permettrait de réduire le nombre de crypte intestinale analysé par le modèle de *ranking* et pourrait ainsi accélérer le traitement des images. Cette algorithme n'est pas encore aboutit et repose pour le moment sur une méthode relativement coûteuse en temps de calcul mais nous estimons que cette méthode optimisée pourrait présenter un apport intéressant pour la détection decryptes déficientes.

Finalement, notre approche orientée apprentissage statistique semble très prometteuse et devrait être poursuivit. Nous ne pouvons pas à ce stade pas affirmer qu'un modèle de *pairwise ranking* est plus pertinent qu'un modèle de réseau neuronal convolutif utilisant du transfert d'apprentissage et il semble que l'intérêt de ce premier diminuerait à mesure qu'un grand nombre de cryptes déficientes labelisées seraient disponibles pour l'entraînement. Une approche intéressante pour améliorer les performances d'un modèle de réseau neuronal convolutif pourrait être d'utiliser des Generative Adversarial Networks ou GAN pour générer des images synthétiques de cryptes déficientes à donner durant la phase d'entraînement. Cette méthode présenté dans [12] nous semble être une piste très intéressante à poursuivre.

Glossaire

Aberrant crypt foci Traduit par crypte déficiente.
☞ Voir crypte déficiente
✍ Pages I, V, 2, 14–17, 21, 25.

Adénome Un adénome est une tumeur bénigne développée au niveau d'une glande (testicule, sein, rien, thyroïde, surrénales, hypophyse…) ou de certaines muqueuses glandulaires (côlon, estomac, rectum, foie, utérus…).
✍ Page 2.

Anamnèse L'anamnèse retrace les antécédents médicaux et l'historique de la plainte, la douleur actuelle du patient (c'est-à-dire l'histoire de la maladie - terme qui n'est pas synonyme d'anamnèse, mais plutôt de remémoration), ainsi que les résultats des différentes explorations déjà faites et les traitements entrepris.
✍ Page 2.

Apprentissage automatique L'étude scientifique des algorithmes et des modèles statistiques que les systèmes informatiques utilisent pour effectuer une tâche spécifique sans utiliser d'instructions explicites, en se basant plutôt sur des modèles et des inférences. Cette discipline est considérée comme un sous-ensemble de l'intelligence artificielle. Les algorithmes d'apprentissage automatique construisent un modèle mathématique fondé sur des données d'échantillon, appelées « données d'entraînement », afin de faire des prédictions ou de prendre des décisions sans être explicitement programmés pour effectuer la tâche
✍ Pages 13, 23.

Apprentissage profond Sous catégorie du *machine learning*, un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires.
☞ Voir *machine learning*
✍ Page 5.

Apprentissage supervisé Une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé. Les exemples annotés constituent une base d'apprentissage, et la fonction de prédiction apprise peut aussi être appelée « hypothèse » ou « modèle ». On suppose cette base d'apprentissage représentative d'une population d'échantillons plus large et le but des méthodes d'apprentissage supervisé est de bien généraliser, c'est-à-dire d'apprendre une fonction qui fasse des prédictions correctes sur des données non présentes dans l'ensemble d'apprentissage..
☞ Voir apprentissage automatique
✍ Page 8.

Average Precision Décrit en détail dans cette sous-section 3.5.1
✍ Pages 13, 16.

Cancer colorectal Le cancer colorectal ou colo-rectal est une tumeur maligne de la muqueuse du côlon ou du rectum. Le cancer colorectal peut toucher tous les segments anatomiques du gros intestin comme le caecum, le côlon ascendant, le côlon transverse, le côlon descendant, le côlon sigmoïde et le rectum mais ne concerne pas le cancer du canal anal qui est une entité distincte
✍ Pages I, 2, 24.

Cellules caliciformes ou cellules de Gobelet Une cellule caliciforme est une cellule en forme de vase allongé spécialisée dans la synthèse du mucus. Ces cellules épithéliales glandulaires sécrètent de la mucine empaquetée dans les granules de sécrétion stockés au pôle apical, puis libérée dans la lumière intestinale pour former le gel de mucus. Elles bordent les glandes exocrines ou composant en partie les épithéliums absorbants, comme celui de l'intestin grêle et du gros intestin.
✍ Pages V, 22.

