



Rapport de Projet Apprentissage, Intelligence Artificielle et Optimisation (AIAO) :

RSNA Screening Mammography Breast Cancer Detection

M2 Biologie Informatique

Année 2023/2024

BETHUEL Elouan - GHRAIZI Marwa - SALAUN Nicolas
N°22218464 - N°22212451 - N° 22200082

Introduction

Le cancer du sein est le type de cancer le plus observé chez les femmes, avec quelque 61000 nouveaux cas en 2023 touchant majoritairement des femmes âgées de plus de 50 ans. Dans un contexte global de vieillissement de la population, la prévalence de ce cancer est naturellement en hausse. De nombreuses campagnes de dépistages ont été réalisées afin de déceler ce cancer à son stade le plus précoce, permettant de sauver de nombreuses patientes. En 2018, c'est 5 millions de patientes dépistées en France, et donc autant de mammographies, qui ont été réalisées.

C'est dans ce contexte de gestion de grandes quantités de données médicales, que nous avons choisi d'entreprendre ce projet de Deep Learning. Cette approche suscite un intérêt particulier, car elle offre la possibilité d'exploiter les nombreuses images de mammographies pour développer un modèle diagnostique. En effet, le Deep Learning se prête idéalement à l'analyse d'images complexes telles que les mammographies, en détectant des patterns subtils et des caractéristiques qui échappent parfois à l'œil humain.

Notre objectif est donc d'élaborer un modèle de Deep Learning capable de fournir un pré diagnostic, aidant ainsi à orienter les médecins dans leur évaluation. Il est essentiel de souligner que ce modèle n'a en aucun cas pour but de remplacer l'expertise médicale. Il doit être considéré comme un outil supplémentaire au service des professionnels de santé, visant à renforcer leur capacité à prendre des décisions éclairées. Le diagnostic final reposera toujours sur une évaluation médicale approfondie, prenant en compte l'ensemble des informations du patient.

Ce projet est réalisé dans le cadre d'une compétition organisée par la Société Radiologique de l'Amérique du Nord (RSNA) sur la plateforme Kaggle en Novembre 2022 ([lien vers le site de la compétition](#)). Les scripts et les notebooks Google Colab sont disponibles sur [ce répertoire github](#).

Matériels et Méthodes

Matériel

Le jeu de données disponible sur le site de la compétition contenait les images de mammographies en format .dcm ainsi qu'un fichier .csv contenant les métadonnées des images avec de l'information sur la classe des images, l'âge des patientes, la latéralité des images etc. L'archive du dossier des images ayant une taille de 314.72 Go se révèle être excessivement volumineux et donc particulièrement difficile à manipuler en pratique. Nous avons alors utilisé les images croppées fournies généreusement par l'utilisateur Kaggle [Fabien Daniel](#). Le jeu de données consistait au final des images à plusieurs résolutions (256, 512, 768 et 1024) et au format .png avec 53548 images de mammographies saines et 1158 cancéreuses.

Pre-processing

Cropping des images

Le prétraitement des images est essentiel dans notre projet car nous disposons initialement de 54 706 fichiers au format .dcm. Toutefois, bon nombre de ces images comportent des zones noires qui ne contenaient aucune information pertinente pour notre objectif de détection du cancer du sein.

Dans un premier temps, nous avons converti les images au format .dcm en un format plus accessible et moins lourd, à savoir le format .png. Nous avons par la suite décidé d'entraîner un modèle YOLOX à reconnaître nos zones d'intérêt dans l'image originale, soit le sein. Afin d'entraîner ce modèle, il a tout d'abord fallu annoter nos zones d'intérêt, et ce sur un échantillon d'apprentissage de 320 mammographies. Nous avons tout d'abord envisagé d'utiliser le logiciel Fiji pour cette tâche, cependant, cela s'est avéré être un processus fastidieux, et nous avons donc décidé d'utiliser le logiciel Roboflow, ce qui nous a permis de gagner en efficacité lors de l'annotation. La qualité des annotations ayant un impact direct sur notre modèle final de détection de la zone d'intérêt, ces annotations ont été réalisées à la main avec précision pour tout notre échantillon.

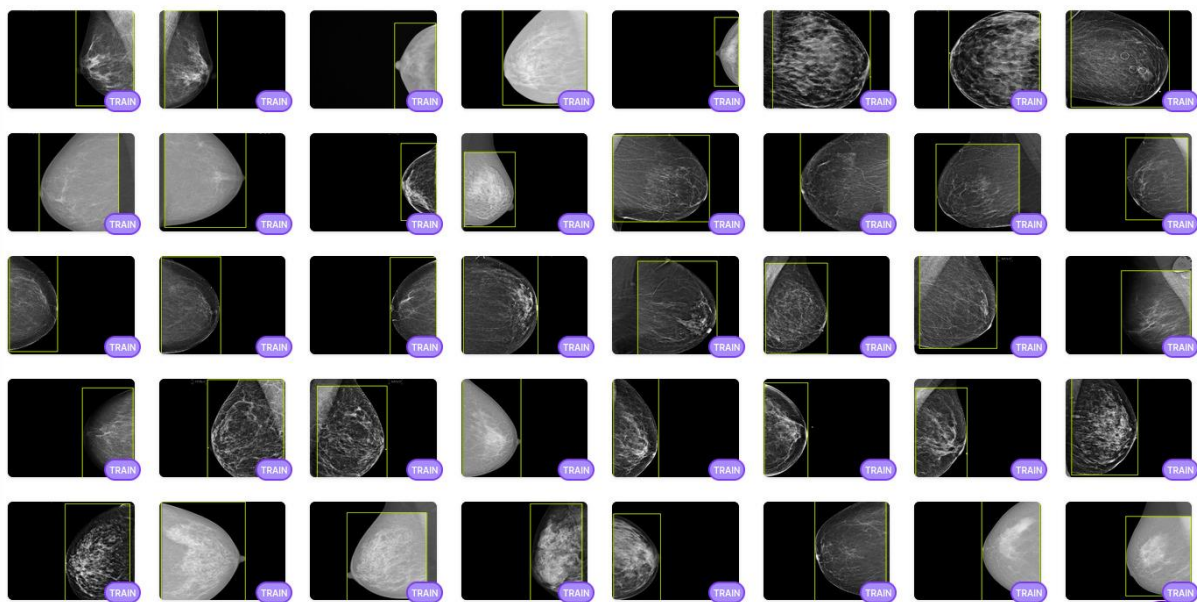


Fig. 1 Images d'annotation des ROIs des mammographies pour le modèle YOLOX.

Suite à cette phase d'annotation, nous avons divisé notre jeu de données en deux sous-ensembles distincts : un ensemble d'entraînement comprenant 250 images et un ensemble de test avec 70 images. Enfin, nous avons utilisé le modèle YOLOX en appliquant les paramètres prédéfinis disponibles sur le logiciel. Grâce à ce modèle, nous avons atteint une précision de 99,7%, ce qui était attendu au vu du type de détection demandé. Une fois ce modèle entraîné, nous pouvons pour chaque image passée, obtenir les ROI (Region Of Interest) détectées par le modèle, sous la forme de quatre points de coordonnées x et y sur notre image, nous permettant de facilement garder uniquement la ROI de chaque image (exemple de résultat du modèle en annexe).

