

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOU D MAMMERRI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D' INFORMATIQUE
DEPARTEMENT D' AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies Filière
: Automatique

Spécialité : Automatique et Système

Présenté par
Wissam Bengana
Leticia kaddour

Thème

Classification Des tumeurs Cérébrales par les Réseaux de Neurones Convolutifs (CNN)

Mémoire soutenu publiquement le 02/10/2024..... devant le jury composé de :

Mme MCA	Saadia UMMTO,	Alkama Président
Mme MAA	Ouiza UMMTO,	Nait Belaid Encadrant
Mme MCB	Kahina UMMTO,	Titouche Examineur
Mme MCA	Zohra UMMTO,	Haliche Examineur

Remerciements

Nous tenons à exprimer avant tout notre profonde reconnaissance à **ALLAH** pour Sa guidance et Son aide précieuse tout au long de ce travail. Sans Sa bénédiction, rien n'aurait été possible.

Nous adressons nos plus sincères remerciements à **Madame NAIT BELAID Ouiza**, notre encadrante, pour son soutien attentif et ses conseils avisés durant toute l'élaboration de ce mémoire. Sa compétence, sa grande disponibilité et son engagement ont été essentiels à la réussite de ce travail. Grâce à ses orientations précieuses, nous avons su surmonter les défis rencontrés. Ses encouragements constants et sa pédagogie ont largement contribué à la progression de notre étude.

Nous exprimons également notre gratitude aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner, ainsi que pour leurs précieuses remarques et suggestions, qui enrichiront ce mémoire. Leur expertise et leurs conseils sont d'une grande valeur pour nous.

Nos remerciements vont aussi à nos familles, dont le soutien moral, la patience et les sacrifices ont été essentiels tout au long de ce projet. Leur présence réconfortante a grandement contribué à cette réussite.

Enfin, nous exprimons notre gratitude à tous nos amis et collègues, qui ont su nous encourager, partager leurs idées, et nous soutenir dans les moments les plus exigeants de ce travail. À chacun d'entre vous, un grand merci.

Dédicace

Je dédie ce modeste travail

À mes précieux parents, pour leur amour inestimable, leur confiance, leur soutien indéfectible, leurs sacrifices et toutes les valeurs qu'ils m'ont transmises.

À mes sœurs, « Yasmine », « Nabila » et « Lynda » qui m'ont aidée et encouragée, ainsi qu'à mon frère « Nassim ».

À Leticia, ma chère amie, avant d'être mon binôme.

À toutes mes amies, qui ont toujours été présentes.

À tous ceux qui m'ont soutenue.

Wissam

Dédicace

Je dédie ce modeste travail

À mes précieux parents, pour leur amour inestimable, leur confiance, leur soutien indéfectible, leurs sacrifices et toutes les valeurs qu'ils m'ont transmises.

À mes sœurs, « Melissa » et « Zina Lisa », qui m'ont aidée et encouragée, ainsi qu'à mon frère « Rayan ».

À ma nièce « Eva », qui est une source de joie.

À Wissam, ma chère amie, avant d'être mon binôme.

À toutes mes amies, qui ont toujours été présentes.

À tous ceux qui m'ont soutenue.

Leticia

Résumé

L'objectif de ce projet est d'explorer l'utilisation de l'apprentissage profond, et plus spécifiquement des réseaux de neurones convolutifs (CNN), pour classier trois types de tumeurs cérébrales à partir d'images d'IRM. Trois modèles CNN ont été utilisés pour cette tâche : DenseNet121, VGG19 et un modèle concaténé combinant ces deux architectures. L'objectif principal est de comparer les performances de ces modèles pour la classification des tumeurs en utilisant des métriques telles que la précision, le rappel et le F1-score.

L'environnement de développement choisi est Google Colab, une plateforme cloud qui permet d'accéder à des ressources de calcul puissantes, notamment des GPU, essentiels pour l'entraînement efficace des modèles d'apprentissage profond. Le prétraitement des données inclut la normalisation des images et leur redimensionnement à une taille uniforme. Des techniques comme le transfert d'apprentissage et la régularisation par Dropout ont été utilisées pour optimiser les performances des modèles.

En résumé, ce projet vise à classier avec précision les tumeurs cérébrales à partir d'images d'IRM en utilisant différentes architectures CNN. Les résultats montrent que le modèle concaténé offre une précision supérieure par rapport aux modèles individuels, démontrant ainsi l'efficacité de la combinaison de plusieurs architectures pour améliorer la performance de classification.

Mots-clés : Tumeur cérébrale, IRM, Deep learning, Classification, Transfer learning, DenseNet121, VGG19.

Abstract

The objective of this project is to explore the use of deep learning, specifically convolutional neural networks (CNNs), to classify three types of brain tumors from MRI images. Three CNN models were used for this task : DenseNet121, VGG19, and a concatenated model combining these two architectures. The primary goal is to compare the performance of these models in tumor classification using metrics such as accuracy, recall, and F1-score.

The chosen development environment is Google Colab, a cloud-based platform that provides access to powerful computing resources, including GPUs, which are essential for the efficient training of deep learning models. Data preprocessing includes image normalization and resizing to a uniform size. Techniques such as transfer learning and Dropout regularization were used to optimize model performance.

In summary, this project aims to accurately classify brain tumors from MRI images using different CNN architectures. The results show that the concatenated model offers superior accuracy compared to individual models, demonstrating the effectiveness of combining multiple architectures to improve classification performance.

Keywords : Brain tumor, MRI, Deep learning, Classification, Transfer learning, DenseNet121, VGG19.

Table des matières

Table des matières

Liste des tableaux

Table des figures

Introduction générale	1
1 Les tumeurs cérébrales et IRM	3
1.1 Introduction	3
1.2 Quelques éléments d'anatomie du cerveau :	3
1.2.1 Système nerveux périphérique (SNP) :	3
1.2.2 Le système nerveux central (SNC) :	3
1.2.2.1 La moelle épinière :	4
1.2.2.2 L'encéphale :	4
1.2.3 Principaux tissus cérébraux :	5
1.2.3.1 Substance blanche :	5
1.2.3.2 Substance Grise :	6
1.2.4 Liquide céphalo-rachidien (LCR) :	6
1.3 Les tumeurs cérébrales :	6
1.3.1 Tumeurs cérébrales primaires :	6
1.3.2 Tumeurs cérébrales métastatiques :	7
1.4 Classification des tumeurs cérébrales	7
1.4.1 La localisation de la tumeur	7
1.4.2 Le type de tumeur :	7
1.4.3 Le degré d'agressivité de la tumeur :	8
1.5 Les symptômes et le diagnostic	8
1.5.1 Les symptômes :	8
1.5.2 Le diagnostic :	8
1.6 L'imagerie par résonance magnétique	9
1.6.1 Qu'est ce que c'est l'IRM :	9
1.6.2 Fonctionnement de l'IRM :	9
1.7 Signal IRM :	9
1.8 L'IRM cérébral :	10