Commission nationale de l'informatique et des libertés Une autorité administrative indépendante française. La CNIL est chargée de veiller à ce que l'informatique soit au service du citoyen et qu'elle ne porte atteinte ni à l'identité humaine, ni aux droits de l'homme, ni à la vie privée, ni aux libertés individuelles ou publiques.
✍ Page 23.

Comparaison par paires La comparaison par paires est une méthode consistant à comparer des éléments deux à deux. De nombreux travaux théoriques ont démontré que les relations complexes entre divers éléments indépendants pouvaient être étudiées au moyen de comparaisons analogiques binaires, c'est-à-dire, en les

comparant deux à deux. En effet, une telle démarche permet de décomposer le problème posé en réduisant la masse d'informations à acquérir et à intégrer d'une part, en concentrant la réflexion du répondant sur ses composantes essentielles, d'autre part.
✍ Page 7.

Crypte déficiente Des cryptes de Lieberkühn (ou cryptes intestinales) sont disposées à des malformations visibles sur les images obtenues par microscopie avec teinture des cellules intestinales aux bleu de méthylène avec des lésions, précurseurs de la carcinogénèse colorectale, constituées de cryptes grosses et épaisses
✍ Pages I, V, 2–5, 7, 10–22, 25.

Crypte intestinale Cellule principale composant l'intestin.
☞ Voir entérocytes
✍ Pages 3–5, 7, 9–15, 17, 18, 20, 21, 25.

Entérocytes Les entérocytes sont un des quatre principaux types de cellules de l'épithélium intestinal, au sein de la muqueuse intestinale. Ils proviennent de la division asymétrique de cellules somatiques.
✍ Pages I, 22.

Epoch Une epoch consiste à présentée une fois les données du jeu d'apprentissage au modèle lors de la phase d'apprentissage, le nombre d'epochs est donc le nombre de fois où on a présenté le jeu d'apprentissage au modèle.
✍ Pages V, 15–19.

Few Shot Learning Le Few Shot Learning est un type de problèmes d'apprentissage automatique pour lesquels les données d'entraînement contiennent peu d'informations. Cependant, ce modèle d'apprentissage vise à construire d'autres modèles précis avec moins de données d'entraînement. Étant donné que la dimension des données d'entrée est un facteur qui détermine le coût des ressources (par exemple, le coût du temps, les coûts de calcul, etc.), les entreprises peuvent réduire les coûts d'analyse des données et d'apprentissage automatique (ML) en recourant à l'apprentissage en petit nombre.
✍ Pages 10, 17.

Fonction d'Activation Dans le domaine des réseaux de neurones artificiels, la fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel
✍ Page 10.

Image processing Traduit par traitement de l'image.
☞ Voir traitement de l'image
✍ Page 24.

Instabilité des micro satellites La genèse de certains cancers colorectaux est liée à une instabilité génétique induite par la défaillance du système de réparation des mésappariements de l'ADN appelé système MMR (MisMatch Repair). Ce système est composé de 4 gènes appelés MLH1, MSH2, MSH6 et PMS2. Les protéines codées par ces gènes interagissent pour identifier puis corriger les mésappariements de l'ADN qui résultent d'erreurs commises par l'ADN polymérase lors de la réplication de l'ADN. Les microsatellites correspondent à des séquences d'ADN réparties sur l'ensemble du génome (séquences codantes ou non codantes) dont la structure est répétitive. Du fait de cette structure répétitive, ils sont particulièrement sujets aux erreurs de réplication en cas de défaillance du système MMR. Les cancers présentant un tel phénotype sont dits de type MSI (MicroSatellite Instability).
✍ Page 2.

Intelligence artificielle L'intelligence artificielle est l'étude des systèmes capables d'interpréter correctement des données externes, à tirer des enseignements de ces données et à utiliser ces enseignements pour atteindre des objectifs et des tâches spécifiques
✍ Pages 23, 24.

Jeu d'apprentissage Un jeu d'apprentissage est l'ensemble des données qu'on fait apprendre à un modèle d'apprentissage automatique lors de la phase d'apprentissage.
✍ Pages 3, 8–10, 15–19, 25.

Jeu de test Un jeu de test est un ensemble des données sur lequel on évalue les performances du modèle à généraliser sur des données qu'il n'a pas rencontrées lors de la phase d'apprentissage.
✍ Pages 3, 8, 17, 18.