T-distributed Stochastic Neighbor Embedding (T-SNE)

Dans un but purement exploratoire, nous avons commencé par utiliser une méthode non-supervisée de visualisation, le T-SNE à l'aide du module manifold de la librairie sklearn, sur notre jeu de données [1]. Cette approche est fréquemment utilisée pour des jeux de données de grandes dimensions pour identifier des structures locales dans les données. Pour cela, 400 images cancéreuses et 400 images saines ont été utilisées avec différentes valeurs de perplexité, des valeurs faibles afin d'essayer d'identifier les structures locales (tumeurs, calcifications) et des valeurs plus élevées dans l'espoir de séparer les images selon les deux classes, saines ou cancéreuses. La perplexité est un paramètre fondamental car il influe sur la balance entre la préservation des structures locales et globale des données.

Traitement des images, normalisation et contraste

L'objectif de la normalisation est de modifier la plage de valeur des niveaux de gris de chaque image, ainsi que la distribution des niveaux de gris des pixels afin d'avoir des images moins variables entre elles et ainsi faciliter l'apprentissage du réseau. Deux stratégies de normalisation ont été testées, par z-score et par quantiles, avec différentes valeurs de percentiles. Sachant que notre objectif était de diminuer la variabilité de répartition des pixels entre les images, le calcul de la moyenne, des écarts-types et des quantiles a été effectué sur l'ensemble du jeu d'apprentissage.

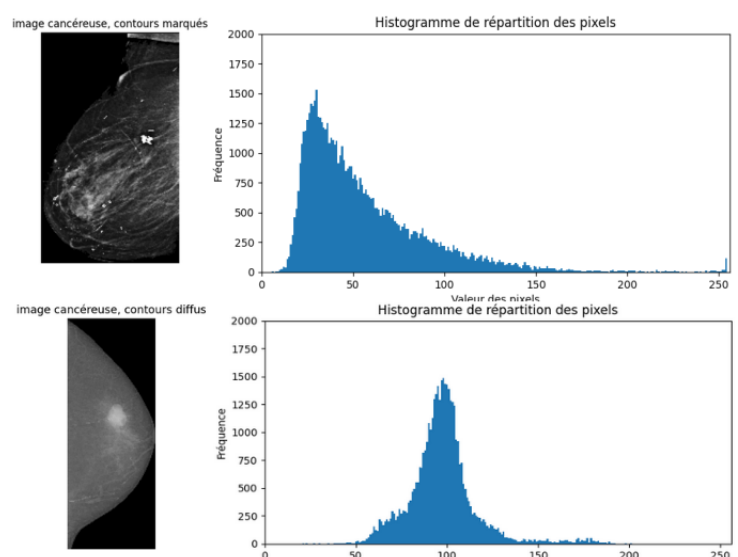


Fig. 2 Comparatif de la répartition des pixels entre deux images à luminosité différente.

1) Normalisation z-score :

$$\text{Formule : } \text{pixel normalisé} = \text{pixel} - \frac{\text{moyenne globale}}{\text{ecart type global}}$$

2) Normalisation par quantiles :

$$\text{Formule : } \text{pixel normalisé} = \frac{\text{pixel} - \text{quantile inferieur global}}{\text{quantile superieur global} - \text{quantile inferieur global}}$$

Deux stratégies d'accentuation du contraste par égalisation des histogrammes ont été testées à l'aide du module cv2 de python. À l'inverse de la normalisation, la modification a été apportée à chaque image individuellement.

1) Égalisation d'histogramme [2]:

L'égalisation d'histogramme est fréquemment employée en traitement d'image dans le but d'améliorer la qualité visuelle. Cette transformation d'image va rendre l'histogramme de cette dernière le plus proche possible d'une distribution uniforme, ce qui aura pour effet de tasser les niveaux de gris avec peu de pixels et d'étaler ceux avec beaucoup de pixels, ce qui se traduit visuellement par une image plus contrastée.

2) CLAHE (Égalisation d'histogramme adaptatif limité par contraste) [2] :

La méthode CLAHE diffère légèrement de l'égalisation d'histogramme classique par la séparation de l'image et le calcul d'un nouvel histogramme par chaque sous-partie. Elle permet ainsi d'améliorer le contraste de manière plus locale.

Dans la suite de ce rapport, nous ferons référence aux images égalisées avec CLAHE comme les images enhanced, et celles traitées par égalisation d'histogramme classique comme les images égalisées.