1.8.1	Observations du cerveau :	10
1.8.2	Séquences d'IRM :	11
1.9	Conclusion	12
2	La classification des images :	13
2.1	Introduction :	13
2.2	Les étapes de la classification :	13
2.3	Méthode de la classification :	15
2.3.1	Méthode non supervisée :	15
2.3.1.1	Méthode de k-means :	15
2.3.1.2	Méthodes Hiérarchiques	16
2.3.2	Méthode supervisé :	17
2.3.2.1	Méthode des k plus proches voisins (kNN) :	17
2.3.2.2	Les machines à vecteurs de support (SVM) :	18
2.3.2.3	Les Reseaux de Neurons :	20
2.4	La Classification des images par CNN :	20
2.5	Conclusion :	22
3	Les Réseaux de Neurons Convolutifs :	23
3.1	Introduction :	23
3.2	Deep learning :	23
3.2.1	Qu'est ce que c'est le deep learning :	23
3.2.2	Historique de deep learning	24
3.3	Les Réseaux de Neurons :	24
3.3.1	Perceptron :	24
3.3.2	Perceptron multi-couches :	25
3.3.3	L'architecture d'un réseau de neurones :	26
3.4	Les Réseaux de Neurons Convolutifs	26
3.4.1	L'architecture d'un réseau de neurones convolutifs :	26
3.4.1.1	Couche de convolution :	27
3.4.1.2	Couche de regroupement(pooling) :	28
3.4.1.3	Couche complètement connectées (fully-connected) :	29
3.4.2	Fonction d'activation :	30
3.4.3	L'optimisation et l'hyperparamètres d'un CNN :	31
3.4.4	Modeles pré-entraînées :	32
3.4.4.1	Un CNN pré-entraînées :	32
3.4.4.2	Transfert learning (apprentissage par transfert) :	32
3.4.5	Les avantages et les inconvénients d'un CNN :	33
3.4.6	Quelque reseaux de neurones convolutifs :	33
3.4.6.1	Densenet :	34
3.4.6.2	Resnet :	36
3.4.6.3	VGGnet :	37
3.4.6.4	LeNet :	38
3.5	Conclusion :	38

4	Implémentation , Résultats et Discussions	39
4.1	Inroduction :	39
4.2	Environnement et outils de développements	39
4.2.1	Environnement matériel	39
4.2.2	Environnement logiciel	40
4.2.2.1	Latex :	40
4.2.2.2	Python :	40
4.2.2.3	Google colab :	42
4.3	Base de données :	44
4.4	Préparation de données :	45
4.5	Modèles CNN utilisées :	46
4.5.1	Modèle Densenet121 :	46
4.5.2	Modèle VGG19 :	46
4.5.3	Modèle Concaténé :	47
4.6	Métriques de performance :	48
4.6.1	Courbes de performance :	49
4.7	Implémentation,Discussions et Comparaison des résultats	50
4.7.1	Implémentation :	50
4.7.2	Résultats et Discussions	51
4.7.2.1	Résultats de Densenet121 :	51
4.7.2.2	Résultats de VGG19 :	53
4.7.2.3	Résultats Modèle Proposé :	56
4.7.3	Comparaison des Résultats	58
4.8	Conclusion :	59
	Conclusion générale	60
	Bibliographie	61

Liste des tableaux

3.1	Tableau comparatif des fonctions d'activation	30
3.2	Avantages et inconvénients des réseaux neuronaux convolutifs (CNN) . . .	33
4.1	Comparaison des résultats des trois modèles CNN	58

Table des figures

1.1	Le système nerveux central	4
1.2	(a).Cerveau, Cervelet, Tronc cérébral, (b) Les trois matières de cerveau. . .	5
1.3	Coupes du cerveau	10
1.4	Séquence IRM (a).pondéré en T2 , (b).pondéré en T1	12
2.1	Processus de la classification	14
2.2	Comparaison entre les méthodes de classification supervisée et non supervisée	15
2.3	Points similaires de l'algorithme KNN	17
2.4	Exemple illustrant différentes séparations de deux classes	18
2.5	Exemple de transformation à l'aide du kernel trick	19
2.6	Un exemple comparatif peut illustrer la différence la classification classique et par CNN.	21
3.1	Schéma de perceptron simple	25
3.2	Schéma simple d'un MLP	25
3.3	Schéma d'un CNN	27
3.4	Convolution	28
3.5	(a)pooling maximal,(b) pooling moyen	29
3.6	Couche complètement connectées	29
3.7	Historique des CNN	34
3.8	La connectivité entre les couches	34
3.9	Blocs denses	35
3.10	Architecture de densenet121	36
3.11	Architecture de Resnet50	37
3.12	Architecture de Vgg19	37
3.13	Architecture de Lenet-5	38
4.1	Quelques bibliothèque utilisées	40
4.2	Page d'accueil	43
4.3	Comment ouvrir un nouveau notebook	43
4.4	Ajouter le programme dans une cellule	44
4.5	Changer le type d'exécution	44
4.6	Choisir GPU	44
4.7	Les images de chaque classe de la base de données.La ligne bleu indique la region de la tumeur	45

4.8	Modèle Densnet121	46
4.9	Modèle VGG19	47
4.10	Modèle Proposé (La concaténation des deux modèles)	48
4.11	Matrice de confusion	48
4.12	Étapes d'Implémentation	50
4.13	Apprentissage du modèle Densenet121	51
4.14	Accuracy du modèle densenet121	52
4.15	Courbe de Precision et Perte du modèle Densenet121	52
4.16	Matrice de confusion de Densnet121	52
4.17	Rapport de Classification de Densenet121	53
4.18	Apprentissage de modèle VGG19	54
4.19	Accuracy et Erreur du VGG19	54
4.20	Les Courbes du modèle VGG19	55
4.21	Matrice de confusion De VGG19	55
4.22	Le rapport de classification du modèle VGG19	56
4.23	Apprentissage du modèle proposé	56
4.24	Accuracy et erreur du modèle proposé	56
4.25	Courbes de precision et de perte du modèle proposé	57
4.26	Matrice de confusion du modèle proposé	57
4.27	Rapport de Classification du modèle proposé	58
4.28	Quelques prédictions	58

Liste des abréviations

- **IA** : Intelligence Artificielle
- **SNP** : Système Nerveux Périphérique
- **SNC** : Système nerveux central
- **LCR** : Céphalo-Rachidien liquide
- **LCS** : Liquide Cérébro-Spinal
- **TDM** : Tomodensitométrie (Scanner)
- **IRM** : Imagerie par Résonance Magnétique
- **TR** : le temps de répétition
- **TE** : le temps d'écho
- **T1** : Temps de relaxation longitudinal
- **T2** : Temps de relaxation transversal
- **CNN** : Réseaux de Neurones Convolutifs
- **k-NN** : k plus proches voisins
- **SVM** : Machines à Vecteurs de Support
- **GAN** : Réseau Antagoniste Génératif
- **VGG** : Groupe de Géométrie Visuelle
- **Adam** : Adaptive Moment Estimation
- **CONV** : couches convolutives
- **MLP** : Perceptron Multicouche (Multilayer Perceptron)
- **ReLU** : Unité de Rectification Linéaire
- **SGD** : Descente de Gradient Stochastique
- **GPU** : Graphics Processing Unit (Unité de Traitement Graphique)
- **CPU** : Central Processing Unit (Unité Centrale de Traitement)
- **TP** : True Positive
- **TN** : True Negative
- **FP** : False Positive
- **FN** : False Negative
- **RAM** : Mémoire vive (Random Access Memory)
- **BN** : Batch Normalization

Introduction générale

L'intelligence artificielle (IA) a profondément transformé de nombreux secteurs au cours de la dernière décennie, en particulier avec l'émergence de l'apprentissage profond (deep learning). Parmi les nombreuses architectures utilisées en apprentissage automatique, les réseaux de neurones convolutifs (CNN) se sont imposés comme une technologie incontournable, notamment pour les tâches de traitement d'images et de vision par ordinateur. Leur capacité à extraire automatiquement des caractéristiques complexes à partir d'images, sans nécessiter une intervention humaine pour la sélection de ces caractéristiques, en fait des outils puissants pour diverses applications allant de la reconnaissance faciale à l'analyse médicale.

