

Devoir : Classification de tumeurs cérébrales à partir d'images IRM avec CNN

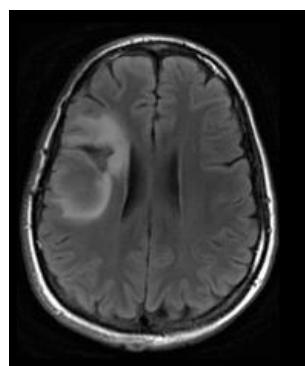
Objectifs du TP

L'objectif de ce travail pratique est de :

1. Concevoir un réseau de neurones convolutionnel (CNN) pour classer les images d'IRM du cerveau selon la présence et le type de tumeur.
2. Apprendre à paramétriser un CNN (nombre de filtres, taille des noyaux, Dropout, batch size, etc.).
3. Expérimenter plusieurs combinaisons de paramètres et observer leur impact sur la performance.
4. Comparer votre modèle avec des modèles pré-entraînés tels que ResNet50 et DenseNet121.

Contexte médical

Les tumeurs cérébrales sont des masses anormales de cellules dans le cerveau. L'analyse des images IRM (Imagerie par Résonance Magnétique) permet de détecter ces tumeurs précocement. Les techniques d'intelligence artificielle peuvent aider à automatiser cette détection et à améliorer la précision du diagnostic.



Jeu de données

Nous utiliserons le dataset Brain Tumor MRI Dataset disponible sur Kaggle :

<https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>

Structure du jeu de données :

/Training

```
    └── glioma  
    ├── meningioma  
    ├── pituitary  
    └── no_tumor
```

/Testing

```
    └── glioma  
    ├── meningioma  
    ├── pituitary  
    └── no_tumor
```

Chaque dossier contient plusieurs centaines d'images IRM annotées selon le type de tumeur.

Partie 1 : Création et paramétrage de votre CNN

1. Préparation des données

Avant d'entraîner le modèle, il faut :

- Charger les images avec ImageDataGenerator.
- Redimensionner les images à (128, 128, 3) ou (224, 224, 3).
- Normaliser les pixels (rescale = 1./255).
- Créer les ensembles d'apprentissage, de validation et de test.

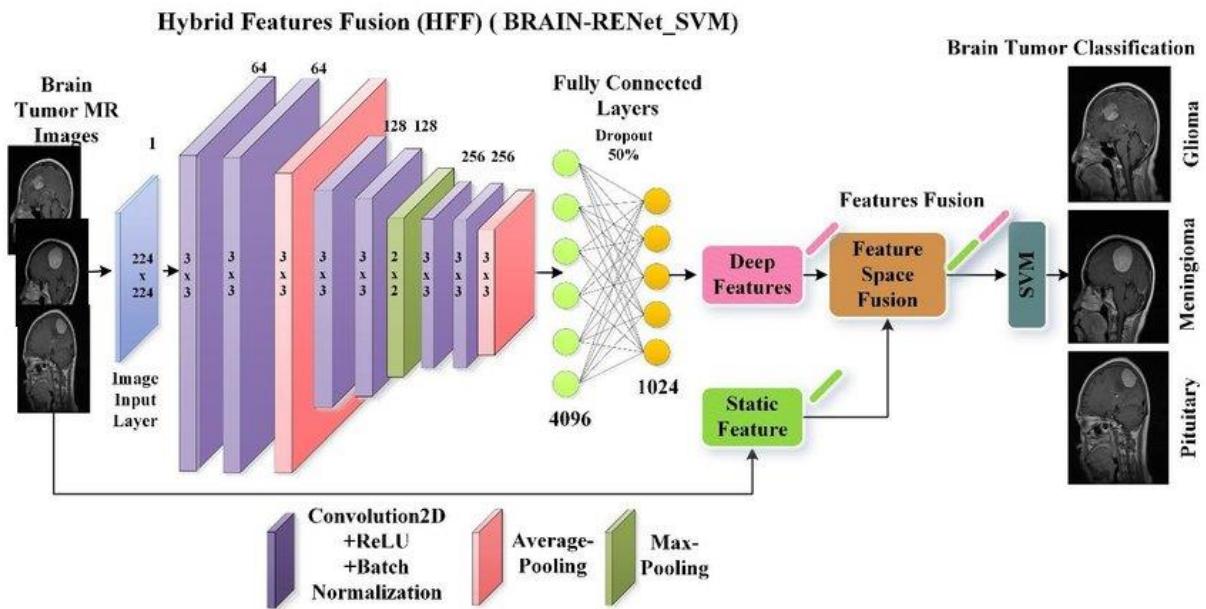
2. Conception du modèle CNN

Vous allez concevoir votre propre modèle CNN, en empilant plusieurs blocs :

Conv2D → MaxPooling2D → Flatten → Dense → Dropout → Dense

Paramètres à faire varier :

- Le nombre de filtres : 32 → 64 → 128 → 256
- La taille du noyau (kernel_size) : 3×3, 5×5
- Le taux de Dropout : 0.3, 0.4, 0.5
- Le batch size : 16, 32, 64
- Le nombre d'époques : 10 à 20



3. Expérimentation et tests

Testez au moins trois variantes de votre modèle.

Pour chaque essai, notez vos paramètres et résultats dans un tableau :

Essai	Filtres	Kernel	Dropout	Batch Size	Accuracy (train)	Accuracy (val)	Commentaire
A	32→64→128	3×3	0.3	32	—
B	16→32→64→128	3×3	0.5	64	—
C	32→64→128	5×5	0.4	32	—

Quelle combinaison semble offrir le meilleur équilibre entre précision et stabilité ?

Observez également si votre modèle fait du surapprentissage (overfitting).

4. Évaluation du modèle

Pour le modèle retenu :

- Évaluez la performance sur le jeu de test.
- Calculez les métriques : accuracy, précision, rappel, F1-score.
- Affichez la matrice de confusion.
- Interprétez les erreurs de classification éventuelles.

Partie 2 : Comparaison avec les modèles pré-entraînés

Les modèles pré-entraînés (ou *Transfer Learning*) permettent d'utiliser des architectures déjà entraînées sur de grands ensembles d'images (ex. ImageNet).

Ils peuvent offrir de très bons résultats en classification médicale.

Modèles proposés :

- VGG16
- ResNet50
- DenseNet121

1. Tâche à réaliser

- a. Choisissez un modèle pré-entraîné (par exemple, ResNet50 ou DenseNet121).
- b. Remplacez la dernière couche par votre propre “tête” de classification (Dense(4, softmax)).
- c. Entraînez uniquement la tête, puis testez votre modèle.
- d. Comparez les résultats obtenus avec ceux de votre CNN.

2. Question finale à rédiger dans votre rapport

- a. Comparez la performance (accuracy, stabilité, temps d'entraînement) entre votre CNN et le modèle pré-entraîné.
- b. Quels sont les avantages et inconvénients de chaque approche ?
- c. Dans le contexte du diagnostic médical réel, lequel recommanderiez-vous ?
- d. Quelle serait la prochaine amélioration possible (data augmentation, fine-tuning, explicabilité, etc.) ?

Livrable attendu

Vous devez rendre un notebook clair et bien commenté (ou un rapport PDF) contenant :

1. Le prétraitement des données.
2. Les architectures CNN testées et les résultats.
3. Les graphiques d'entraînement et d'évaluation.
4. La comparaison avec un modèle pré-entraîné.
5. Une conclusion synthétique (10 à 15 lignes).