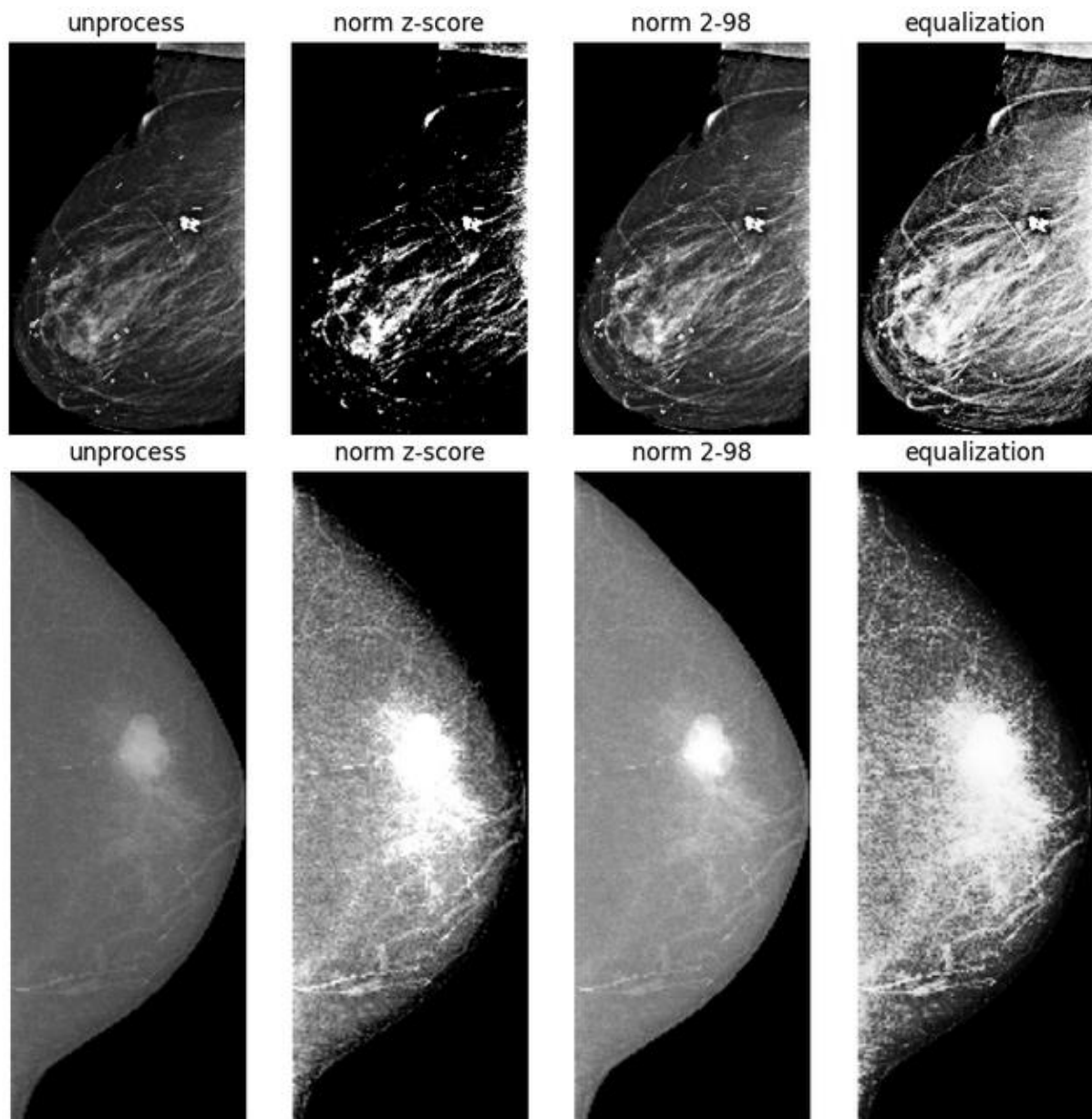


Fig 3. Illustrations de l'effet des traitements sur deux images aux luminosités distinctes (les histogrammes associés sont disponible en annexe 2)

Data Augmentation

Vu le déséquilibre des classes d'observations dont seulement 1158 images cancéreuses, nous avons effectué une augmentation des images avec la librairie python Augmentor dans le but de générer un échantillon d'images cancéreuses plus important.

Le choix des modifications appliquées aux images a été fait suite à l'étude de l'état de l'art de l'augmentation appliquée par les participants à la compétition Kaggle. Les modifications choisies se résument par des rotations horizontales et verticales avec une probabilité de 0.5, des changements aléatoires du contraste et de la luminosité avec une probabilité de 0.5 et une magnitude de modification entre 0.8 et 1.2, des distorsions d'image de probabilité de 0.5 et finalement des déviations avec une probabilité de 0.4 et une magnitude de 0.6. Ces modifications permettent aux modèles de s'exposer aux tumeurs qui peuvent mieux ressortir dans des angles et des luminosités variées. L'augmentation des images nous a permis de passer de 965 images d'entraînement par classe à 2895 images. Un exemple de data augmentation se trouve en annexe 4.

Organisation des données d'entrées

Il est important de fournir aux fonctions de chargement de données de keras la bonne structure de dossiers qui séparent bien les données d'entraînement et de validation ainsi que les données cancéreuses et saines pour l'attribution des bonnes étiquettes aux images. Les images brutes, initialement mélangées, ont été séparées en premier lieu en images cancéreuses et saines suivant les informations dans le fichier des métadonnées. Ensuite, 80% des images ont été échantillonnées dans le jeu de données "train" et 20% dans le jeu de données "test". La structure finale des dossiers est décrite dans la figure ci-contre.

```
--data
----train
-----cancerous
-----healthy
----test
-----cancerous
-----healthy
```

La mise en place de cette structure a été réalisée par un pipeline de scripts python pour chaque résolution d'images à partir des images brutes. Dans le cas de prétraitement, seules les images du jeu de données "train" ont subi des traitements de normalisation, amélioration de qualité (enhancement) et data augmentation. Il faut noter que seulement 1158 images saines ont été conservées pour maintenir un équilibre avec les images cancéreuses. Les mêmes images saines ont été sélectionnées pour les différentes résolutions, préservant ainsi la cohérence des données.

Modèles

Modèle CNN basique

Le premier type de modèle vers lequel nous nous sommes penchés sont les CNN (Convolutional Neural Networks). Ces derniers sont des réseaux de neurones conçus pour le traitement d'images et de données en grille. Ils utilisent des opérations de convolution pour extraire des caractéristiques clefs, suivies de couches de pooling pour réduire les dimensions ainsi que des fonctions d'activation, telles que ReLU, introduisant de la non-linéarité. Les CNN utilisent souvent des couches de normalisation pour stabiliser l'entraînement, et ils sont largement utilisés dans des tâches de classification d'images. Pour ce premier test de modèle, nous avons choisi une approche exploratoire, dans le but d'observer l'impact des différentes structures sur les résultats. Une fonction a pour cela été créée afin de tester différentes architectures. Ainsi, pour 32, 64 et 128 neurones par couche, quatre architectures ont été choisies, contenant des couches de Conv2D (conv), de MaxPooling2D (maxpool) ainsi que des couches Dense :

Type I : conv / maxpool / conv / maxpool / dense

Type II : conv / maxpool / conv / maxpool / dense / dense

Type III : conv / maxpool / conv / maxpool / conv / maxpool / dense / dense,

Type IV : conv / maxpool / conv / maxpool / conv / maxpool / conv / maxpool / dense / dense

Augmenter de manière croissante la complexité des architectures CNN, caractérisées par l'ajout de couches de convolution et de pooling, permet de capturer des informations de plus en plus abstraites et complexes dans les données, et augmentent ainsi la capacité de représentation du modèle.

Les couches de pooling réduisent la dimension des caractéristiques, diminuant la charge de calcul et aidant à éviter le sur-apprentissage. Enfin, les couches finales entièrement connectées permettent d'adapter le modèle à des tâches spécifiques. Il est important de garder un équilibre entre les couches et d'éviter de les multiplier, car plus de couches n'impliquent pas nécessairement une meilleure performance.

EfficientNet

Vu la complexité de nos données, nous avons principalement adopté l'approche transfer learning qui consiste à utiliser des modèles pré-entraînés sur de vastes ensembles de données comme les modèles EfficientNet. Ces derniers sont une famille de réseaux de neurones convolutifs efficace dans les tâches de classification d'images. Ils se reposent sur trois facteurs : la largeur des réseaux (nombre de canaux des couches), la profondeur des réseaux (nombre de couches) et la résolution des données d'entrées. Il existe 8 variantes de la 1ère version d'EfficientNet de B0 à B7 [3] avec des paramètres de profondeur, largeur et résolution de plus en plus complexes.