K-means Le partitionnement en k-moyennes (ou k-means en anglais) est une méthode de partitionnement de données et un problème d'optimisation combinatoire. Étant donnés des points et un entier k, le problème est de diviser les points en k groupes, souvent appelés clusters, de façon à minimiser une certaine fonction. On considère la distance d'un point à la moyenne des points de son cluster ; la fonction à minimiser est la somme des carrés de ces distances. Il existe une heuristique classique pour ce problème, souvent appelée méthodes des k-moyennes, utilisée pour la plupart des applications. Les k-moyennes sont notamment utilisées en apprentissage non supervisé où l'on divise des observations en k partitions. Les nuées dynamiques sont une généralisation de ce principe, pour laquelle chaque partition est représentée par un noyau pouvant être plus complexe qu'une moyenne
✍ Page 3.

La transmission autosomique La transmission des caractères génétiques est dite autosomique lorsque ces caractères sont portés sur les chromosomes non sexuels qu'on appelle les autosomes.
✍ Page 2.

Learning To Rank Le machine-learned ranking (MLR) est l'application de l'apprentissage automatique, généralement supervisé, semi-supervisé ou de renforcement, dans la construction de modèles de classement pour les systèmes de recherche d'information. Cet ordre est généralement obtenu en attribuant une score souvent binaire (ppar exemple, pertinent ou non-pertinent) pour chaque élément. Le modèle de classement vise à classer, c'est-à-dire à produire une permutation d'éléments dans de nouvelles listes invisibles, semblable aux classements des données d'entraînement.
✍ Pages 8, 13.

Lumen Traduit par lumiere.
☞ Voir lumiere
✍ Page 22.

Lynch Syndrome Prediction Model Le modèle PREMM5 est un algorithme de prévision clinique qui estime la probabilité cumulative qu'une personne porte une mutation germinale des gènes MLH1, MSH2, MSH6, PMS2 ou EPCAM. Les mutations de ces gènes causent le syndrome de Lynch, un syndrome de prédisposition héréditaire au cancer.
✍ Page 2.

Machine learning Traduit par apprentissage automatique.
☞ Voir apprentissage automatique
✍ Pages 5, 15, 23.

Myofibroblastes Les myofibroblastes sont des fibroblastes (cellule présente dans le tissu conjonctif ; elle est parfois appelée cellule de soutien) possédant la particularité d'exprimer l'actine α-SMA. Ils jouent un rôle important dans la plasticité, la migration et la motilité de la cellule au sein du tissu conjonctif.
✍ Pages I, 22.

One Class Classification En apprentissage automatique, la classification d'une classe, également connue sous le nom de classification unaire ou de modélisation de classe, tente d'identifier les objets d'une classe spécifique parmi tous les objets, principalement en tirant les enseignements d'un ensemble d'entraînement contenant uniquement les objets de cette classe.
✍ Pages 7, 8, 10.

Overclocking Le sur-cadencement, en français, est une manipulation ayant pour but d'augmenter la fréquence du signal d'horloge d'un processeur au-delà de la fréquence nominale afin d'augmenter les performances de l'ordinateur. Le processeur surcadencé exécutera davantage d'instructions par seconde, d'où la réduction du temps d'exécution des programmes. La production de chaleur étant liée au carré de la fréquence, il chauffera aussi davantage, ce qui peut être source d'erreurs ou d'autobridage du processeur. Si elle est trop faible, sa tension d'alimentation le rendra instable. Si elle est trop forte, le composant peut casser prématurément.
✍ Page 24.

Pairwise ranking Traduit par comparaison par paires.
☞ Voir comparaison par paires
✍ Pages V, 7–11, 15, 16, 20, 25.

Perceptron Le perceptron est un algorithme d'apprentissage supervisé de classifieurs binaires (c'est-à-dire séparant deux classes). Il s'agit d'un neurone formel muni d'une règle d'apprentissage qui permet de déterminer automatiquement les poids synaptiques de manière à séparer un problème d'apprentissage supervisé..
☞ Voir réseau de neurones artificiels
✍ Pages 9, 15.

Processeur graphique GPU est une unité de calcul, pouvant être présent sous forme de circuit intégré sur une carte graphique ou carte mère, ou encore intégré au même circuit intégré que le microprocesseur général et assurant les fonctions de calcul d'image, à afficher à l'écran ou à écrire sur mémoire de masse.
✍ Page 24.

