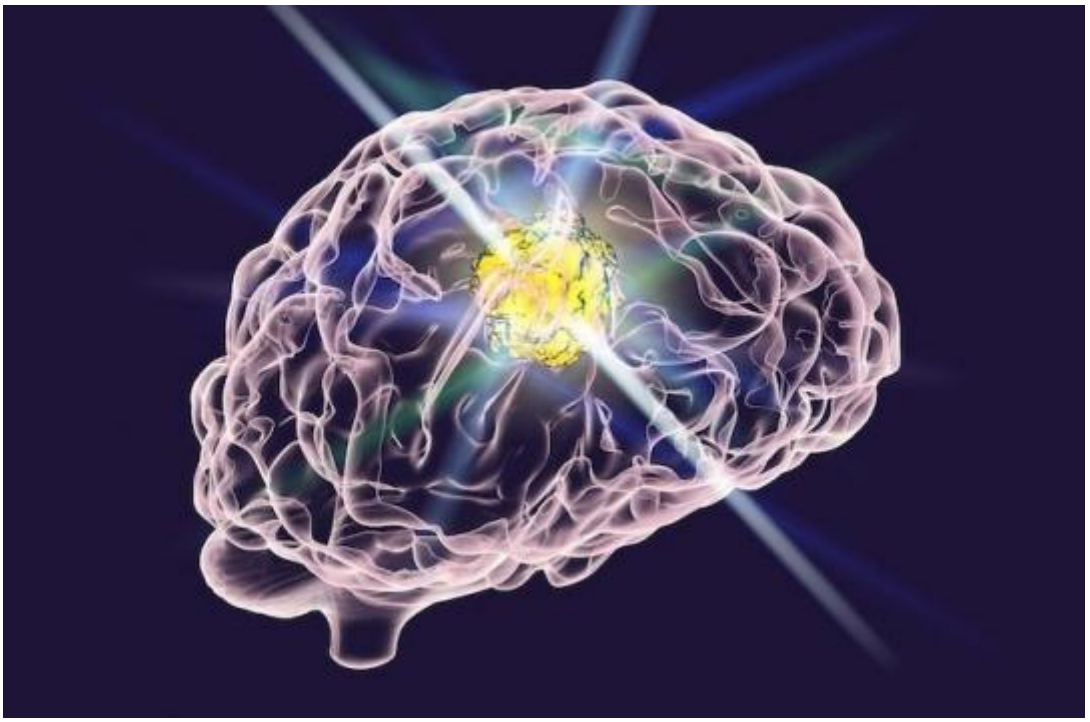
UNIVERSITÉ DE NANTES

IAE NANTES
ÉCONOMIE & MANAGEMENT

Classification d'IRM dans le cadre de la détection des tumeurs au cerveau



Luc LE MEE - Antoine PETITEAU

M2 EKAP - SVM & Réseaux de neurones

Novembre 2021

RSNA®
Radiological Society
of North America

MICCAI

Sommaire

1. Introduction

Dans un monde actuellement touché par une épidémie que l'on pensait au départ éphémère, nous nous rendons compte que cette épidémie persiste. Cependant, ce n'est pas la première fois que l'humanité est confrontée à un virus. Bien que l'ampleur n'ait pas toujours été aussi importante, ces virus ont également marqué l'histoire. Les virus sont problématiques et nous nous en rendons fortement compte de nos jours avec la Covid-19.

Néanmoins, il existe d'autres maladies plus mortelles que les virus et qui perdurent dans le temps, et ce, depuis que la médecine a permis de les détecter. C'est notamment le cas des cancers qui touchent une grande partie de la population. Il existe une multitude de cancers qui touchent différentes parties du corps. La partie du corps concernée dans l'étude que nous avons mené est le cerveau. Nous avons traité un type de tumeur particulier : le glioblastome. Cette tumeur est à la fois la plus courante et la plus mortelle. Pour un adulte, la survie médiane à cette tumeur est de moins d'un an. Pour détecter la présence de cette tumeur, une méthode courante est de rechercher une séquence génétique, du nom de méthylation du promoteur MGMT, qui est un facteur informant sur la présence, ou non, d'une tumeur au cerveau. Il s'agit d'un biomarqueur.

L'objectif de notre étude est de prédire la présence de cette séquence à partir de scanners IRM afin d'anticiper l'apparition d'une tumeur et ainsi réduire le nombre d'interventions chirurgicales tout en affinant le type de thérapie nécessaire. Cela aiderait les patients atteints de cancer du cerveau à recevoir des traitements moins invasifs.

Pour répondre à cette problématique la RSNA (Radiological Society of North America) et la MICCAI Society (Medical Image Computing and Computer Assisted Intervention Society) se sont associés pour améliorer le diagnostic et la planification

du traitement des patients atteints de glioblastome. Ils ont ainsi lancé une compétition sur Kaggle[1] avec une base de données que nous décrirons.

Pour commencer, nous présenterons les données mises à disposition pour notre travail. Nous étudierons tout particulièrement les images issues des scanners. Dans un second temps, nous présenterons la méthode employée pour prédire l'apparition de tumeur au cerveau à partir des images issues des scanners. Ensuite, nous exposerons les résultats de notre modèle. La partie suivante sera consacrée à une analyse critique de notre modèle et de ses résultats. Enfin, avant de conclure, nous proposerons des pistes d'amélioration pour perfectionner notre modèle et corriger les problèmes que nous avons repérés.