Dans ce travail, nous nous intéressons plus spécifiquement à l'application des CNN à la détection et à la classification de tumeurs cérébrales à partir d'images d'IRM. L'imagerie par résonance magnétique (IRM) est l'une des techniques d'imagerie médicale les plus couramment utilisées pour examiner les structures du cerveau et diagnostiquer les anomalies telles que les tumeurs. Les tumeurs cérébrales, qui incluent des types tels que les gliomes, les méningiomes et les tumeurs hypophysaires, nécessitent un diagnostic précis et rapide pour garantir une prise en charge médicale appropriée. Grâce aux avancées dans les techniques de deep learning, il est désormais possible de développer des modèles capables d'analyser automatiquement ces images et de fournir des résultats précis, facilitant ainsi le travail des radiologues et des cliniciens.

Dans ce mémoire, nous avons proposé d'étudier deux modèles de CNN : DenseNet121 et VGG19, qui sont reconnus pour leurs performances dans le domaine de la classification d'images. DenseNet121 se distingue par ses connexions denses, facilitant ainsi le flux d'informations entre les couches du réseau, tandis que VGG19 est apprécié pour sa profondeur et sa simplicité. Une approche de concaténation de ces deux modèles a été proposée, dans le but de combiner leurs forces respectives et d'optimiser les performances en termes de classification des tumeurs cérébrales.

Le transfert d'apprentissage (transfer learning) a joué un rôle clé dans ce travail. Cette technique permet de réutiliser des modèles pré-entraînés sur de grands ensembles de données, comme ImageNet, afin d'améliorer la précision des modèles et de réduire le temps de calcul nécessaire pour les adapter à des tâches spécifiques, ici la classification des tumeurs cérébrales. L'environnement d'implémentation, comprenant des outils tels que Google Colab et les bibliothèques Python comme TensorFlow et Keras, a permis une exécution efficace des modèles et une optimisation des ressources, notamment grâce à l'utilisation des GPU pour accélérer l'entraînement.

Cette étude ne se limite pas à l'implémentation technique des modèles, mais inclut également une analyse approfondie des résultats obtenus. Les courbes de performance, les matrices de confusion et des métriques comme la précision, le rappel et le F1-score ont été utilisées pour évaluer l'efficacité des modèles dans la classification des images. Bien que les performances obtenues soient très encourageantes, elles montrent aussi que des améliorations futures restent possibles, notamment en optimisant les architectures et les méthodes d'entraînement.

TABLE DES FIGURES

Pour ce faire, nous avons structuré notre mémoire en quatre chapitres :

- Le premier chapitre présente l’anatomie du cerveau et introduit l’imagerie par résonance magnétique (IRM), une technique fondamentale pour l’évaluation des tumeurs cérébrales.
- Le deuxième chapitre aborde la classification des images et décrit ces méthodes , ainsi que certains travaux récents d’application des CNN pour la classification d’IRM cérébrales.
- Le troisième chapitre présente en détail les réseaux de neurones convolutifs et explore leur importance dans le domaine de la classification des images.
- Enfin, le quatrième chapitre expose la partie expérimentale de notre travail et discute des différents résultats obtenus.

Chapitre 1

Les tumeurs cérébrales et IRM

1.1 Introduction

Les tumeurs cérébrales représentent un défi majeur en neurologie et en oncologie, nécessitant une compréhension approfondie de l'anatomie cérébrale et des techniques d'imagerie avancées pour leur diagnostic et leur prise en charge. Ce chapitre vise à offrir une vue d'ensemble des tumeurs cérébrales, en commençant par les fondamentaux de l'anatomie du cerveau, puis en explorant les différents types de tumeurs, leurs classifications et leurs symptômes. Une attention particulière sera portée à l'Imagerie par Résonance Magnétique (IRM), outil indispensable dans le diagnostic et le suivi de ces pathologies. Nous examinerons le fonctionnement de l'IRM, ses principes physiques, et son application spécifique dans l'imagerie cérébrale. Cette introduction aux tumeurs cérébrales et à l'IRM fournira une base solide pour comprendre les défis diagnostiques et thérapeutiques associés à ces conditions complexes.

1.2 Quelques éléments d'anatomie du cerveau :

Le système nerveux est un réseau complexe de nerfs et de cellules nerveuses (neurones) qui font circuler des signaux et des messages provenant du cerveau et de la moelle épinière vers différentes parties du corps, et vice versa. Il est constitué du système nerveux central et du système nerveux périphérique[11].

1.2.1 Système nerveux périphérique (SNP) :

LE SNP est composé des nerfs qui transmettent les messages entre la moelle épinière et la périphérie du corps, incluant les bras, les jambes et les organes.

1.2.2 Le système nerveux central (SNC) :

Le SNC comprend l'encéphale et la moelle épinière(voir figure 1.1).

1.2.2.1 La moelle épinière :

Se rattache à l'encéphale au niveau du tronc cérébral et est protégée par les vertèbres, qui forment la colonne vertébrale. Les nerfs émergent de la moelle épinière pour innerver les deux côtés du corps. La moelle épinière fait circuler les signaux nerveux, leur permettant d'aller et venir entre l'encéphale et les nerfs du reste du corps.

1.2.2.2 L'encéphale :

L'encéphale contrôle la plupart des fonctions du corps, dont la perception, les mouvements, les sensations, les pensées, la parole et la mémoire. Il comprend le cerveau, le cervelet, et le tronc cérébral(voir figure 1.2a).

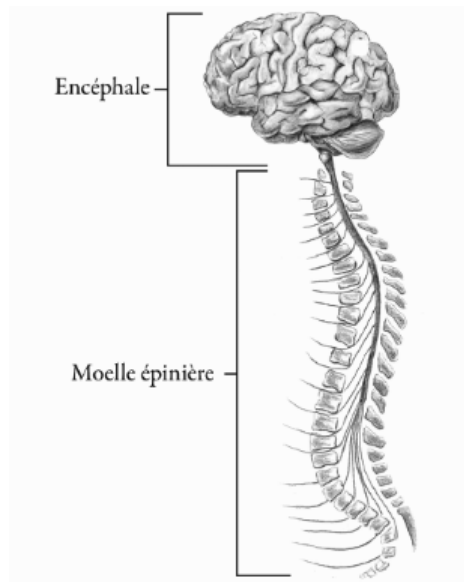


FIGURE 1.1 – Le système nerveux central

- **Cerveau** : Le cerveau appartient au système nerveux central, situé dans la boîte crânienne, c'est un organe essentiel qui assure la régulation de toutes les fonctions vitales. Il est composé de deux hémisphères, droit et gauche, connectés entre eux par le corps calleux composé de substance blanche (axones gainés de myéline). Divisés en lobes, les hémisphères sont les parties les plus développées chez les mammifères[50].
- **Le Cervelet** : Le cervelet situé en parallèle sur deux grandes voies nerveuses (celles qui amènent les messages sensoriels vers les zones du cortex qui en font l'analyse et celles qui partent du cortex et descendent vers les muscles pour les faire contracter). Le cervelet est en partie impliqué dans la coordination des mouvements du corps.

- **Le tronc cérébral** : Il représente un carrefour majeur du système nerveux central entre le cerveau et le cervelet. Il est responsable en particulier de la régulation de la respiration et du rythme cardiaque. C'est également un centre de passage des voies motrices et sensitives, ainsi qu'un centre de contrôle de la douleur.