Précision à Rappel Maximal Décrit en détail dans cette sous-section 3.5.2
✍ Pages 14, 17.

Règlement général sur la protection des données Le règlement UE 2016/679 du Parlement Européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE dit règlement général sur la protection des données, est un règlement de l'Union européenne qui constitue le texte de référence en matière de protection des données à caractère personnel. Il renforce et unifie la protection des données pour les individus au sein de l'Union européenne.
✍ Page 23.

Régression En mathématiques, la régression recouvre plusieurs méthodes d'analyse statistique permettant d'approcher une variable à partir d'autres qui lui sont corrélées. Par extension, le terme est aussi utilisé pour certaines méthodes d'ajustement de courbe.
✍ Page 8.

Régression Linéaire Un modèle de régression linéaire est un modèle de régression qui cherche à établir une relation linéaire entre une variable, dite expliquée, et une ou plusieurs variables, dites explicatives. En général, le modèle de régression linéaire désigne un modèle dans lequel l'espérance conditionnelle de y connaissant x est une fonction affine des paramètres..
☞ Voir Régression
✍ Pages 11, 16.

Régression Logistique ou Bayésienne La régression logistique ou modèle logit est un modèle de régression binomiale. Comme pour tous les modèles de régression binomiale, il s'agit de modéliser au mieux un modèle mathématique simple à des observations réelles nombreuses. En d'autres termes d'associer à un vecteur de variables aléatoires $(x_1, \ldots, x_K)$ $(x_1, \ldots, x_K)$ une variable aléatoire binomiale génériquement notée y. La régression logistique constitue un cas particulier de modèle linéaire généralisé. Elle est largement utilisée en apprentissage automatique..
☞ Voir Régression
✍ Pages 7, 16.

Réseau de neurones artificiels Le réseau neuronal artificiel ou réseau neuronal simulé est un réseau interconnecté de neurones artificiels (ou naturels en neurosciences) qui utilise un modèle mathématique ou informatique pour le traitement de l'information basé sur une approche connectiviste du calcul
✍ Pages 3, 7–9, 24.

Réseau neuronal convolutif Une variante de réseau de neurones artificiels acyclique majoritairement utilisée pour le traitement d'images. La différence avec les réseau de neurones artificiels basiques est l'ajout de couches convolutives qui transforment leurs propres entrées, d'une manière ou d'une autre, avant de les passer aux couches prochaines ; cela dans le but de simplifier l'image avant le traitement..
☞ Voir réseau de neurones artificiels
✍ Pages 7–10, 25.

Taux de Faux Négatif à Rappel Maximal Décrit en détail dans cette sous-section 3.5.3
✍ Pages 14, 16, 17.

Techniques immunohistochimiques L'immunohistochimie est une méthode de localisation de protéines dans les cellules d'une coupe de tissu, par la détection d'antigènes au moyen d'anticorps.
✍ Page I.

Traitement de l'image Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou

d'en extraire de l'information.Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées
✍ Pages 23, 24.

Validation Croisée à k-blocs En anglais, k-fold cross-validation : on divise l'échantillon original enk échantillons (ou blocs), puis on sélectionne un des k échantillons comme ensemble de validation pendant que les k-1 autres échantillons constituent l'ensemble d'apprentissage. Après apprentissage, on peut calculer une performance de validation. Puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les blocs prédéfinis. À l'issue de la procédure nous obtenons ainsi k scores de performances, un par bloc. La moyenne et l'écart type des k scores de performances peuvent être calculés pour estimer le biais et la variance de la performance de validation.
✍ Pages 13, 20.

Sigles et acronymes

ACF *Ensemble de cryptes aberrantes.*
☞ Voir crypte déficiente
✍ Pages V, 15, 16, 21, 22.

AP *Average Precision.*
☞ Voir Average Precision
✍ Pages 13, 15–17.

CCR *Cancer colorectal.*
☞ Voir cancer colorectal
✍ Pages I, 2, 24.

CNIL *Commission nationale de l'informatique et des libertés.*
☞ Voir Commission nationale de l'informatique et des libertés
✍ Page 23.

GPU *Processeur graphique.*
☞ Voir processeur graphique
✍ Page 24.

IA *Artificial Intelligence.*
☞ Voir intelligence artificielle
✍ Page 23.

IP *Traitement de l'image.*
☞ Voir *image processing*
✍ Page 24.