[1] Lien de la compétition Kaggle : https://www.kaggle.com/c/rsna-miccai-brain-tumor-radiogenomic-classification/overview (Consulté le 13/11/2021).

2. Présentation des données

Les données du concours fournies par les deux organisations sont structurées en 3 bases. Les bases d'entraînement et de validation qui nous sont fournies et la base test qui est gardée privée. Celles qui nous concernent sont donc la base d'entraînement et celle de validation. Chacune de ces bases contient un certain nombre d'individus identifiés par un numéro à 5 chiffres. La base d'entraînement appelée train est composée de 585 individus et celle de validation appelée test est composée de 87 individus. Pour chacun de ces dossiers (individus), il y a quatre sous-dossiers qui correspondent à différents types d'imagerie qui ne mettent pas en évidence les mêmes aspects.

2.1. Structure des données

La base de données contient 2 colonnes. La première : "BraTS21ID", contient l'identifiant de l'individu. L'identifiant permet notamment de trouver le dossier contenant les images associées à cet individu. Une deuxième colonne : "MGMT_value", correspond à notre variable à prédire. La valeur est 1 lorsque le patient est atteint de glioblastome et 0 sinon. Nous avons donc 2 classes, le phénomène à prédire est binaire.

Ci-dessous, nous présentons les premières lignes de la base train pour illustrer notre propos :

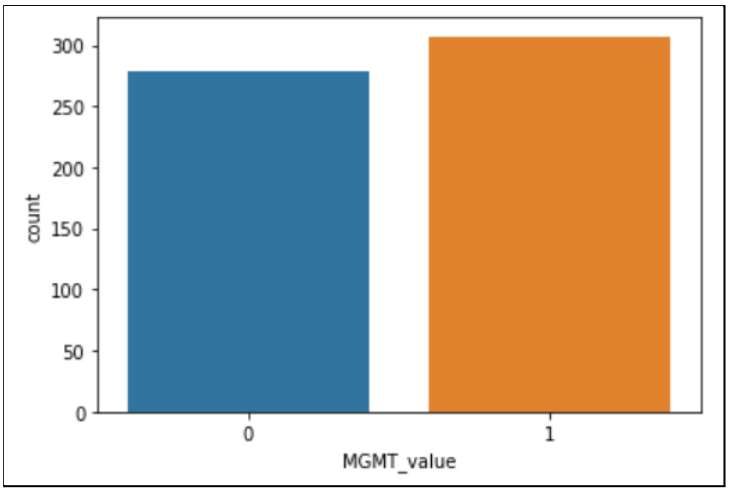
Tableau 1 - Premières lignes de la base “train”

[3]:

	BraTS21ID	MGMT_value
0	0	1
1	2	1
2	3	0
3	5	1
4	6	1
5	8	1
6	9	0
7	11	1
8	12	1
9	14	1

Nous pouvons remarquer que certains ID sont manquants. Ils se trouvent dans la base test de notre projet. À noter que les 3 individus : 109, 123 et 709 ont été retirés de la base de données à cause de défaillance sur leurs images. Cette information nous a été transmise par les organisateurs de la compétition. Sur cette base train, la répartition de nos 2 classes est la suivante :

Graphique 1 - Répartition de la présence de MGMT parmi les patients



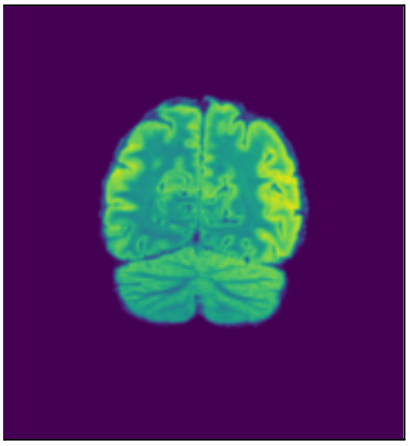
Nous pouvons remarquer que la distribution est bien équilibrée entre les patients ayant la présence de MGMT et ceux qui ne l'ont pas.

2.2. Présentation des images

Les images que nous devons utiliser pour prédire l'apparition des tumeurs au cerveau sont des fichiers DICOM (Digital Imaging and Communications in Medicine). Il s'agit du type standard de fichiers pour des données issues d'imageries médicales. Notre projet comporte presque 140 Go d'images. Face à ce volume conséquent d'informations, nous n'avons ni les machines adaptées ni les connexions internet suffisamment puissantes. C'est la raison pour laquelle nous ne pourrons traiter qu'une partie des images.