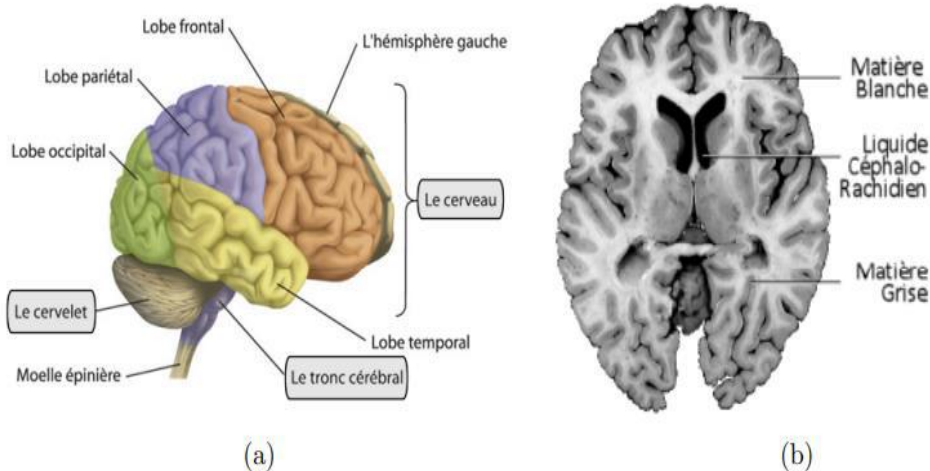


FIGURE 1.2 – (a).Cerveau, Cervelet, Tronc cérébral, (b) Les trois matières de cerveau.

1.2.3 Principaux tissus cérébraux :

Lorsque nous examinons le cerveau, nous remarquons qu'il est principalement composé d'une substance appelée matière blanche. À la surface du cerveau se trouve une fine couche de matière grise connue sous le nom de cortex cérébral. Cette différence de couleur est due au fait que le cortex contient principalement les corps cellulaires des neurones, tandis que la matière blanche est formée de faisceaux d'axones qui relient différentes régions corticales les unes aux autres. Les axones sont responsables du transport des influx nerveux.

1.2.3.1 Substance blanche :

La substance blanche est une catégorie de tissu du système nerveux central, principalement composé des axones myélinisés des neurones. Elle relie différentes aires de la substance grise où se situent les corps cellulaires des neurones. Elle constitue la partie interne du cerveau et la partie superficielle de la moelle épinière. La substance blanche a en fait un aspect rosé à l'œil nu qui est principalement dû à la présence de capillaires sanguins. Ce n'est qu'après avoir été traité au formaldéhyde, pour la conservation, que ce tissu prend sa couleur blanche caractéristique.

1.2.3.2 Substance Grise :

La substance grise, également appelée matière grise, est constituée de neurones, de cellules gliales qui les entourent et les protègent, ainsi que de vaisseaux sanguins. Elle est présente dans l'ensemble du cerveau (hémisphères, cervelet et tronc cérébral), ainsi que dans la moelle épinière[45].

1.2.4 Liquide céphalo-rachidien (LCR) :

Le liquide céphalo-rachidien (LCR) (ou liquide cérébro-spinal, LCS) est le liquide dans lequel baignent le cerveau et le cervelet. D'un volume moyen de 150ml, il est absorbé par le système veineux cérébral et continuellement renouvelé. Ses principaux rôles sont :

- la protection mécanique du système nerveux central contre les chocs par amortissement des mouvements.
- la protection contre les infections, car il contient les médiateurs de l'immunité humorale et cellulaire.
- le transport des hormones entre les différentes régions éloignées du cerveau.

1.3 Les tumeurs cérébrales :

Une tumeur cérébrale est une masse de cellules qui se développe dans le cerveau. Elle peut se former à partir de cellules cérébrales elles-mêmes (tumeurs cérébrales primaires) ou se propager au cerveau à partir d'autres parties du corps (tumeurs cérébrales métastatiques). Ses types sont :

1.3.1 Tumeurs cérébrales primaires :

Elles se forment dans le cerveau et sont classées en fonction du type de cellules à partir desquelles elles se développent. Les types courants de tumeurs cérébrales primaires comprennent les gliomes, les méningiomes, les neurinomes acoustiques et les astrocytomes.

Les principales tumeurs cérébrales primitives sont :

•Les gliomes :

Un gliome est une tumeur qui se développe dans le cerveau ou la moelle épinière, prenant naissance dans les cellules gliales, responsables du soutien et de la protection des neurones. Il existe trois types principaux de cellules gliales pouvant former des tumeurs, chacune ayant des caractéristiques moléculaires et génétiques propres. Ces informations aident à prédire l'évolution du gliome et à déterminer les traitements les plus efficaces. Le type de gliome diagnostiqué influence donc le pronostic et le choix des thérapies. Les trois principaux types de gliomes sont les astrocytomes (incluant les astrocytomes, les astrocytomes anaplasiques et les glioblastomes), les épendymomes (dont les épendymomes anaplasiques, myxopapillaires et sub-épendymomes) et les oligodendrogliomes

(oligodendrogliomes et oligodendrogliomes anaplasiques). Souvent, avant d'obtenir un diagnostic précis, les médecins utilisent le terme "gliome" de manière générale. Ce n'est qu'après une intervention chirurgicale et une analyse de la lésion sous microscope, réalisée par des neuropathologistes, que le type spécifique de gliome est déterminé [32].

• **les méningiomes** : Représentent environ 30 pour-cent des tumeurs cérébrales primitives : ce sont des tumeurs qui se développent à partir de cellules des méninges, en superficie du cerveau. Elles sont plus fréquentes chez les femmes. Dans l'immense majorité des cas, il s'agit de tumeurs bénignes .

• **Tumeurs pituitaires** :

Une tumeur pituitaire est une tumeur qui se développe dans l'hypophyse. Elle peut causer des problèmes hormonaux, des troubles visuels en appuyant sur les nerfs optiques, ou des déficits hormonaux en endommageant l'hypophyse [24].

1.3.2 Tumeurs cérébrales métastatiques :

Elles se propagent au cerveau à partir d'autres parties du corps, telles que les poumons, les seins, les reins, etc. Ces tumeurs sont considérées comme des métastases cérébrales plutôt que des tumeurs cérébrales primaires[9].

1.4 Classification des tumeurs cérébrales

On distingue les tumeurs cérébrales en fonction de 3 caractéristiques :

- La localisation de la tumeur.
- Le type de tumeur.
- Le degré d'agressivité de la tumeur.

1.4.1 La localisation de la tumeur

Le cerveau est organisé en plusieurs zones, qui gèrent chacune des activités spécifiques : le langage, l'équilibre du corps, les battements du cœur, la circulation du sang, la mémoire, etc. Une tumeur peut entraîner des troubles très différents selon la zone dans laquelle elle se développe.

La localisation de la tumeur est également un élément essentiel pour le choix des traitements. Une tumeur située à la surface du cerveau par exemple, est généralement plus facile à extraire qu'une tumeur située au centre du cerveau[30].

1.4.2 Le type de tumeur :

Les tumeurs du cerveau sont classées selon le type de cellule dont elles sont issues, par exemple, les gliomes proviennent des cellules gliales et les méningiomes des cellules des méninges.

1.4.3 Le degré d'agressivité de la tumeur :

Plus la tumeur se développe rapidement, plus elle est jugée agressive. Les tumeurs du cerveau sont classées en différents grades selon leur agressivité. On parle de bas grade pour les tumeurs les moins agressives et de haut grade pour les tumeurs qui le sont davantage [49].

1.5 Les symptômes et le diagnostic

1.5.1 Les symptômes :

L'apparition de symptômes peut être soudaine ou très progressive, selon la rapidité à laquelle la tumeur se développe. Certaines tumeurs ne provoquent aucun symptôme, ce qui signifie généralement qu'elles se développent lentement. Lorsque la tumeur est de petite taille, elle n'entraîne généralement aucun symptôme. Mais plus elle grossit, plus les symptômes peuvent survenir. Les maux de tête constituent le symptôme le plus fréquent, quel que soit le type de tumeur cérébrale. Ils se manifestent souvent le matin, parfois accompagnés de nausées et/ou de vomissements. Ils sont dus à la masse de la tumeur qui comprime la région où elle est localisée[9][24].