Notre implémentation de transfer learning consiste à entraîner le modèle EfficientNet en figeant les couches du modèle pour les 25 premières epochs la structure consiste en :

- 1) Initialisation du modèle EfficientNet avec les poids pré-entraînés sur ImageNet et `model.trainable = false` pour ne pas mettre à jour les poids durant l'entraînement.
- 2) Couche Global Average Pooling pour réduire les caractéristiques spatiales en une seule valeur par canal
- 3) Couche normalization par batch pour améliorer la stabilité de l'entraînement
- 4) Couche drop out avec un taux de 0.2 pour régulariser le surapprentissage
- 5) Couche finale à un neurone avec une fonction d'activation sigmoïde pour effectuer la classification binaire
- 6) Modèle compilé avec l'optimiseur Adam et un taux d'apprentissage de $10e-3$

Une fois le modèle gelé entraîné, les 20 dernières couches du modèle sont dégelées, avec un taux d'apprentissage plus faible ($10e-4$) pour leur permettre d'apprendre nos données spécifiques sur 15 epochs.

Le choix des variantes d'Efficient Net dépend de la congruence entre la résolution de nos images et la résolution ImageNet sur laquelle le modèle a été pré-entraîné. Ainsi, les données de résolution 256 ont été entraînées avec EfficientNet B2, et les images de résolution 512 avec EfficientNet B6 et EfficientNet V2S [4], comme recommandé par les participants de la compétition Kaggle. Les images de résolution 512 ont aussi été réduites en 380 pour être entraînées avec EfficientNet B4. Le taille de batch optimal a été déterminé comme 32 pour prendre en considération les grandes résolutions des images.

DenseNet

Au vu des résultats peu encourageants avec les CNN classiques, un transfer learning à l'aide d'un réseau de neurones à l'architecture différente, ici un DCNN, a été essayé. Un DCNN

est un réseau pour lequel chaque couche est connectée à l'ensemble des autres couches de manière à ce qu'une couche ait en entrée les sorties des couches précédentes. Cette architecture 'dense' se démarque par plusieurs avantages, en particulier de meilleures performances sur des tâches jugées complexes et une diminution du sur-apprentissage. Densenet est communément utilisé pour la classification d'images et en particulier pour des données médicales, comme par exemple la détection de kystes pancréatiques, d'après cette étude de 2019. [5]

Notre démarche fut similaire au transfert learning par EfficientNet à l'exception d'ajout de couches de dropout avec un taux plus élevé (0.4-0.6) pour palier à un fort sur-apprentissage. Plusieurs essais ont aussi été réalisés en jouant avec le poids accordé aux classes, afin de pénaliser les mauvaises prédictions de la classe cancéreuse. Un poids entre 4 et 5 fois plus élevé pour la classe cancéreuse a permis d'augmenter légèrement le taux de vrais positifs sans trop perturber les prédictions des vrais négatifs.

Evaluation

Pour évaluer les performances de nos modèles, nous avons utilisé plusieurs métriques notamment le F1 score employé par la compétition, ainsi que le taux de bien prédits (accuracy), la précision et le recall. Nous avons également généré des matrices de confusion pour visualiser les prédictions ainsi que des courbes Receiver Operating Characteristics (ROC) à partir desquelles on a déduit l'index de Youden qui nous a permis de choisir le seuil de classification optimal à partir des probabilités prédites par un modèle. Les formules des métriques d'évaluation se trouvent dans l'annexe 5.

Résultats & Discussion

Exploration des données

D'après l'histogramme ci-contre, on constate un fort déséquilibre entre les deux classes. Ceci est un problème couramment rencontré en deep learning qui empêche le bon apprentissage des réseaux de neurones qui vont avoir tendance à toujours prédire la classe majoritaire pour augmenter l'accuracy. Pour y remédier, nous avons utilisé trois stratégies. Le sous-échantillonnage, en réduisant le nombre d'images de la classe majoritaire 'healthy', le sur-échantillonnage, en générant de nouvelles images de la classe minoritaires 'cancerous' par data augmentation. Enfin, nous avons également joué sur les poids des classes lors de l'apprentissage de nos différents modèles dans le but de pénaliser davantage une mauvaise prédiction de la classe minoritaire.

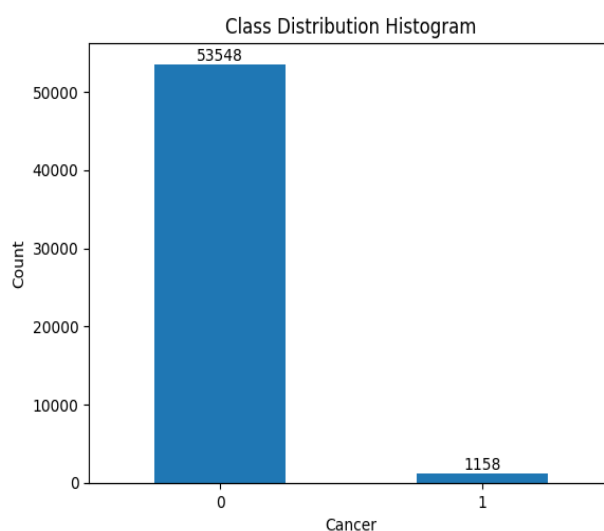


Fig 4. Distribution des classes dans le jeu de données

T-SNE

Le résultat produit sur les deux premières dimensions, montre que cette approche n'a pas été concluante. En effet, pour des faibles valeurs de complexités, nous nous attendions à obtenir un certain nombre de clusters qui auraient pu être associés à des structures locales (tumeurs, calcifications), or les données se répartissent de manière uniforme dans l'espace. Il en va de même avec des valeurs plus élevées de complexité, alors que nous nous attendions à obtenir deux clusters distincts, correspondant à nos deux classes d'images. Ceci illustre la complexité d'identifier des structures permettant de classifier nos images et la nécessité d'utiliser des approches supervisées et plus complexes. Un comparatif de différentes valeurs de perplexité est disponible en annexe.

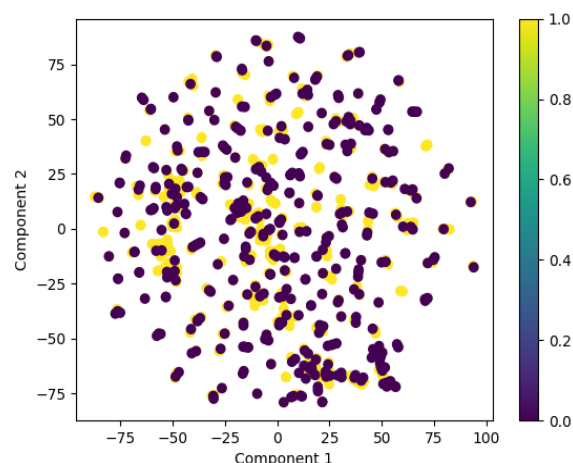


Fig 5. Projection des données sur les deux premières dimensions T-SNE

Modèle CNN

Les résultats des modèles CNN basiques sont renseignés ci-dessous. Tous les modèles ont été entraînés sur les images non traitées.