JPEG Correspond à une norme qui définit le format d'enregistrement et l'algorithme de décodage pour une représentation numérique compressée d'une image fixe
✍ Page 4.

JSON Correspond à un format de fichier qui permet de représenter de l'information structurée et de la stocker
✍ Page 4.

ML *Machine learning.*
☞ Voir *machine learning*
✍ Pages I, 23.

MMR *Instabilité des microsatellites.*
☞ Voir Instabilité des micro satellites
✍ Page 2.

OCC *One Class Classification.*
☞ Voir One Class Classification
✍ Pages 7, 8.

PRM *Précision à Rappel Maximal.*
☞ Voir Précision à Rappel Maximal
✍ Pages 13–17.

PWR *Pairwise ranking.*
☞ Voir *pairwise ranking*
✍ Pages 16, 17.

RGPD *Règlement général sur la protection des données.*
☞ Voir Règlement général sur la protection des données
✍ Page 23.

TFPRM *Taux de Faux Négatif à Rappel Maximal.*
☞ Voir Taux de Faux Négatif à Rappel Maximal
✍ Pages 13–17.

TIFF Un format de fichier pour image numérique. Adobe en est le dépositaire et le propriétaire initial. Plus exactement, il s'agit d'un format de conteneur (ou encapsulation), à la manière de avi ou zip, c'est-à-dire pouvant contenir des données de formats arbitraires.
✍ Page 4.

Bibliographie

[1] S. Alrawi, M. Schiff, R. Carroll, M. Dayton, J. Gibbs, M. Kulavlat, D. Tan, K. Berman, D. Stoler et G. Anderson, « Aberrant crypt foci, » English (US), *Anticancer Research*, t. 26, n° 1 A, p. 107-119, jan. 2006, issn : 0250-7005.

[2] A. Anka, « YOLOv4 : Optimal Speed & Accuracy for object detection, » *Towards Science*, mai 2020. adresse : https://towardsdatascience.com/yolo-v4-optimal-speed-accuracy-for-object-detection-79896ed47b50.

[3] C. Bouchez, E. Kempf et C. Tournigand, « Traitement des autres tumeurs solides métastatiques MSI/dMMR, » *Bulletin du Cancer*, t. 106, n° 2, p. 143-150, 2019, issn : 0007-4551. doi : https://doi.org/10.1016/j.bulcan.2019.01.008. adresse : https://www.sciencedirect.com/science/article/pii/S0007455119300633.

[4] J. Brownlee, *How to Configure Image Data Augmentation in Keras*, https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training\-deep-learning-neural-networks/, avr. 2019.

[5] J. M. Cebrian, G. D. Guerrero et J. García, « Energy Efficiency Analysis of GPUs, » *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, p. 1014-1022, 2012.

[6] W. Chen, T.-Y. Liu, Y. Lan, Z. Ma et H. Li, « Ranking Measures and Loss Functions in Learning to Rank., » p. 315-323, jan. 2009.

[7] F. Chollet et al., *Keras*, keras.io, 2015.

[8] A. Delaitre et A. Calliger, *Peu d'images labellisées ? Optez pour la Data Augmentation !* https://www.quantmetry.com/blog/data-augmentation-image/, mai 2020.

[9] Y. B. Denis Drodzdov Maxim Kolomeychenko, *Supervisely*, https://supervise.ly/, 2017.

[10] A. Echle et al., « Clinical-Grade Detection of Microsatellite Instability in Colorectal Tumors by Deep Learning, » *Gastroenterology aga*, t. 159, n° 4, p. 1406-1416, 2020. doi : https://doi.org/10.1053/j.gastro.2020.06.021.

[11] B. O. Fenech M Strukelj N, *Ethical, social and polictical challenges of artificial intelligence in health*, http://futureadvocacy.com/wp-content/uploads/2018/04/1804_26_FA_ETHICS_08-DIGITAL.pdf, avr. 2018.

[12] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger et H. Greenspan, « GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, » *Neurocomputing*, t. 321, p. 321-331, 2018, issn : 0925-2312. doi : https://doi.org/10.1016/j.neucom.2018.09.013. adresse : https://www.sciencedirect.com/science/article/pii/S0925231218310749.

[13] J. Fürnkranz et E. Hüllermeier, « Pairwise preference learning and ranking, » *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, t. 2837, p. 145-156, jan. 2003.