Le patient peut aussi présenter :

- Des troubles de la vision.
- Des problèmes d'équilibre.
- Des changements de comportement.
- Des convulsions.
- Des troubles de la mémoire.
- Des difficultés de coordination.

1.5.2 Le diagnostic :

Le diagnostic d'une tumeur fait partie d'un processus clinique précis et complexe qui conduit ultérieurement à une décision thérapeutique appropriée. Pour détecter les tumeurs cérébrales, deux examens sont privilégiés : le scanner (TDM) et l'IRM. Le scanner, utilisant des rayons X, repère environ 80 pour-cent des tumeurs et nécessite souvent l'injection d'un produit spécial pour une meilleure visualisation. Pendant l'examen, le patient demeure allongé sur une table qui traverse un grand anneau émettant des rayons. L'IRM, quant à elle, exploite un champ magnétique, offrant une sensibilité supérieure au scanner. Capable de détecter des tumeurs non visibles au scanner, elle fournit des images détaillées de la taille et de l'emplacement des tumeurs, facilitant ainsi la planification des traitements [24].

1.6 L'imagerie par résonance magnétique

1.6.1 Qu'est ce que c'est l'IRM :

L'IRM, qui signifie imagerie par résonance magnétique, est un examen de radiologie permettant d'observer l'intérieur du corps. Pour cela, un appareil émet un champ magnétique dépourvu de radiations ionisantes.

Le corps, étant composé de 70 pour-cent d'eau, réagit à ses ondes. Les protons d'hydrogène qui le composent captent cette énergie puis la retransmet. Ces signaux sont alors retranscrits sous forme d'images sur un écran d'ordinateur.

Par sa technologie, l'IRM offre une grande précision. En effet, les différents angles captés sont ensuite rassemblés afin d'obtenir une image en 3D de la zone concernée.

Étant dépourvu de radiations ionisantes, l'IRM est un examen qui ne présente aucun danger d'irradiation pour l'organisme. Néanmoins, il existe des contre-indications à la réalisation d'un IRM[40].

1.6.2 Fonctionnement de l'IRM :

Les IRM utilisent des aimants puissants qui produisent un champ magnétique puissant qui force les protons du corps à s'aligner avec ce champ. Lorsqu'un courant de radiofréquence est ensuite pulsé à travers le patient, les protons sont stimulés et perdent leur équilibre, s'opposant à l'attraction du champ magnétique. Lorsque le champ de radiofréquence est désactivé, les capteurs IRM sont capables de détecter l'énergie libérée lorsque les protons se réalignent avec le champ magnétique. Le temps nécessaire aux protons pour se réaligner avec le champ magnétique, ainsi que la quantité d'énergie libérée, varient en fonction de l'environnement et de la nature chimique des molécules. Les médecins sont capables de faire la différence entre différents types de tissus en fonction de ces propriétés magnétiques[26] .

1.7 Signal IRM :

Le corps humain est constitué d'une multitude d'atomes d'hydrogène. L'imagerie par résonance magnétique (IRM) est basée sur les propriétés magnétiques des atomes d'hydrogène. Chaque atome d'hydrogène contient un seul proton chargé positivement, agissant comme un petit aimant en mouvement. Lorsqu'un patient est placé dans un champ magnétique à l'intérieur du tunnel de l'IRM, ces protons s'alignent dans la même direction, créant un équilibre.

Pour obtenir des images, on perturbe cet équilibre en émettant des ondes radio à une fréquence spécifique, ce qui met les protons d'hydrogène dans un état de résonance et les fait tourner. Lorsque les protons reviennent à leur état d'équilibre, cela induit une variation du champ magnétique. Cette variation est détectée par une antenne spéciale et convertie en signal électrique, formant ainsi l'image IRM. En somme, l'IRM repose

sur la capacité des protons d'hydrogène à réagir à un champ magnétique et à émettre un signal détectable lorsqu'ils reviennent à leur état initial.

1.8 L'IRM cérébral :

L'IRM du cerveau permet d'étudier le système nerveux central, mais aussi les différents composants du cerveau. Elle permet de visualiser précisément la substance blanche (partie superficielle) et la substance grise (partie profonde) du cerveau. Mais aussi le système ventriculaire cérébral ainsi que l'état des vaisseaux sanguins (veines, artères). Cet examen est utilisé pour poser un diagnostic sur des symptômes neurologiques et pour détecter des anomalies cérébrales. L'IRM cérébrale peut être réalisée sans ou avec injection d'un produit de contraste (sels de gadolinium) selon les cas[2].

1.8.1 Observations du cerveau :

L'observation par des coupes en deux dimensions du cerveau peut être effectuée selon plusieurs angles de vue. Ainsi, on distingue trois axes anatomiques pour réaliser les coupes[36].

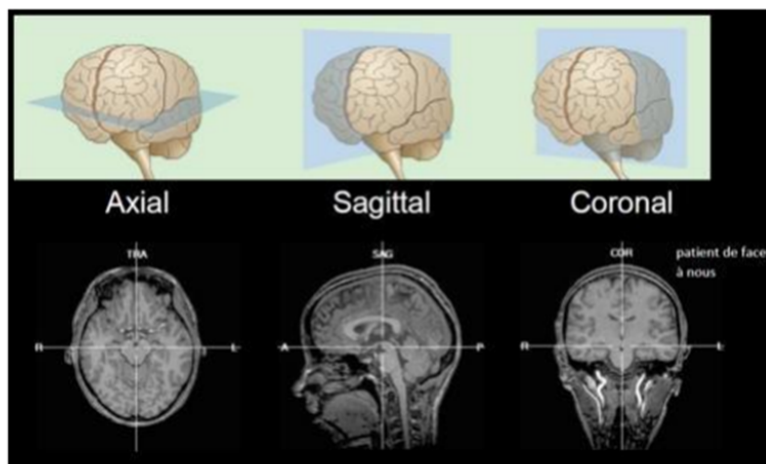


FIGURE 1.3 – Coupes du cerveau

- **Coupe Axiale** : Cette coupe représente une vue de dessus du cerveau. Elle correspond à un plan perpendiculaire au champ magnétique statique.
- **Coupe Sagittale** : Cette coupe offre une vue latérale du cerveau, montrant ce qui se trouve à l'intérieur le long d'un plan parallèle à la ligne qui divise les deux hémisphères cérébraux. Elle est utile pour visualiser les structures médianes du cerveau, telles que le corps calleux.

- **Coupe Coronale** : Cette coupe offre une vue de face du cerveau, permettant d'examiner les structures cérébrales dans leur orientation antéro-postérieure. Elle est acquise sur un plan perpendiculaire aux coupes axiale et sagittale, offrant une perspective complémentaire des structures cérébrales.

1.8.2 Séquences d'IRM :

La séquence d'IRM est une méthode utilisée pour obtenir des images avec des contrastes spécifiques basés sur les propriétés de relaxation des tissus. Deux paramètres réglables cliniquement entrent en jeu lors de l'acquisition : le temps de répétition des séquences d'impulsion (TR), et le temps d'écho (TE)[4].

a .Les images dites « pondérées en T1 » :

T1= Constante de temps propre à chaque tissu, correspond à la repousse de l'aimantation longitudinale de ce tissu. Chaque tissu possède donc sa constante T1.

Plus un tissu possède un T1 rapide, plus ce dernier aura un signal élevé en pondération T1 (hypersignal = blanc). Dans une image pondérée en T1, la graisse apparaît hyperintense.