Un scanner du cerveau se présente comme ceci :

Illustration 1 - Scanner du cerveau issus de la base train

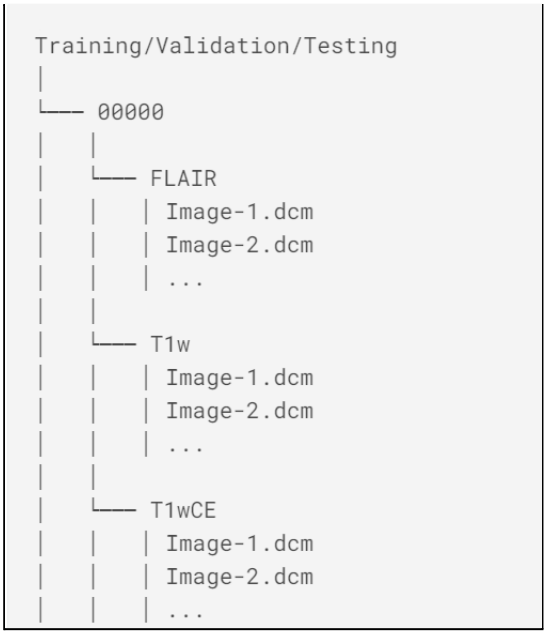


Les images sont au format 512x512. Après plusieurs recherches, nous avons décidé de passer ces images au format 256x256. Cela présente plusieurs avantages. En effet, le temps de chargement des images est grandement diminué tout comme le temps d'exécution de notre modèle. De plus, ce format permet de conserver une qualité très raisonnable d'images.

Pour chaque patient, il existe 4 fichiers contenant des scanners du cerveau. Chacun de ces fichiers contient une séquence de scan en particulier. Il s'agit de ce que l'on

souhaite analyser lors du scan. Ces 4 séquences sont appelées, dans notre projet, “FLAIR”, “T1w”, “T1wCE” et “T2w”. De manière générale, les séquences T1 vont surtout chercher à mettre en évidence l’eau que contient la partie scanner, tandis que les T2 (FLAIR est une séquence T2) mettent plutôt en évidence la graisse. En réalité c’est plus complexe que ça et il existe des exceptions, mais puisque l’intérêt de ce dossier ne porte pas sur la définition de ces séquences, nous n’allons pas rentrer dans les détails. Pour illustrer cela, nous pouvons présenter l’arborescence ci-dessous :

Illustration 2 - Arborescence des fichiers contenant les scanners



Sur l’image ci-dessus, nous pouvons voir l’arborescence des fichiers pour le premier patient. Dans chaque séquence, il y a plusieurs images représentant chacune un moment de la phase de scans. Il est possible de reconstituer le scanner complet en faisant afficher à la suite et dans l’ordre les images du scanner. À noter qu’il n’y a pas le même nombre d’images en fonction des séquences, du type de coupe et en fonction des individus. Cela est une difficulté à prendre en compte pour la suite. Par exemple, pour le premier individu de la base, le nombre d’images par coupe est le suivant :

FLAIR : 400 images

T1w : 33 images

T1wCE : 129 images

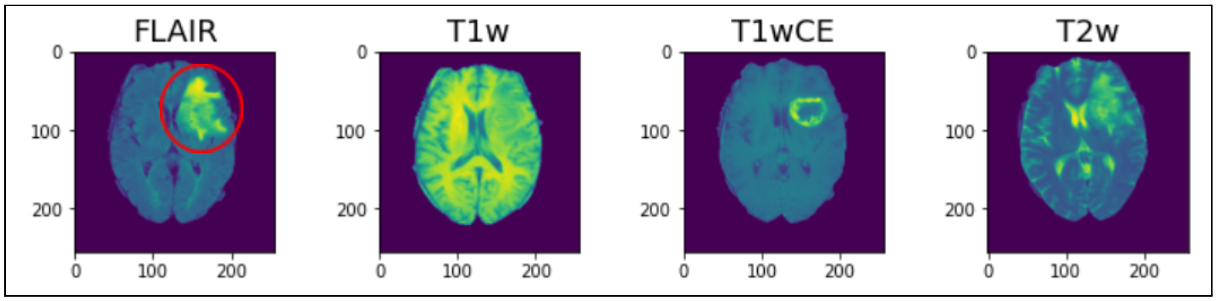
T2w : 408 images

Pour le deuxième individu :

FLAIR : 129 images
T1w : 31 images
T1wCE : 129 images
T2w : 384 images

Ci-dessous, nous visualisons des images correspondant aux 4 différentes séquences pour un individu malade.

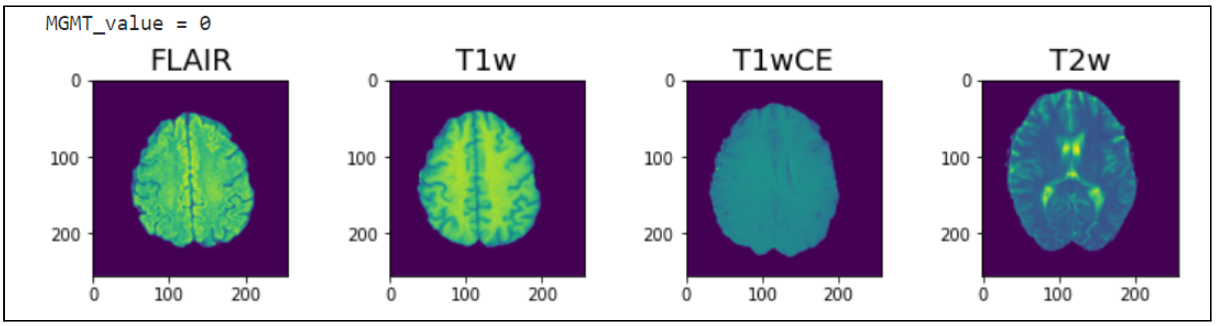
Illustration 3 - Visualisation des 4 types de coupe pour un patient atteint de glioblastome (MGMT_value = 1)