[14] C. G, G. A, L. L, B. A. Luceri C et D. P., « Characterisation of aberrant crypt foci in carcinogen-treated rats : association with intestinal carcinogenesis, » *Br J Cancer*, t. 6206, p. 763-769, avr. 1995. doi : 10.1038/bjc.1995.148.

[15] L. Gao, L. Zhang, C. Liu et S. Wu, « Handling Imbalanced Medical Image Data : A Deep-Learning-Based One-Class Classification Approach, » *Artificial Intelligence in Medicine*, t. 108, p. 101 935, août 2020. doi : 10.1016/j.artmed.2020.101935.

[16] N. Hariri, F. Babalhavaeji, M. Farzandipour et S. Nadi-Ravandi, « Evaluation Criteria of Information Retrieval Systems : What We Know and What We Do Not Know, » *Iranian Journal of Information Processing Management*, t. 30, p. 199-221+xiii, sept. 2014.

[17] *IMageNet*, http://www.image-net.org/, 2009.

[18] Intel, *OpenCV*, https://opencv.org/, 2010.

[19] Järvelin, Kalervo et J. Kekäläinen, « IR evaluation methods for retrieving highly relevant documents, » *ACM Transaction of Information Systems*, t. 20, p. 41-48, jan. 2000. doi : 10.1145/345508.345545.

[20] M. Jogin, M. Mohana, M. Madhulika, G. Divya, R. Meghana et S. Apoorva, « Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning, » mai 2018, p. 2319-2323. doi : 10.1109/RTEICT42901.2018.9012507.

[21] S. D. Joseph Redmon et al., *Yolov3*, https://pjreddie.com/darknet/yolo/, 2015.

[22] S. Khan et M. Madden, « A Survey of Recent Trends in One Class Classification, » *Artificial Intelligence and Cognitive Science, Lecture Notes in Computer Science, vol. 6206*, t. 6206, p. 188-197, août 2009. doi : 10.1007/978-3-642-17080-5_21.

[23] J. C. Kirk Martinez, *VIPS*, https://github.com/libvips/libvips, 1997.

[24] G. Koch, R. Zemel et R. Salakhutdinov, « Siamese Neural Networks for One-shot Image Recognition, » 2015.

[25] C. Lanfranchi, *Automated detection in artificial intelligence of deficient MMR crypts to aid in the diagnosis of Lynch syndrome*. août 2020.

[26] F. T. Liu, K. M. Ting et Z. Zhou, « Isolation Forest, » p. 413-422, 2008. doi : 10.1109/ICDM.2008.17.

[27] B. A. Magnuson, I. Carr et R. P. Bird, « Ability of Aberrant Crypt Foci Characteristics to Predict Colonic Tumor Incidence in Rats Fed Cholic Acid, » *Cancer Research*, t. 53, n° 19, p. 4499-4504, 1993, issn : 0008-5472. eprint : https://cancerres.aacrjournals.org/content/53/19/4499.full.pdf. adresse : https://cancerres.aacrjournals.org/content/53/19/4499.

[28] F. Meng, Y. Fu, F. Lou et Z. Chen, « An Effective Network Attack Detection Method Based on Kernel PCA and LSTM-RNN, » in *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, 2017, p. 568-572. doi : 10.1109/ICCSEC.2017.8447022.

[29] B. P., « Health disparities and health equity : concepts and measurement, » *Annu Rev Public Health*, t. 27, n° 4, p. 94-167, 2006. doi : https://doi.org/10.1146/annurev.publhealth.27.021405.102103.

[30] P. Perera, P. Oza et V. Patel, « One-Class Classification : A Survey, » *ArXiv*, t. abs/2101.03064, 2021.

[31] B. S. Pinto Dos Santos D Giese D et al., « Medical students' attitude towards artificial intelligence : a multicentre survey, » *Eur Radiol*, t. 29, n° 4, p. 1640-1646, 2019. doi : https://doi.org/10.1007/s00330-018-5601-1.

[32] W. N. Price, « Medical Malpractice and Black-Box Medicine, » in *Big Data, Health Law, and Bioethics*, I. G. Cohen, H. F. Lynch, E. Vayena et U. Gasser, éd. Cambridge University Press, 2018, p. 295-306. doi : 10.1017/9781108147972.027.