Model	Type I	Type II	Type III	Type IV
32 neurons / layers	0,5	0,47	0,51	0.5
64 neurons / layers	0,51	0,49	0,53	0.52
128 neurons / layers	0,53	0,53	0,55	0.53

Tableau. 1 F1 score des différents modèles CNN

Nous pouvons voir avec les modèles CNN basiques, que nous obtenons des résultats globalement moyens. L'utilisation d'images de plus haute résolution ne fait qu'augmenter le temps de calcul, sans apporter de bénéfice significatif au modèle. De plus, nous pouvons observer ici que le meilleur modèle de CNN semble être le Type III, ce qui prouve bien qu'augmenter la complexité n'améliore pas forcément les résultats.

De plus, l'utilisation de couche dense rend quasi impossible l'utilisation de plus de neurones dans chaque couche, sans avoir d'erreur de mémoire. De manière générale, les CNN basiques non entraînés ne semblent pas être la solution optimale à notre problème.

EfficientNet

Les résultats des modèles Efficient Net sont décrits dans le tableau ci-dessous:

Model	EfficientNet B2 Resolution 256		EfficientNet B4 Resolution 380		EfficientNet B6 Resolution 512			EfficientNet V2S Resolution 512	
Data	un-processed	enhanced	un-processed	enhanced	un-processed	enhanced	augmented	un-processed	enhanced
F1 score	0.56	0.48	0.56	0.34	0.58	0.57	0.57	0.63	0.52
Accuracy	0.58	0.56	0.60	0.54	0.57	0.57	0.56	0.59	0.57
Precision	0.56	0.56	0.60	0.54	0.57	0.57	0.56	0.59	0.56
Recall	0.54	0.41	0.51	0.23	0.59	0.58	0.59	0.70	0.48

Tableau. 2 Résultats des modèles Efficient Net B2, B4, B6 et V2S sur des images de résolution 256, 380 et 512 prétraitées ou pas.

De manière générale, les modèles EfficientNet réussissent relativement bien à distinguer les images cancéreuses des images saines. Cependant, les images de patientes saines seront globalement toujours mieux détectées que les images de patientes cancéreuses. Cela est illustré par un taux de recall plus faible que le taux de précision.

Nous pouvons observer que la performance des modèles s'améliore à mesure que la résolution des images augmente, ce qui est attendu particulièrement dans le cas des tumeurs qui occupent un nombre limité de pixels. Cela est aussi relié à la complexité des images sur lesquelles les variantes de EfficientNet, tel que le B6 par rapport au B2, ont été entraînées.

Si l'on prend par exemple la version V2S de EfficientNet, l'utilisation de cette variante apporte une amélioration significative de performance, au prix d'un fort biais vers les images cancéreuses manifesté par le taux de recall élevé.

On observe également que l'augmentation de données n'apporte pas de gain significatif en termes de performance. Cela n'est pas surprenant car la data augmentation n'apporte pas de diversité, et ne fait que modifier des images déjà existantes. Cela illustre l'importance de fournir aux modèles des quantités suffisantes de données d'entraînement correctes. Nous pouvons pour cela utiliser des jeux de données externes.

La Fig. 6 du modèle EfficientNet B6 sur les images de résolution 512 montre que les valeurs de la fonction de perte pour le set d'entraînement diminuent au cours des epochs de l'entraînement dégelé, alors que celles du jeu de validation augmente. Cela indique un surapprentissage des données du jeu d'apprentissage par ce modèle. Cela est également illustré dans la Fig. 7 par la différence significative de la précision du set d'entraînement bien plus supérieure à la précision sur le set de validation.

Ce surapprentissage global pourrait être attribué à la quantité d'images d'entraînement insuffisante pour une bonne classification. De plus, la précision de la validation ne converge pas malgré un faible taux d'apprentissage. Cela peut être amélioré en entraînant le modèle sur un plus grand nombre d'epochs.

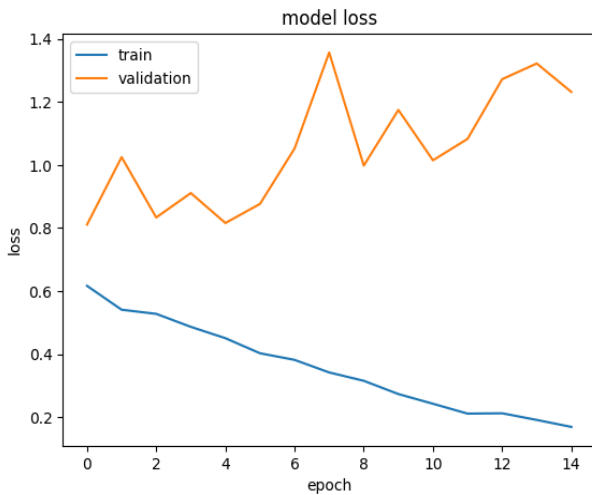


Fig 6. Evolution de la valeur de perte au cours des epochs du modèle EfficientNet B6 sur les images de résolution 512

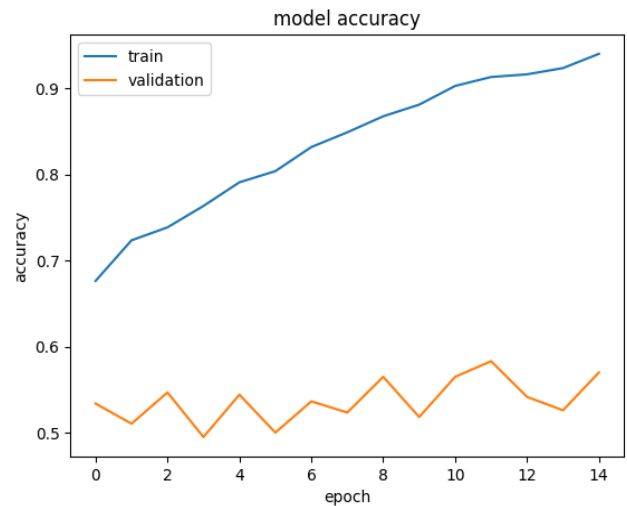


Fig 7. Evolution de l'accuracy au cours des epochs du modèle EfficientNet B6 sur les images de résolution 512

Le modèle EfficientNet B6 semble donner les meilleurs résultats en termes de bon prédit cancéreux et sains. Le modèle avec les données non traitées parvient ici à détecter deux images cancéreuses (label 1) de plus en comparaison avec le modèle basé sur les images enhanced.