Plus un tissu possède un T1 lent, plus ce dernier aura un signal faible en pondération T1 (hyposignal = noir). Dans ce cas l'eau est hypointense, de couleur sombre.

Elles sont souvent utilisées pour l'anatomie, dans la mesure où la séquence d'acquisition favorise la détection de l'eau peu mobile c'est à dire intracellulaire (signal élevé pour la substance grise et faible pour les os par exemple).(Figure 1.4 b).

b .Les images dites « pondérées en T2 » :

T2 = Constante de temps propre à chaque tissu correspond à la décroissance de l'aimantation transversale de ce tissu. Chaque tissu possède, aussi, sa constante T2. Elle dépend fortement de la nature des tissus.

Plus un tissu possède un T2 rapide, plus ce dernier aura un signal faible en pondération T2 (hyposignal = noir). Dans une image pondérée en T2, l'eau apparaît hyperintense, de couleur claire. Plus un tissu possède un T2 lent, plus ce dernier aura un signal élevé en pondération T2 (hypersignal = blanc). La graisse apparaît, alors, un peu plus sombre que l'eau(Figure 1.4 a).

Elles sont utilisées comme images fonctionnelles dans la mesure où la séquence d'acquisition favorise la détection de l'eau mobile c'est à dire extracellulaire ou intravasculaire, ce qui constitue, par exemple, une approche du débit sanguin cérébral local.

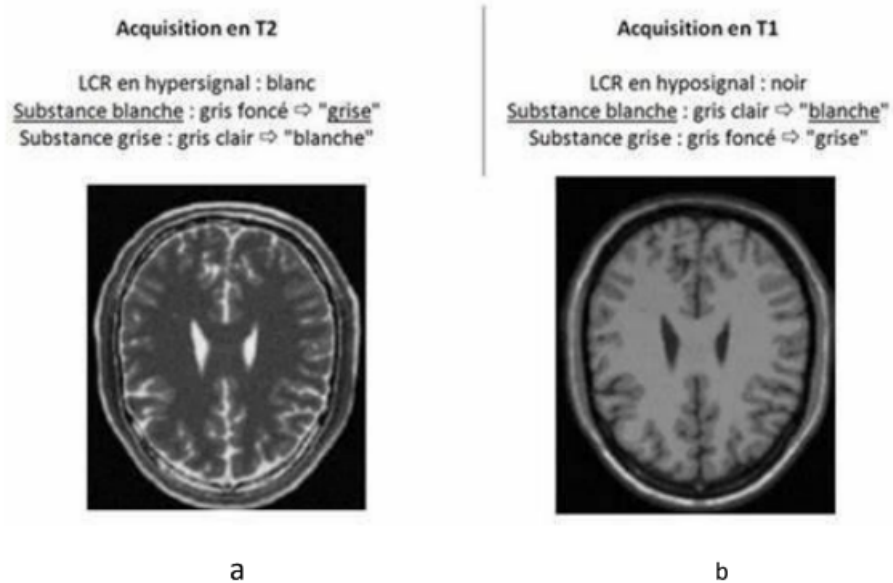


FIGURE 1.4 – Séquence IRM (a).pondéré en T2 , (b).pondéré en T1

1.9 Conclusion

En conclusion, ce chapitre a permis d'explorer l'anatomie cérébrale, les caractéristiques des tumeurs cérébrales, et le rôle de l'IRM dans leur diagnostic et leur suivi. La classification des tumeurs, basée sur leur localisation, leur type et leur degré d'agressivité, souligne l'importance d'un diagnostic précis pour une prise en charge optimale. L'IRM, avec ses différentes séquences et ses capacités d'imagerie multiplan, s'est révélée être un outil inestimable dans ce processus. La compréhension des principes de l'IRM, notamment les séquences T1 et T2, permet une interprétation fine des images obtenues, facilitant ainsi la détection et la caractérisation des tumeurs cérébrales. Cette connaissance approfondie de l'anatomie cérébrale, des tumeurs et de l'imagerie par résonance magnétique constitue un socle essentiel pour les professionnels de santé impliqués dans le diagnostic et le traitement des tumeurs cérébrales, ouvrant la voie à des approches thérapeutiques toujours plus précises et personnalisées.

Chapitre 2

La classification des images

2.1 Introduction :

La classification d'images est une technique de reconnaissance de formes qui attribue une catégorie spécifique à chaque image à l'aide d'un système de classification. Elle est essentielle dans divers domaines, et particulièrement précieuse dans le secteur médical. Elle permet aux médecins de poser des diagnostics plus précis et de surveiller les traitements grâce à des outils offrant une plus grande précision et performance. En facilitant l'analyse automatique des images, la classification améliore considérablement la qualité et l'efficacité des soins médicaux. Ce chapitre se penche sur les différentes étapes de la classification d'images, en explorant les méthodes supervisées et non supervisées, et met un accent particulier sur l'application des réseaux de neurones convolutifs (CNN). Ces techniques révolutionnaires trouvent une utilité croissante dans le domaine médical, notamment pour la classification des images IRM cérébrales, contribuant ainsi de manière significative à l'amélioration du diagnostic et à l'avancement de la recherche médicale.

2.2 Les étapes de la classification :

La classification d'images est un processus systématique qui comprend plusieurs étapes fondamentales pour attribuer des étiquettes à des données visuelles (voir fig.2.1). Chaque étape joue un rôle important dans la réalisation d'une classification précise et efficace[44].

- **Collecte de données** : Réunir un ensemble d'images représentatives des différentes classes que vous souhaitez classifier.
- **Prétraitement des données** : Les données d'entrée sont prétraitées pour être conformes avec le modèle de classification.
- **Extraction des attributs** : Extraire des informations pertinentes des images (par exemple, des bords, des textures, ou des représentations plus complexes via

des réseaux de neurones.

- **Choix d'un algorithme de classification** : Le choix d'un algorithme de classification implique de sélectionner la méthode la mieux adaptée pour attribuer des étiquettes à des données en fonction de leurs caractéristiques.
- **Entraînement du modèle** : Un modèle de classification est formé à partir des données d'entraînement en utilisant un algorithme d'apprentissage automatique.
- **Évaluation du modèle** : Le modèle formé est évalué en utilisant des données de validation pour déterminer son niveau de performance.
- **Prédiction** : Le modèle formé peut être utilisé pour prédire la classe d'une entrée donnée en dehors .

