Brain Tumor Classification

Projet deep learning

Réalisé par : ZITANE Zakariae (DS)

Encadré par : Mr. RIHI Ayoub

Année universitaire : 2023/2024

Table des matières

Introduction :

Dans ce projet, nous explorons l'application du deep learning pour la classification d'images d'IRM cérébrales. Nous visons à distinguer entre différentes catégories de tumeurs cérébrales, ainsi qu'à identifier les cas sans tumeur. L'accent est mis sur l'optimisation des performances du modèle pour minimiser les erreurs, notamment les faux négatifs, en utilisant des techniques avancées de traitement d'images et de réseaux de neurones convolutionnels.

Liste des figures :

I. Prétraitement des données :

La base de données contient des images IRM du crâne humain. Il y'a 4 types d'image ce qui correspond à 4 modalités dans la variable Y :

- glioma : 1321 images dans la base d'entrainement et 1311 images dans la base de test.
- Meningioma : 1339 images dans la base d'entrainement et 306 images dans la base de test.
- notumor : 1595 images dans la base d'entrainement et 405 images dans la base de test.
- Pituitary : 1457 images dans la base d'entrainement et 300 images dans la base de test.

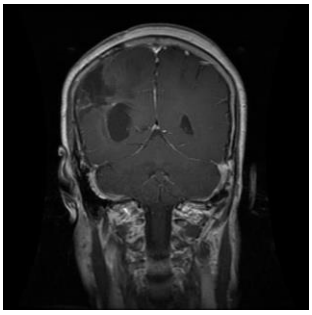

Figure 2 : exemple d'IRM glioma

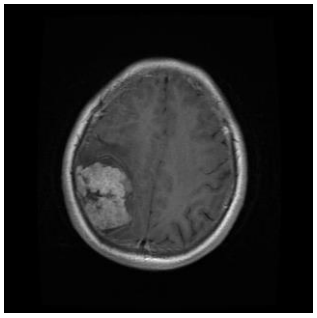

Figure 1 : exemple d'IRM meningioma:

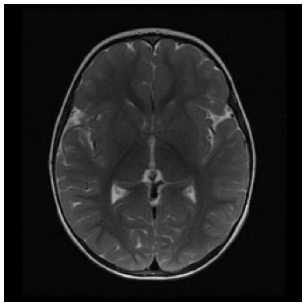

Figure 4 : exemple d'une IRM sans aucune tumeur

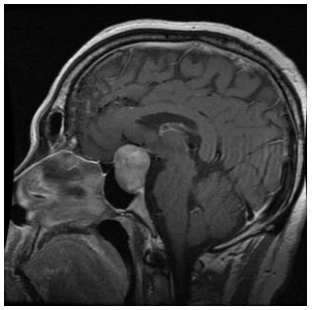

Figure 3 : exemple d'IRM pituitary

Etant donné que notre base de données contient quatre classes, entrainer un modèle à différencier entre les quatre classes d'un seul coup risque de nous donner un modèle avec des performances très faibles. On va donc opter pour un modèle capable de détecter avec une grande précision les IRM avec tumeurs des IRM sans tumeur.

Le but sera de minimiser le risque d'erreur de classer une IRM avec tumeur en une IRM sans tumeur.

Commençons par un traitement d'image. On va utiliser un ensemble de techniques sur les images pour les préparer à l'entrainement des modèles. Ces techniques sont les suivantes :

- **Le redimensionnement :** c'est une technique de prétraitement d'images qui consiste à ajuster la taille de l'image. Cette technique peut être utilisée pour réduire la taille de l'image afin de réduire le temps de traitement. Le redimensionnement peut également être utilisé pour augmenter la taille de l'image en interpolant de nouvelles valeurs de pixels à partir des pixels existants. Cependant, il convient de noter que le redimensionnement peut entraîner une perte de qualité d'image, en particulier lorsqu'une image est agrandie.

- **La normalisation :** c'est une technique qui consiste à ajuster la plage de valeurs des pixels de l'image à une plage spécifique. Cette technique peut être utile pour faciliter le traitement des images, en particulier lorsque les plages de valeurs des pixels varient considérablement entre différentes images. La normalisation peut être effectuée en soustrayant la moyenne de l'image et en divisant par l'écart-type, ou en utilisant d'autres méthodes telles que la normalisation min-max. La normalisation peut aider à améliorer la qualité de l'image en augmentant le contraste et en réduisant les différences de luminosité entre les pixels. Cependant, il est important de noter que la normalisation peut également avoir des effets sur les caractéristiques de l'image, tels que la texture ou les bords, et il est donc important de prendre en compte les effets potentiels sur l'image finale lors de la sélection de la méthode de normalisation appropriée.

- **L'augmentation :** L'augmentation de données est une technique qui consiste à créer de nouvelles images à partir d'images existantes en effectuant des transformations telles que la rotation, le zoom, le recadrage ou le changement de luminosité. Cette technique est souvent utilisée pour augmenter la taille de l'ensemble de données d'apprentissage et améliorer la précision des modèles d'apprentissage automatique. L'augmentation de données peut aider à augmenter la diversité et la variabilité des données d'apprentissage, ce qui peut améliorer la capacité des modèles à généraliser et à reconnaître des motifs dans de nouvelles images.

- **La décoloration :** La décoloration est une technique couramment utilisée en traitement d'images qui consiste à convertir des images colorées en niveaux de gris. Cette technique permet de réduire la complexité des données d'entrée en utilisant des images à une seule couche de couleur, ce qui peut être bénéfique en termes de temps de traitement et de ressources informatiques nécessaires pour entraîner un modèle. La décoloration peut aider à améliorer l'efficacité de l'entraînement du modèle en réduisant la dimensionnalité des données d'entrée. Cela peut également permettre au

modèle d'apprendre plus rapidement les caractéristiques importantes des données.