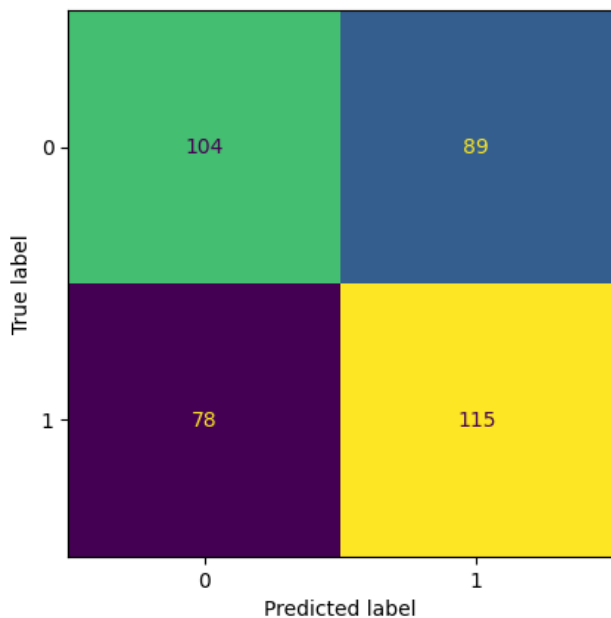


Fig 8. Matrice de confusion des résultats de EfficientNet B6 entraîné sur les images non-traitées à résolution 512

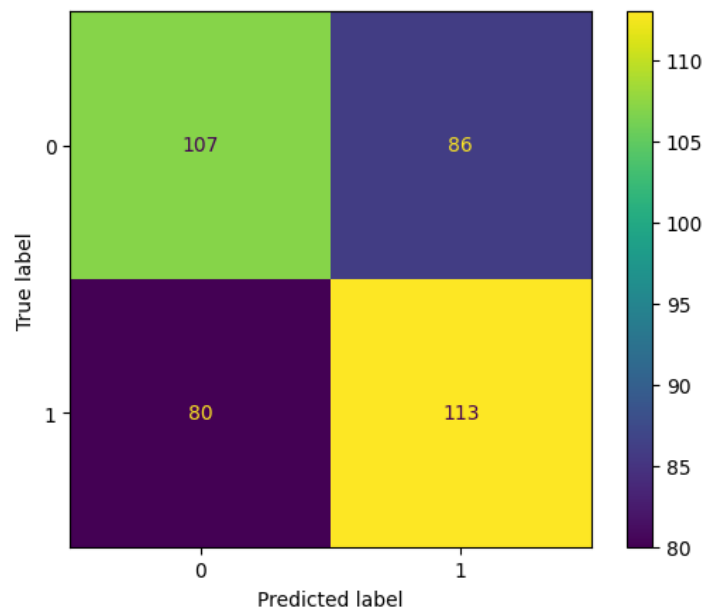


Fig 9. Matrice de confusion des résultats de EfficientNet B6 entraîné sur les images enhanced à résolution 512

Dans tous les cas sauf avec EfficientNet B6, les données enhanced favorisent souvent la classification en saine si le seuil de classification est maintenu à 0.5, cela se traduit par les faibles score de recall. Pour résoudre ce problème, nous avons étudié les courbes ROC issues des prédictions pour déterminer l'indice de Youden qui nous permet de trouver le meilleur équilibre entre la sensibilité et la spécificité. Dans le cas des images enhanced avec le modèle Efficient Net V2S (ROC curve Fig. 10), le meilleur seuil a été déterminé à 0.46 pour favoriser la classification en tant qu'image cancéreuse (label 1). Le nouveau seuil a considérablement

amélioré les performances pour atteindre un F1 score de 0.59, une accuracy de 0.61, une précision de 0.6 et un recall de 0.54.

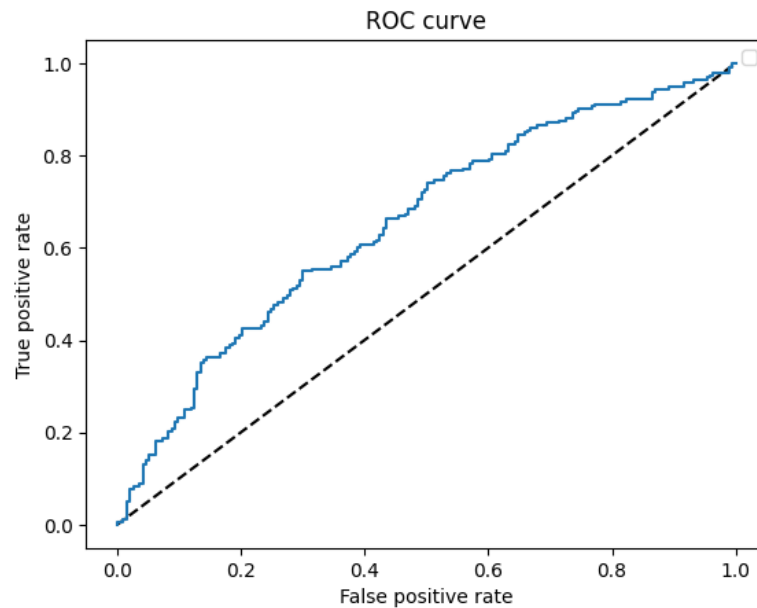


Fig. 10 courbe ROC des résultats de prédiction du modèle EfficientNet V2S sur les images non traitées de résolution 512

DenseNet

Avec les images non traitées Densenet semble avoir des performances plutôt bonnes en termes d'accuracy et de F1 score, mais en réalité ceci est dû à une bonne prédiction des images 'healthy' et non pas des 'cancerous' comme en témoignent le Recall et la matrice de confusion. Or, dans une démarche de prédiction médicale, ceci n'est pas souhaitable. De plus, malgré l'ajout de plusieurs couches de dropout élevé, le sur-apprentissage lors de l'entraînement restait problématique et l'accuracy sur le jeu de données test stagnait au-delà de 40 epochs car le modèle atteignait une accuracy de 0.97 sur le jeu d'entraînement.

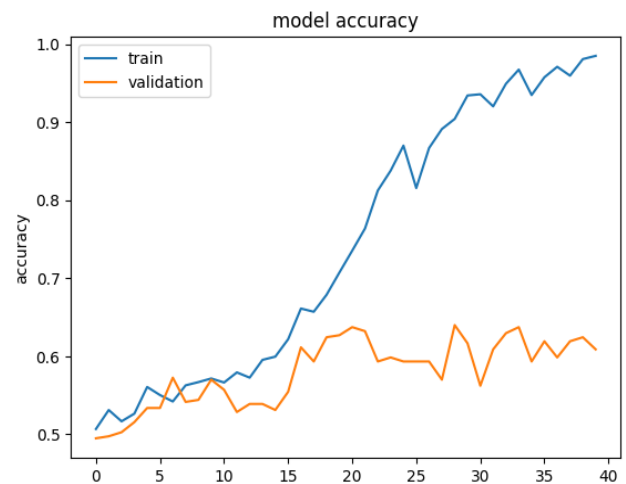


Fig 11. Evolution de l'accuracy au cours des epochs d'entraînement dégelé

Il est intéressant de noter qu'en entraînant le modèle avec des hyper-paramètres fixes, les meilleurs résultats ont été obtenus avec les données non traitées. Dans le cas de la normalisation, une des explications possibles serait que la proportion importante d'images trop claires dans notre jeu de données provoque un décalage de la répartition des pixels vers des valeurs plus élevées, donc plus claires, ce qui aurait pour effet de camoufler les patterns locaux. L'augmentation du contraste pourrait avoir un impact similaire.