[33] L. Roncucci, D. Stamp, A. Medline, J. B. Cullen et W. Robert Bruce, « Identification and quantification of aberrant crypt foci and microadenomas in the human colon, » *Human Pathology*, t. 22, n° 3, p. 287-294, 1991, issn : 0046-8177. doi : https://doi.org/10.1016/0046-8177(91)90163-J. adresse : https://www.sciencedirect.com/science/article/pii/004681779190163J.

[34] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller et M. Kloft, « Deep One-Class Classification, » Proceedings of Machine Learning Research, t. 80, J. Dy et A. Krause, éd., p. 4393-4402, juil. 2018. adresse : http://proceedings.mlr.press/v80/ruff18a.html.

[35] N. J. Samadder, K. R. Smith, J. Wong, A. Thomas, H. Hanson, K. Boucher, C. Kopituch, L. A. Cannon-Albright, R. W. Burt et K. Curtin, « Cancer Risk in Families Fulfilling the Amsterdam Criteria for Lynch Syndrome, » *JAMA Oncology*, t. 3, n° 12, p. 1697-1701, déc. 2017, issn : 2374-2437. doi : 10.1001/jamaoncol.2017.0769. eprint : https://jamanetwork.com/journals/jamaoncology/articlepdf/2646792/jamaoncology_samadder_2017_br_170008.pdf. adresse : https://doi.org/10.1001/jamaoncol.2017.0769.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov et L.-C. Chen, « MobileNetV2 : Inverted Residuals and Linear Bottlenecks, » 2019. arXiv : 1801.04381 [cs.cv].

[37] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor et J. Platt, « Support Vector Method for Novelty Detection, » *NIPS*, t. 12, p. 582-588, jan. 1999.

[38] N. Sharma, V. Jain et A. Mishra, « An Analysis Of Convolutional Neural Networks For Image Classification, » *Procedia Computer Science*, t. 132, p. 377-384, 2018, International Conference on Computational Intelligence and Data Science, issn : 1877-0509. doi : https://doi.org/10.1016/j.procs.2018.05.198. adresse : https://www.sciencedirect.com/science/article/pii/S1877050918309335.

[39] P. M. Stewart, « The Actual Difference Between Statistics and Machine Learning, » *Towards Data Science*, mar. 2015.

[40] S. Syngal, R. Brand, J. Church, F. Giardiello, H. Hampel et R. Burt, « ACG Clinical Guideline : Genetic Testing and Management of Hereditary Gastrointestinal Cancer Syndromes, » *The American journal of gastroenterology*, t. 110, p. 223-62, fév. 2015. doi : 10.1038/ajg.2014.435.

[41] C. Szegedy, S. Ioffe, V. Vanhoucke et A. Alemi, *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, 2016. arXiv : 1602.07261 [cs.cv].

[42] T. Takayama, S. Katsuki, Y. Takahashi et al., « Aberrant Crypt Foci of the Colon as Precursors of Adenoma and Cancer, » *N Engl J Med*, t. 339, p. 1277-1284, oct. 1998. doi : 10.1056/NEJM199810293391803.

[43] D. Taubman, *Kakadu*, https://kakadusoftware.com/, 2002.

[44] A. Umar, C. R. Boland, J. P. Terdiman, S. Syngal, A. d. l. Chapelle, J. Rüschoff, R. Fishel, N. M. Lindor, L. J. Burgart, R. Hamelin, S. R. Hamilton, R. A. Hiatt, J. Jass, A. Lindblom, H. T. Lynch, P. Peltomaki, S. D. Ramsey, M. A. Rodriguez-Bigas, H. F. A. Vasen, E. T. Hawk, J. C. Barrett, A. N. Freedman et S. Srivastava, « Revised Bethesda Guidelines for Hereditary Nonpolyposis Colorectal Cancer (Lynch Syndrome) and Microsatellite Instability, » *JNCI : Journal of the National Cancer Institute*, t. 96, n° 4, p. 261-268, fév. 2004, issn : 0027-8874. doi : 10.1093/jnci/djh034. eprint : https://academic.oup.com/jnci/article-pdf/96/4/261/10892566/zv800404000261.pdf. adresse : https://doi.org/10.1093/jnci/djh034.

[45] C. I. Vayena E Blasimme A, « Machine learning in medicine : Addressing ethical challenges, » *PLoS Med 15*, t. 11, n° e1002689, 2018. doi : https://doi.org/10.1371/journal.pmed.1002689.