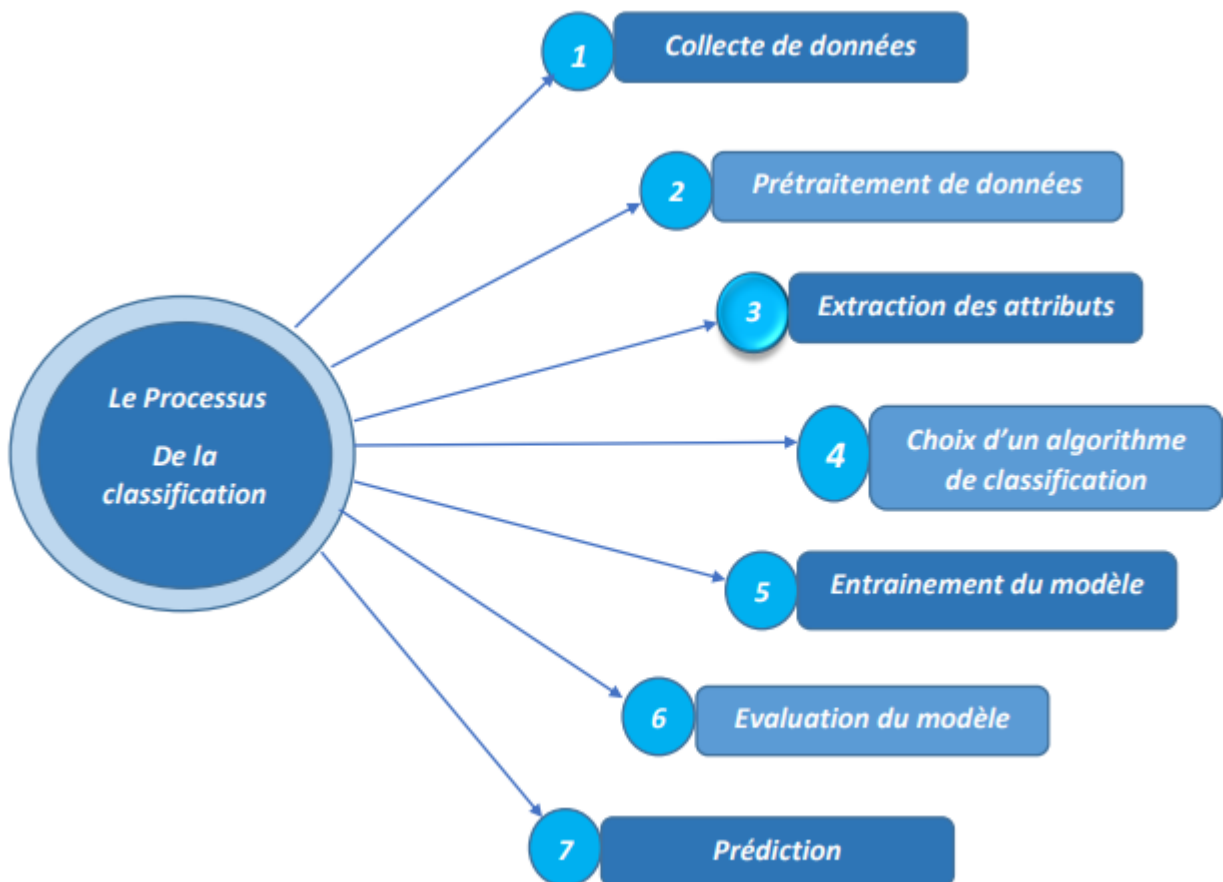


FIGURE 2.1 – Processus de la classification

2.3 Méthode de la classification :

Les méthodes de classification se divisent principalement en deux catégories : supervisée et non supervisée. Les méthodes supervisées nécessitent des données étiquetées pour entraîner le modèle, tandis que les méthodes non supervisées se basent sur des données non étiquetées pour découvrir des regroupements ou des structures naturelles dans les données.

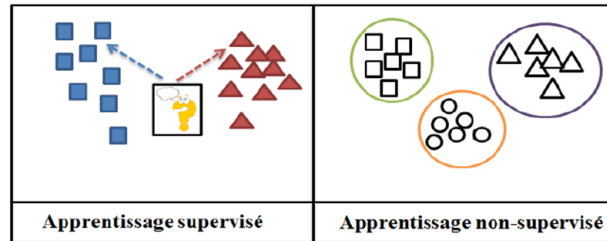


FIGURE 2.2 – Comparaison entre les méthodes de classification supervisée et non supervisée

2.3.1 Méthode non supervisée :

La méthode non supervisée, aussi appelée "clustering", regroupe les données sans avoir besoin d'étiquettes prédéfinies. Les algorithmes analysent les caractéristiques des objets pour les répartir en groupes ou clusters. L'objectif est de définir une fonction qui associe chaque élément à une ou plusieurs classes.

Parmi les méthodes non supervisées les plus utilisées, on trouve les techniques basées sur des centres mobiles, comme **K-means** et **Hiérarchiques**[46] [37].

2.3.1.1 Méthode de k-means :

La méthode des k-means, développé par McQueen en 1967, est l'une des méthodes les plus couramment utilisées pour la classification en apprentissage automatique. Cet algorithme repose sur la méthode des centroïdes, où chaque classe est représentée par un centre de gravité[5].

Le processus de l'algorithme de k-means se déroule en plusieurs étapes :

1. **Choix aléatoire du nombre de clusters** : Sélection aléatoire du nombre de clusters (k) à identifier dans les données.
2. **Initialisation des centroïdes** : Sélectionnez aléatoirement k points dans l'ensemble de données comme centres initiaux des clusters. Ces centres représentent les différentes classes à classifier.
3. **Attribution des points aux clusters** : Pour chaque point de données, calculez la distance entre le point et chaque centroïde. Assignez ensuite le point au cluster dont le centroïde est le plus proche.

4. **Mise à jour des centroïdes** : Recalculez la position de chaque centroïde en prenant la moyenne des points qui lui sont attribués, ce qui ajuste le centre de chaque cluster.
5. **Répétition des étapes** : Répétez les étapes 3 et 4 jusqu'à ce que les centroïdes ne changent plus ou que les points ne changent plus de cluster. Le processus continue jusqu'à ce qu'une configuration stable soit atteinte.
6. **Finalisation** : L'algorithme se termine lorsque les clusters ne changent plus, ce qui indique que les classes sont maintenant bien définies et stabilisées.

Bien que la méthode de k-means soit efficace du point de vue calcul, elle est sensible à l'initialisation des centroïdes. Une partition initiale non optimale peut conduire à des résultats sous-optimaux, et il existe un risque que certaines classes restent vides si leur centre de gravité ne correspond à aucun point de données. Malgré ces limitations, k-means reste un outil puissant pour la classification basée sur des caractéristiques visuelles et est largement utilisé dans divers domaines d'application.

2.3.1.2 Méthodes Hiérarchiques

Les méthodes de classification hiérarchique reposent sur une approche systématique pour regrouper les objets en classes de manière séquentielle, en partant de partitions très fines et en consolidant progressivement les classes. Cette approche utilise la notion de distance pour évaluer la similitude entre les objets ou les groupes d'objets. L'idée principale est d'agglomérer les objets ou les classes les plus proches les uns des autres à chaque étape, jusqu'à obtenir des classes plus globales.

Le processus des méthodes hiérarchiques se déroule en plusieurs étapes :

1. **Initialisation** : Chaque objet est considéré comme une classe individuelle. On commence donc avec un nombre de classes équivalent au nombre d'objets, où chaque objet forme sa propre classe.
2. **Fusion des classes** : À chaque étape, fusionnez les deux classes les plus proches en fonction d'un critère de dissimilarité. Ce critère est une mesure de distance ou de similarité entre les classes.
3. **Mise à jour des classes** : Après chaque fusion, recalculez les distances entre les nouvelles classes et les autres classes non fusionnées. Cela permet de mettre à jour les informations nécessaires pour les prochaines étapes de fusion.
4. **Itération** : Répétez le processus de fusion jusqu'à ce qu'un critère d'arrêt soit atteint, tel que le nombre désiré de classes ou une distance de dissimilarité maximale.
5. **Finalisation** : L'algorithme se termine lorsque toutes les classes ont été fusionnées en une seule classe ou lorsque le nombre de classes atteint le niveau souhaité.

Deux variantes principales des méthodes hiérarchiques sont :

- **Classification ascendante hiérarchique (CHA)** : Cette méthode commence avec chaque objet comme une classe distincte et fusionne progressivement les

classes en fonction de leur proximité, formant ainsi une hiérarchie de classes de plus en plus grandes.

- **Classification descendante hiérarchique** : À l'inverse, cette méthode débute avec un ensemble global et divise successivement cet ensemble en sous-groupes plus petits, créant une hiérarchie de classes de plus en plus fines jusqu'à atteindre des classes individuelles.