DenseNet121				
Data	un - processed	norm z-score	norm 2-98	norm 5-95
F1 score	0.61	0.50	0.53	0.52
Accuracy	0.64	0.56	0.61	0.58
Precision	0.68	0.57	0.66	0.59
Recall	0.56	0.45	0.44	0.46

Tableau 3. Résultats du modèle DenseNet121 entraînés sur des images avec plusieurs normalisations

Conclusion

Au vu de la complexité de nos données, un travail important a été consacré au pré-traitement afin d'améliorer les performances d'apprentissages de nos modèles. Nous avons exploré plusieurs approches pour normaliser les images, augmenter leur contraste et pour contrer le déséquilibre de classe. Dans une démarche scientifique, nous avons ensuite testé l'impact de ces modifications sur nos résultats. Les différents modèles (CNN, EfficientNet et DenseNet) ont pour cela été entraînés sur les images ayant subi les différents pré-traitements.

Le modèle avec les résultats les plus fiables semble être Efficient Net B6 qui a atteint un F1 score de 0.58. Ce dernier a été entraîné sur les images de résolution 512 sans avoir un biais vers une classe en particulier, à l'instar d'autres modèles tels que Efficient Net V2S, possédant un meilleur F1 score, mais au prix d'un fort biais de prédiction.

De nombreuses améliorations pourraient être apportées à notre projet. Tout d'abord, il est nécessaire d'obtenir un nombre plus important d'images cancéreuses pour équilibrer les classes et apporter plus d'informations aux modèles. Cela peut être réalisé avec des jeux de données externes comme VinDr-Mammo [6] ou CBIS-DDSM [7]

Il est également important de dégeler un plus grand nombre de couches pour permettre aux modèles pré-entraînés d'étudier nos jeux de données sur des epochs plus nombreuses. Nous avons observé une amélioration de performance en augmentant la résolution des images d'entraînement ; il est donc nécessaire d'entraîner les images disponibles à résolution 768 et 1024.

D'autre part, il est recommandé d'utiliser des modèles pré entraînés sur des images médicales ou Greyscale Imagenet. La revue de Morid et al. (2021) recommande les modèles Inception V3 et AlexNet pour la classification des images médicales [8].

Enfin, l'utilisation de l'outil SHAP (SHapley Additive exPlanations) serait intéressante. En effet, il permettrait de comprendre quels sont les pixels qui contribuent le plus à la classification de nos images et ainsi identifier quelles structures permettent de réaliser des prédictions correctes et aussi d'identifier de potentiels artefacts. Pour mettre à contribution ces éventuelles structures locales, l'ajout de réseaux dédiés à la segmentation pourrait être utilisé pour déterminer ces régions d'intérêts puis tenter de les classer séparément.

Malgré ces limitations et ces possibles améliorations, le projet nous a permis d'établir un modèle permettant assez bien de prédire le statut cancéreux ou non de patientes. Travailler sur ce projet nous a permis de mieux comprendre les concepts de Deep Learning, de l'importance du pre-processing, de la création et de l'évaluation de modèles de réseaux neuronaux. Cela nous a aussi permis de comprendre les enjeux, et les difficultés de travailler avec des données médicales brutes, et a été une expérience enrichissante.

Bibliographie

1. van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9:2579-2605 (2008).
2. Benjamin Perret, Enseignant-Chercheur ESIEE - LIGM, Introduction au traitement et à l'analyse d'images avec OpenCV. <https://perso.esiee.fr/~perretb/I5FM/TAI/>
3. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.
4. Tan, Mingxing, and Quoc Le. "Efficientnetv2: Smaller models and faster training." *International conference on machine learning*. PMLR, 2021
5. (Li H., Reichert M., Lin K., et al. Differential diagnosis for pancreatic cysts in ct scans using densely-connected convolutional networks. In : 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2019. p. 2095-2098)
6. Pham, H. H., Nguyen Trung, H., & Nguyen, H. Q. (2022). VinDr-Mammo: A large-scale benchmark dataset for computer-aided detection and diagnosis in full-field digital mammography (version 1.0.0). *PhysioNet*. <https://doi.org/10.13026/br2v-7517>.
7. Sawyer-Lee, R., Gimenez, F., Hoogi, A., & Rubin, D. (2016). Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM) [Data set]. The Cancer Imaging Archive. <https://doi.org/10.7937/K9/TCIA.2016.7O02S9CY>
8. Morid, Mohammad Amin, Alireza Borjali, and Guilherme Del Fiol. "A scoping review of transfer learning research on medical image analysis using ImageNet." *Computers in biology and medicine* 128 (2021): 104115.

Annexes

1) Exemple de résultat donné par le modèle YOLOX

breast-detection-vdydv/2 (latest) ✕



Confidence Threshold:

0% 100%

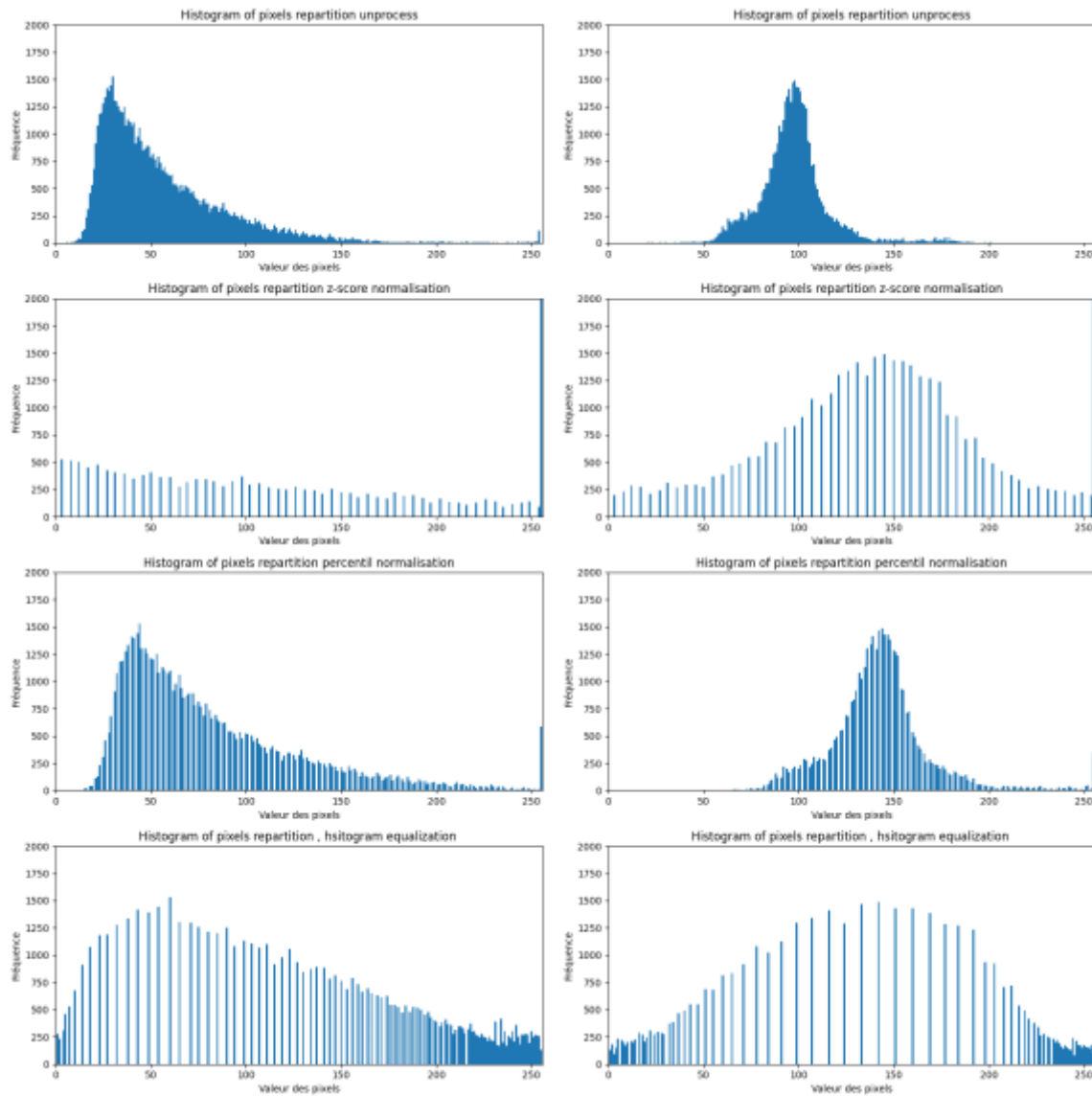
Overlap Threshold:

0% 100%

```
{
  "predictions": [
    {
      "x": 139.5,
      "y": 236.5,
      "width": 277,
      "height": 461,
      "confidence": 0.953,
      "class": "Breast",
      "class_id": 0
    }
  ]
}
```

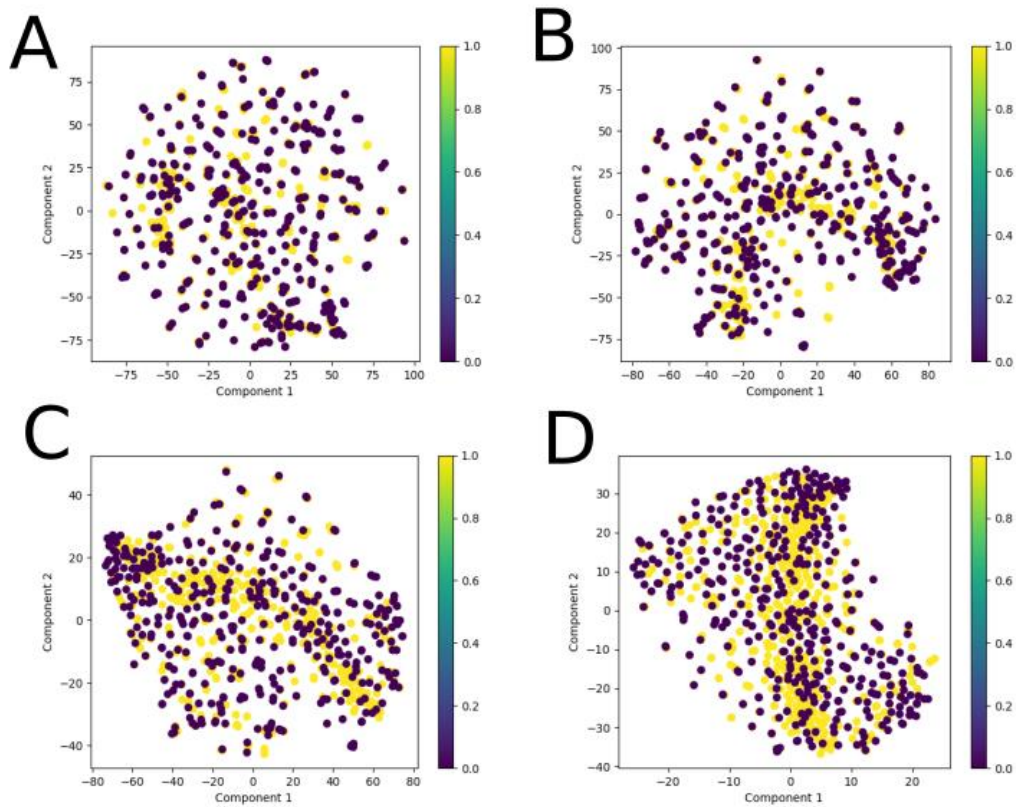


2) Histogrammes de Normalisation



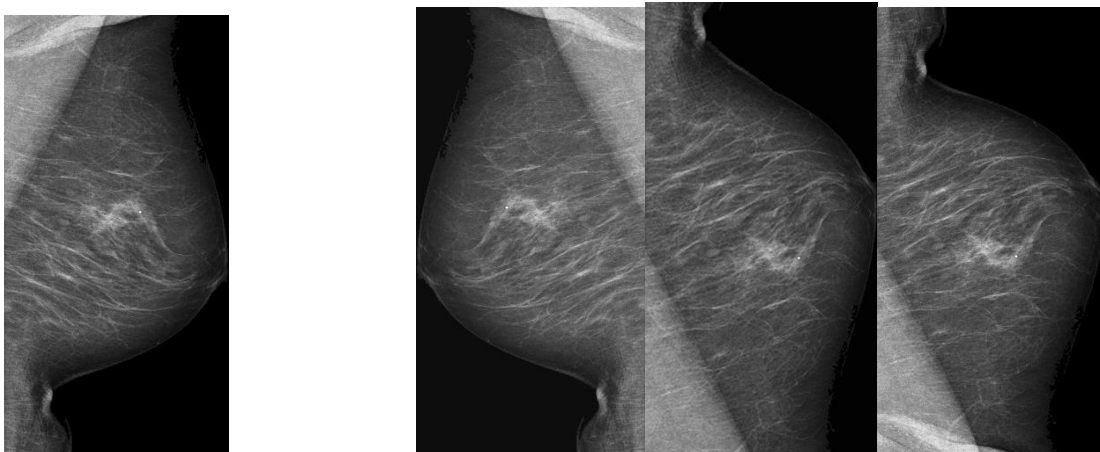
Histogrammes de la répartition des pixels selon le traitement appliqué à l'image. A droite ceux de l'image à faible luminosité, à gauche ceux de l'image à forte luminosité

3) Résultats TSNE



T-SNE sur les deux premières dimensions. En jaune les images cancéreuses et en violets les saines A) perplexité 2 B) perplexité 5 C) perplexité 10 D) perplexité 50

4) Exemple de Data Augmentation



5) Métriques d'évaluation