Pour automatiser le traitement des images, on va définir une fonction qui effectue toutes les transformations citées précédemment :

```python
def traitement_img (img) :
    # Changement des couleurs en nuances de gris
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Changement de la taille de l'image
    img = cv2.resize(img, (128, 128))
    # Normalisation des pixels de l'image (division sur 255)
    img = img / 255
```

Figure 5 : fonction de traitement des images

Après le traitement des images et la création du dataset d'entrainement qui contient 5712 images, voici la répartition des deux classes dans ce dataset :

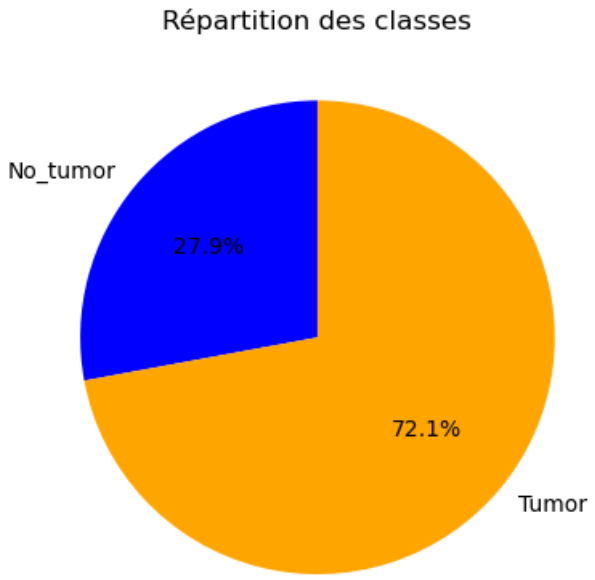

Figure 6 : répartition des deux classes dans le dataset d'entrainement

La répartition des deux classes est équilibrée, on peut donc utiliser ce dataset pour entrainer le modèle qu'on souhaite utiliser.

On va entrainer le modèle avec 75% des données de ce dataset et le tester sur les 25% restante. On dispose également d'un ensemble de validation pour valider le modèle plus tard.

II. Construction du modèle de classification des deux classes principales :

II.1. Architecture du réseau de neurones :

On va commencer la construction d'un réseau de neurones convolutionnel (également appelé réseau de neurones convolutifs). Le modèle sera créé en séquence, ce qui signifie que les couches seront empilées les unes sur les autres de manière linéaire. Voici l'architecture qu'on va adopter pour notre réseau :

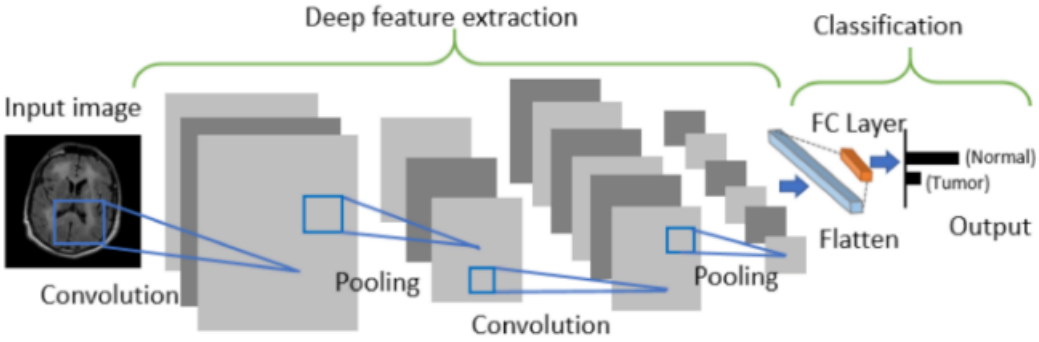

Figure 7: architecture du réseau de neurones

Le modèle sera comme suit : une couche de convolution (avec une fonction d'activation ReLu) suivie d'une couche de pooling (redimensionnement de l'image), ensuite on rajoute une couche de dropout de 20% pour éviter le sur-apprentissage puis à nouveau une couche de convolution et une couche de pooling et pour finir la couche de classification et la couche de sortie qui aura une fonction d'activation sigmoide. Le modèle comportera donc une couche d'entrée, 7 couches cachées et une couche de sortie

II.2. Optimisation des hyper-paramètres :

On va initialiser des boucles pour entrainer des modèles avec toutes les combinaisons possibles des hyper-paramètres afin de déterminer les plus optimales. On va se baser sur la précision obtenue sur la base de test pour comparer les différentes combinaisons testées.

Les différentes combinaisons qu'on va tester seront des combinaisons entre le nombre de filtre (kernel) à appliquer dans les différentes couche de convolution et la taille de la matrice de ces kernels. Les valeurs testées sont les suivantes :

- Nombre de kernel : 16, 32, 64 dans la couche initial et à chaque couche on double la valeur testé. Exemple : dans la première itération la première couche à 16 kernel, la seconde couche en a 32 et la troisième en a 64. Dans la seconde itération la première couche aura 32 kernels.

- La taille de la matrice du kernel : c'est une matrice carrée, elle prendre donc les valeurs suivantes : 1*1, 2*2, … et 6*6.

On va finalement utiliser le modèle avec une matrice 3*3 et un nombre de kernel = 64 car il obtient les meilleures performances. Notre modèle a une accuracy de 77% et ne classifie aucune image comme faux négatifs parmi les 1428 images de l'ensemble de test.

Conclusion :

Le modèle développé a démontré une capacité significative à classer avec précision les images IRM cérébrales. Avec une précision globale de 77% et une efficace détection des cas de tumeur, ce projet souligne l'importance et l'efficacité du deep learning dans le domaine médical.