Nous pouvons remarquer que la présence de MGMT est très marquée sur la séquence FLAIR et se différencie bien du reste de la masse cérébrale. Plusieurs recherches confirment que la séquence FLAIR est bien la plus pertinente à utiliser dans notre modèle. Pour chaque séquence, nous avons fait afficher une image correspondant à un moment avancé dans le scanner. De plus, la présence de MGMT est aussi visible pour la séquence T1wCE et encore plus légèrement pour la T2w. Sur la séquence T1w nous ne pouvons pas distinguer de “taches” sur le scanner.

Par comparaison, nous pouvons regarder à quoi ressemble les scanners d’un individu sain :

Illustration 4 - Visualisation des 4 types de coupe pour un patient sain (MGMT_value = 0)

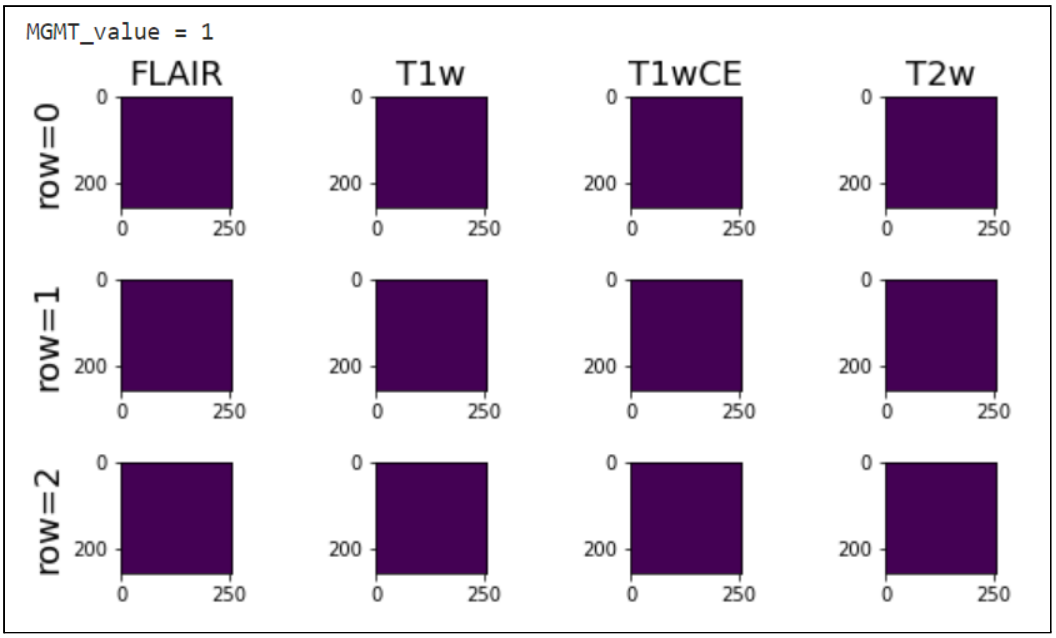


Nous pouvons voir qu'aucune "tâche" n'est visible au niveau du cerveau sur ces images issues des scanners. Lorsqu'une tâche est visible, c'est que l'individu est malade. Ainsi, nous allons devoir construire un algorithme capable de détecter la présence de ces tâches sur les images.

Il a été évoqué précédemment que le volume des images disponibles est trop important pour que nous puissions toutes les traiter convenablement.

Nous remarquons aussi que les premières images des séquences ne sont pas les plus pertinentes :

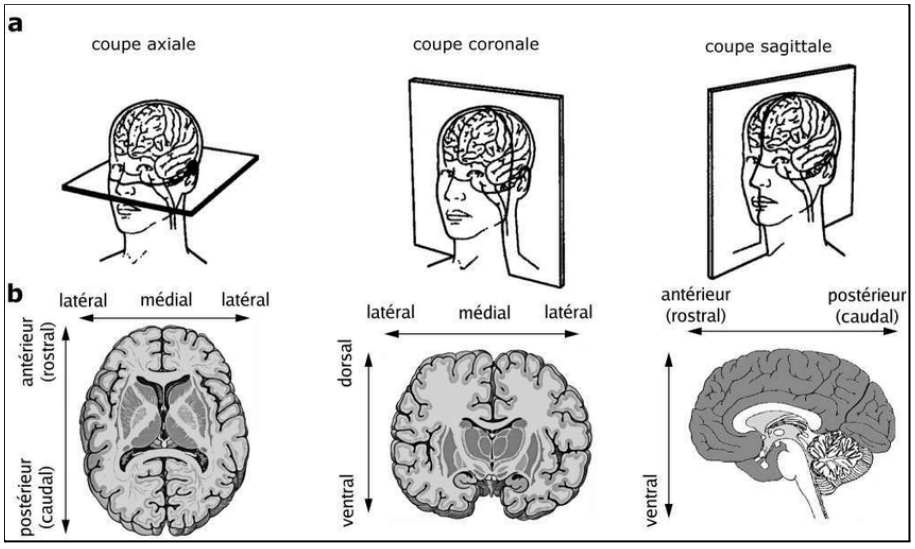
Illustration 5 - Premières images des coupes pour un individu malade



Les premières images des scanners, ainsi que les dernières, ne sont pas intéressantes. En effet, les détails du cerveau n'apparaissent qu'à un stade plus avancé des scanners. Ainsi, la solution envisagée est de sélectionner, pour chaque individu, l'image centrale de chaque séquence. C'est-à-dire l'image présente au milieu du scanner, celle qui contient le plus de détails. Nous avons décidé d'agir ainsi, car cela réduit considérablement le volume des données à utiliser dans le modèle tout en conservant les informations les plus importantes. Si nous étions équipés de machines plus puissantes, nous aurions probablement agi autrement. Ainsi, pour chaque individu, nous obtenons une image par séquence.

Finalement, les scans ont été effectués avec trois coupes ou plans différents, la coupe sagittal, la coupe coronal et la coupe transverse (ou axiale). Pour illustrer la différence entre les plans, nous pouvons observer l'illustration ci-dessous :

Illustration 6 - Premières images des plans pour un individu malade



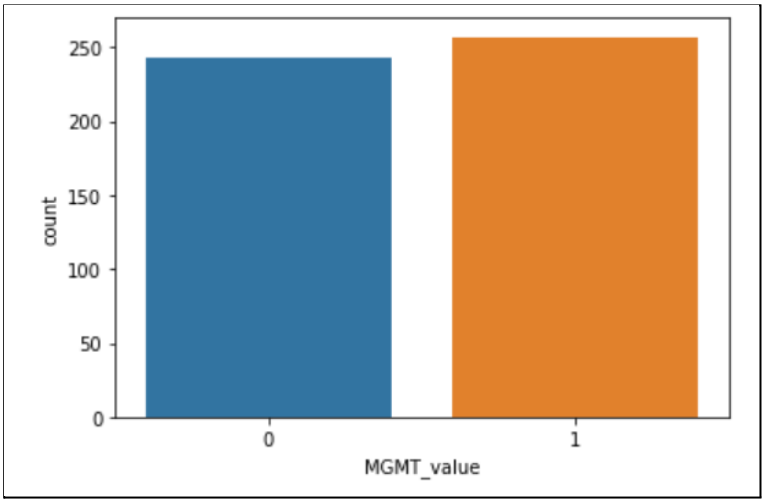
Nous pouvons voir que ces différents plans proposent un angle d'observation du cerveau différent. Par conséquent, il paraît difficile de ne pas prendre en compte ce facteur, car l'information observée sur le scan dépendra de la séquence, comme vu précédemment, mais également du plan.

Cependant, pour chaque séquence, le type de plan pouvait différé entre chaque coupe, mais également d'un individu à un autre. Il faudrait donc prendre en compte pour chaque séquence et chaque individu, le type de coupe observé.

2.3. Échantillonnage

Afin d'étudier la qualité de notre futur modèle, nous avons décidé de séparer la base "train" pour en extraire un échantillon de validation. Pour rappel, la base train contient 582 patients[2]. De manière aléatoire, nous sélectionnons 500 individus qui restent en base train et 82 qui seront utilisés pour la validation. Nous pensons que cette répartition est intéressante car elle conserve un échantillon important pour l'apprentissage de notre modèle. Nous commençons par vérifier que la répartition du phénomène à prédire est pertinente dans la base train :

Graphique 2 - Répartition du phénomène à prédire dans la base train



Nous constatons que la répartition est équilibrée et qu'elle se rapproche de la répartition initiale. Cet échantillon est pertinent à utiliser. Nous pouvons aussi en déduire que l'échantillon test est équilibré et, lui aussi, pertinent pour la suite.

[2] IL y a 585 patients initialement moins 3 patients retirés à cause de problèmes sur leurs scanners.

3. Méthode employée

Pour réaliser notre étude, nous avons décidé de travailler avec les réseaux de neurones convolutifs (en anglais : Convolutional Neural Network, CNN) par l'intermédiaire de la libraire Keras sur Python. La librairie Keras se base sur Tensorflow et est relativement simple à prendre en main.

Nous avons choisi cette méthode car elle est très utilisée et a déjà fait ses preuves dans le domaine de la classification d'images médicales. Nous aurions aussi pu tenter d'utiliser la méthode SVM mais en nous basant sur les travaux existant sur notre projet, nous avons compris que l'utilisation de la méthode CNN était la meilleure des solutions en termes de prévision. Deux autres méthodes auraient pu être tentées, il s'agit des k plus proches voisins (en anglais : k-Nearest Neighbors, k-NN) et des forêts aléatoires (en anglais : Random Forest, RF).

Dans notre cas, la méthode CNN va analyser les pixels des images. En effet, une image est perçue comme un tableau de pixel. Dans notre cas, les images ont trois dimensions : la hauteur, la largeur et les couleurs (RGB). Chaque image va être analysée par un noyau de convolution, appelé Kernel. Ainsi, chaque pixel va se voir attribuer un poids.

Pour évaluer la qualité du modèle, la métrique "LOSS" est utilisée. Celle-ci se calcule en fonction de l'erreur entre la prévision du modèle et la valeur réelle.

Concrètement, notre modèle CNN est le suivant :

Model: "sequential"

Layer (type)	*Output Shape*	*Param #*
conv2d (Conv2D)	*(None, 254, 254, 32)*	*320*
max_pooling2d (MaxPooling2D)	*(None, 127, 127, 32)*	*0*
conv2d_1 (Conv2D)	*(None, 125, 125, 64)*	*18496*
max_pooling2d_1 (MaxPooling2	*(None, 62, 62, 64)*	*0*

conv2d_2 (Conv2D)	*(None, 60, 60, 64)*	*36928*
flatten (Flatten)	*(None, 230400)*	*0*
dense (Dense)	*(None, 64)*	*14745664*
dropout (Dropout)	*(None, 64)*	*0*
dense_1 (Dense)	*(None, 2)*	*130*

Total params: 14,801,538
Trainable params: 14,801,538
Non-trainable params: 0

Notre réseau de neurones contient plusieurs couches. À noter que nous avons appliqué un Dropout car nous avons constaté du potentiel sur-apprentissage sur nos premiers modèles. Le Dropout permet de désactiver certains neurones. Cette action a permis à la précision du modèle, en validation, de gagner 2 points de pourcentage.

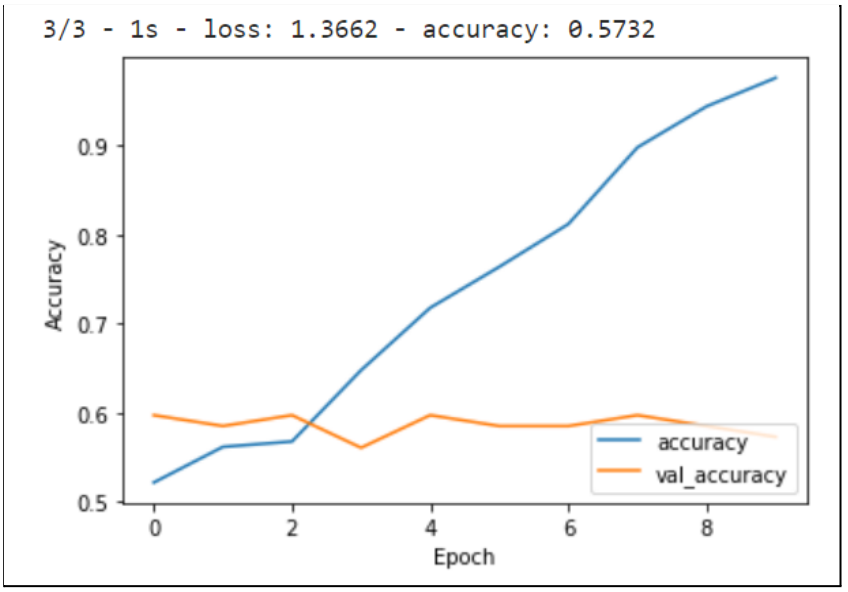
Sur ce modèle, nous avons appliqué l'optimiseur d'Adam. Cela est utile pour la descente de gradient. L'optimiseur d'Adam se base sur les moments d'ordre 1 et 2. Parmi les scanner disponibles, nous avons utilisé la séquence FLAIR car c'est elle qui permet de détecter le mieux la présence du biomarqueur MGMT. C'est ce que nous avons observé en visualisant les images et cette intuition a été confirmée par les autres participants de la compétition. En effet, nous avons obtenu cette information dans l'espace "Discussion" de la compétition sur Kaggle.

Pour finir, il convient de préciser le nombre d'Epoch demandé dans notre modèle.
Nous avons tenté plusieurs valeurs : 8, 10, 12 et 15 Epoch. À chaque fois, nous avons regardé l'impact sur la précision en validation. Nous en avons conclu que le fait d'utiliser 10 Epoch était le plus pertinent. De plus, Le fait d'utiliser 10 Epoch plutôt que 15 permet d'obtenir un modèle dont le temps d'exécution est plus faible.

4. Présentation des résultats

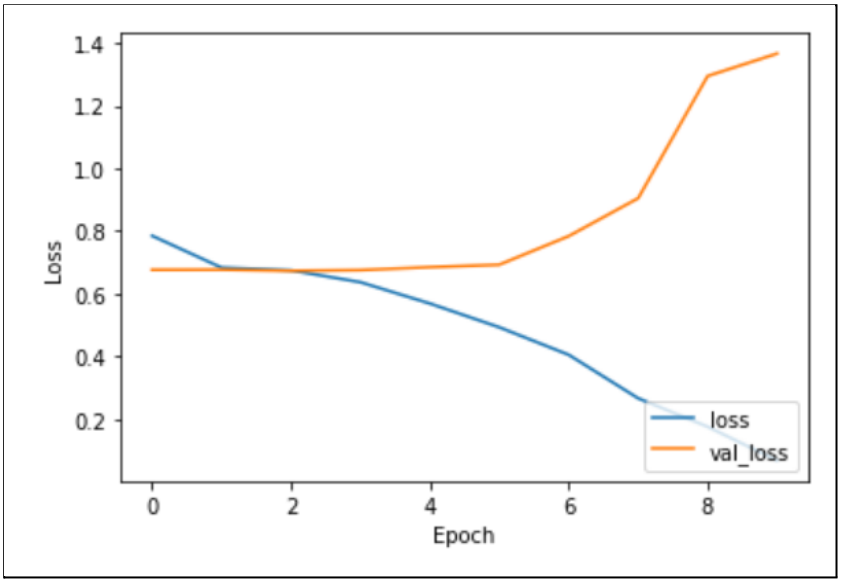
Nous allons, à présent, montrer les résultats de notre modèle. Le taux de précision, en calibration, est proche de 100 %. Nous atteignons la valeur de 97,60 % au bout du dixième Epoch. La précision en validation est plus décevante, nous obtenons 57,32 % sur notre échantillon. Nous pouvons visualiser l'évolution des taux de précisions au cours de l'exécution du programme :

Graphique 3 - Évolution de la précision du modèle durant l'apprentissage



Nous pouvons remarquer que la précision en calibration (courbe bleue) est en croissance relativement régulière durant la phase d'apprentissage. Cependant, la précision en validation (courbe orange) semble relativement constante voire en légère décroissance durant la phase d'apprentissage. Cette observation est étonnante et sera étudiée dans la partie suivante ("5- Les limites et pistes de réflexion sur notre étude").

Graphique 4 - Fonction de perte (LOSS)



Nous pouvons voir que la métrique LOSS diminue lors de l'apprentissage pour la base train. La descente par les gradients s'effectue correctement. Cependant, à partir de la 5e itération, la valeur LOSS augmente pour les données en validation. Cela n'est pas souhaité. Il s'agit d'un défaut dans le cadre de notre travail.

5. Les limites de notre étude

Après avoir présenté notre sujet, les données et notre modèle, nous souhaitons porter un regard critique sur notre travail.

Tout d'abord, il convient de préciser que nous avons été limité par notre niveau en programmation avec Python. En effet, une initiation à Python nous a déjà été présentée. Cependant, nous avons rarement eu l'occasion de manipuler ce langage. Ainsi, nous avons démarré ce projet avec seulement des connaissances très basique en Python. Durant notre étude, nous avons fortement amélioré notre niveau, mais certaines parties du script de programmation nous ont demandé beaucoup d'énergie pour, finalement, peu de gain.

Ensuite, nous avons été limités par la puissance de nos ordinateurs. Comme précisé auparavant, le volume de données était très important. N'étant que peu équipés pour ce projet, nous avons dû nous adapter en réduisant considérablement le volume de données en entrée dans notre modèle. Cela a bien évidemment un impact sur la performance en termes de prévision.

Enfin, nous avons été limités par le temps. Il est probable que certaines équipes participant à la compétition Kaggle ont travaillé à plein temps, voire avec des équipes nombreuses, sur ce projet. Nous avons fait de notre mieux pour proposer un modèle cohérent dans les délais.

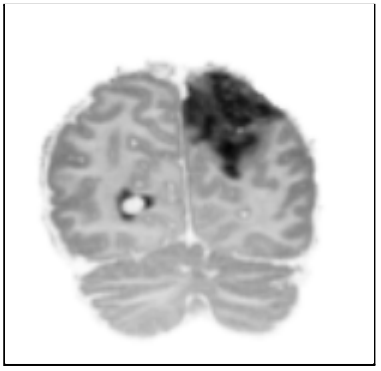
6. Les pistes de réflexion sur notre étude

Il convient maintenant de proposer des pistes de réflexion pour améliorer notre travail. Il s'agit d'éléments auxquels nous avons pensé que nous n'avons pas pu tester. Soit par manque de moyen, soit par manque de temps.

Premièrement, avec des moyens plus conséquents, il serait utile de tenter d'apporter plus de données en entrée pour le modèle. C'est-à-dire qu'il ne faudrait pas prendre seulement l'image médiane de chaque coupe, mais plusieurs images à chaque fois. Cela augmenterait les chances de détecter la présence de MGMT sur les scanner.

Ensuite, il pourrait être pertinent de transformer les images en noir et blanc ou en nuance de gris. Cela pourrait faire ressortir davantage les tâches liées à l'apparition de MGMT. Ci-dessous, nous pouvons voir une transformation en nuance de gris sur une image issue d'une séquence FLAIR d'un patient malade :

Illustration 7 - Scanner en nuance de gris pour un patient malade (MGMT = 1)



Cette idée n'a pas vu le jour car nous n'avons pas réussi à transformer toutes nos images en nuances de gris. En effet, les fichiers de type Dicom (.dcm) ne se manipulent pas comme des images plus classiques. Les fonctions de bases ont échoué sur ce type d'image. La recherche autour de cette solution étant devenue trop chronophage, nous l'avons écartée.

Pour continuer, nous avons mentionné précédemment qu'il existe 3 types de plan parmi les scanners. Un plan est une orientation du scanner vis-à-vis du cerveau. Ainsi, pour chaque individu, nous ne savons pas à l'avance quel type de plan a été utilisé. Il faut visualiser les images pour le constater. Il serait probablement intéressant de distinguer chaque type de plan. Nous nous demandons si l'angle utilisé pour réaliser le scanner ne pourrait pas impacter la détection du biomarqueur MGMT par notre modèle.

Pour continuer, notre réseau pourrait être spécifié autrement. Nous avons décidé de la structure du réseau en nous basant sur des travaux déjà existants. Nous avons choisi une structure relativement simple qui s'avère efficace dans de nombreuses situations (voir le modèle dans la partie "3-Méthode employée"). Peut-être que l'ajout de couches ou la suppression de couches pourraient améliorer notre modèle. Également, le type de neurones est ici ReLU mais d'autres types comme les neurones sigmoïdes pourraient être envisagés. Par manque de temps et dans un objectif de conserver un modèle simple, nous n'avons pas pu tenter toutes les combinaisons de paramètres et de structures possibles.

Enfin, le modèle pourrait être amélioré en ajoutant davantage de Dropout. Nous l'avons vu, l'ajout d'un Dropout a permis d'améliorer la précision en validation. Nous avons tenté d'ajouter un autre Dropout mais cela n'a pas été suivi par une amélioration en termes de prévision. Cependant, peut-être faudrait-il chercher un autre emplacement, entre différentes couches du réseau, pour spécifier le Dropout. Nous avons l'intuition qu'il y a peut-être du sur apprentissage dans notre modèle car lorsque la fonction de perte augmente au cours de l'apprentissage pour les données en calibration, cette fonction de perte tend à augmenter en validation. Ainsi, davantage d'apprentissage semble nuire à la qualité de prévision en validation.

7. Conclusion

Pour conclure, nous avons participé pour la première fois à une compétition Kaggle dans le but de s'exercer à l'analyse de données dans un environnement à la fois stimulant et complexe. Notre objectif était de prédire l'apparition du biomarqueur MGMT sur des images issues de scanners du cerveau. Par ce travail, l'intérêt est d'améliorer la détection de la tumeur du cerveau nommée glioblastome. Ainsi, il s'agit d'une mission de santé publique pouvant améliorer la prise en charge des patients. Pour rappel, les données ont été publiées par la RSNA (Radiological Society of North America) et la MICCAI Society (Medical Image Computing and Computer Assisted Intervention Society). Il s'agit d'un sujet qui nous a beaucoup intéressé étant tous les deux sensibles aux démarches de santé publique.

Pour commencer, nous nous sommes emparé du sujet en lisant toutes les informations transmises par les organisateurs de la compétition. Il s'agit d'un sujet complexe qui a nécessité des recherches pour bien comprendre toutes les dimensions de l'étude. Nous avons ensuite débuté la programmation, en Python, en nous intéressant aux données. C'est-à-dire que nous avons étudié leur volumétrie, leur structure et leur contenu. Il s'agissait de la première fois que nous traitions des données de type image. Cette étape nous a donné des intuitions essentielles pour la suite de notre travail. Nous avons par exemple compris l'importance de la coupe FLAIR et ainsi sélectionné cette coupe pour notre modèle.

En parallèle de ce travail, nous avons étudié les informations transmises par les autres candidats dans l'espace "Discussion" de la compétition sur Kaggle. Ce travail de recherche nous a permis de mieux comprendre le sujet, d'orienter le travail de programmation et de confirmer, ou non, nos intuitions. Puis, nous avons commencé à modéliser en faisant varier les paramètres du modèle CNN avec Keras. Nous nous sommes arrêtés en obtenant un modèle relativement simple, mais que nous pouvions comprendre. Nous avons choisi de réaliser un modèle de type réseau de neurones convolutif car il semble que cette technique soit tout à fait appropriée pour la prédiction à partir d'images.

Ce projet a été pour nous l'occasion de nous familiariser avec l'environnement de Kaggle. Nous avons réalisé notre travail avec le langage Python sur des notebooks Kaggle. Également, ce projet nous a permis de manipuler Git et GitHub afin d'organiser et de versionner notre travail. Ces outils, nouveaux pour nous, se sont avérés très utiles dans le cadre du travail de groupe.

Pour terminer, nous ne sommes pas suffisamment satisfaits de la qualité de prévision de notre modèle sur le jeu de données de validation. Cependant, nous avons tenté d'apporter des pistes de réflexion pour améliorer notre modèle. Ce projet nous a permis de monter en compétence sur Python et de tester en condition réelle un réseau de neurones convolutif.

Sources

Chane M. (2021), "Classification des images médicales : comprendre le réseau de neurones convolutifs (CNN)", https://www.imaios.com/fr/Societe/blog/Classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-CNN *(Consulté le 12/11/2021).*

Cours "SVM et réseau de neurones" dispensés par Véronique CARIOU et Jeff ABRAHAMSON dans le cadre du Master Économétrie et Statistiques de l'IAE de Nantes.

Informations transmises par les organisateurs de la compétition : https://www.kaggle.com/c/rsna-miccai-brain-tumor-radiogenomic-classification/overview *(Consulté le 13/11/2021).*

Inspiration pour la programmation en python : "Brain Tumor : EDA for starter(English version)" : https://www.kaggle.com/chumajin/brain-tumor-eda-for-starter-english-version *(Consulté le 12/11/2021).*

TensorFlow, "Réseau de neurones convolutifs (CNN)", https://www.tensorflow.org/tutorials/images/cnn *(Consulté le 12/11/2021).*