L'avantage principal de cette approche est sa simplicité d'utilisation et sa capacité à produire une hiérarchie complète des classes. Cependant, elle présente des défis, notamment la définition du critère de dissimilarité entre les classes, ce qui peut influencer fortement les résultats. De plus, les méthodes hiérarchiques peuvent être coûteuses en termes de temps de calcul, en particulier pour de grands ensembles de données.

2.3.2 Méthode supervisé :

C'est une méthode d'apprentissage où un algorithme est entraîné à partir de données étiquetées, c'est-à-dire des données pour lesquelles les catégories ou classes sont déjà connues. L'objectif est de permettre au modèle d'apprendre à associer des caractéristiques spécifiques à des étiquettes précises pour ensuite prédire correctement la classe de nouvelles données non étiquetées.

Pendant l'entraînement, l'algorithme analyse les exemples fournis, identifie les motifs et ajuste ses paramètres pour minimiser les erreurs de prédiction. Une fois le modèle entraîné, il peut classer de nouvelles données en fonction des connaissances acquises. Les méthodes supervisées incluent des algorithmes populaires comme les k-plus proches voisins (k-NN), les SVM et les réseaux de neurones[46] [37] .

2.3.2.1 Méthode des k plus proches voisins (kNN) :

L'algorithme KNN repose sur l'idée que des objets similaires sont généralement proches les uns des autres. Il utilise la notion de similitude (distance ou proximité) entre les points de données[10].

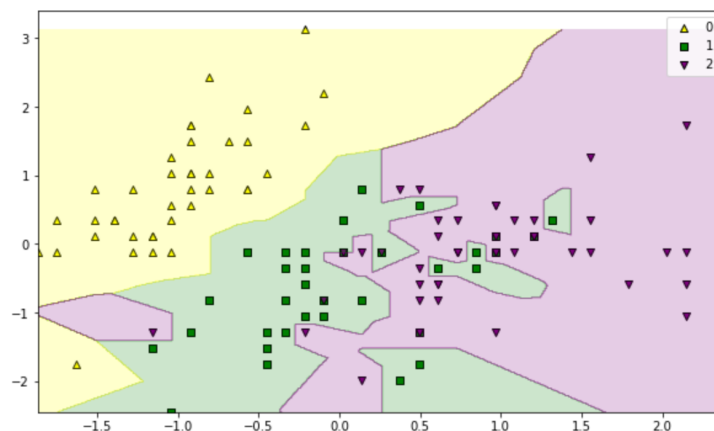


FIGURE 2.3 – Points similaires de l'algorithme KNN

1. Charger les données.
2. Initialiser k , le nombre de plus proches voisins choisi.
3. Pour chaque exemple dans les données :
 - 3.1. Calculer la distance entre notre requête et l'exemple actuel.
 - 3.2. Ajouter la distance et l'indice de l'exemple à une collection ordonnée.
4. Trier cette collection selon la distance.
5. Sélectionner les k premières entrées de la collection (les k plus proches voisins).
6. Obtenir les étiquettes des k entrées sélectionnée
7. retourner le mode des k étiquettes.

Cette méthode tire parti du fait que les points de données similaires sont souvent regroupés ensemble dans l'espace des caractéristiques, ce qui le rend utile pour la classification ou la régression.

2.3.2.2 Les machines à vecteurs de support (SVM) :

Les SVM sont des classificateurs puissants qui abordent les problèmes de classification en reformulant des problèmes non linéaires en problèmes d'optimisation quadratique, facilitant ainsi leur résolution. Cette méthode repose sur deux concepts clés : la marge maximale et la fonction noyau[20].

- **Marge maximale** : C'est la distance entre la frontière de séparation (hyperplan) et les points de données les plus proches (vecteurs de support) de chaque côté de cette frontière. L'objectif des SVM est de maximiser cette marge pour améliorer la généralisation du modèle.

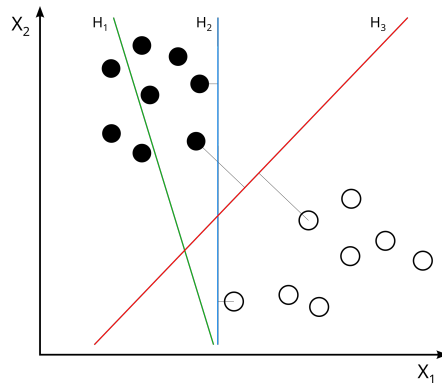


FIGURE 2.4 – Exemple illustrant différentes séparations de deux classes

L'illustration ci-dessus montre un exemple d'hyperplan séparateur pour un problème à deux dimensions ($N=2$). H1 ne sépare pas correctement les données ; H2 le fait, mais sans optimiser la séparation ; H3, en revanche, sépare les données avec une marge maximale.

- **La fonction noyau** : C'est une méthode permettant de transformer des données non linéaires en un espace de dimension supérieure, où elles peuvent devenir linéairement séparables. Au lieu de calculer directement dans cet espace de haute dimension, la fonction noyau permet d'effectuer les calculs nécessaires dans l'espace d'origine, simplifiant ainsi considérablement le processus. Cette technique est connue sous le nom de "kernel trick".

Étapes de la méthode :

1. Charger les données.
2. Déterminer la séparabilité :
 - Pour un problème linéairement séparable : Identifier l'hyperplan qui maximise la marge entre les deux classes en utilisant les vecteurs de support pour définir cet hyperplan.
 - Pour un problème non linéairement séparable : Appliquer la fonction noyau pour projeter les données dans un espace de dimension supérieure afin de permettre une séparation linéaire, puis trouver un hyperplan dans cet espace transformé qui sépare les classes de manière optimale.

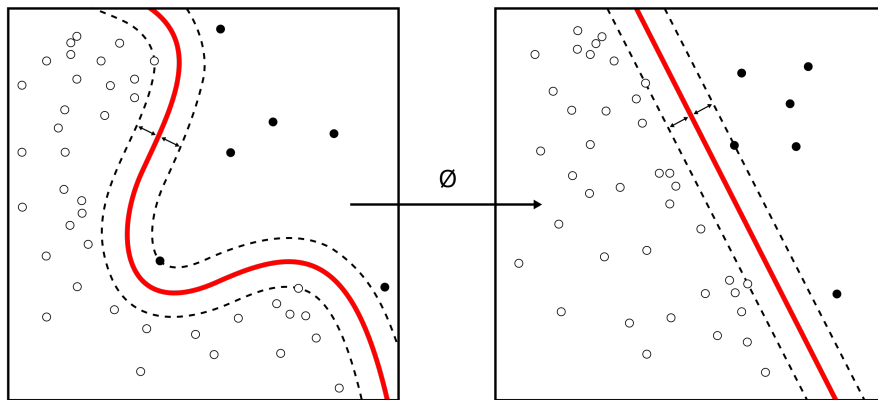


FIGURE 2.5 – Exemple de transformation à l'aide du kernel trick

3. Entraîner le modèle : Après plusieurs phases d'entraînement, le SVM apprend à placer la frontière pour de nouvelles données.
4. Classifier les nouvelles données.

Les SVM ont été initialement conçus pour des problèmes de classification binaire. Pour gérer des problèmes multi-classes, des techniques comme "one-vs-all" ou "one-vs-one" sont utilisées. Par exemple, avec l'approche "one-vs-all", un SVM est entraîné pour séparer chaque classe des autres, puis la classe ayant le score le plus élevé est sélectionnée pour la classification finale.

2.3.2.3 Les Réseaux de Neurones :

Les réseaux de neurones sont des modèles inspirés du cerveau humain, constitués de neurones artificiels organisés en couches. Ils sont largement utilisés en raison de leur capacité à traiter efficacement de grandes quantités de données. Leur précision élevée, leur capacité de traitement parallèle en temps réel, leur robustesse et leur rapidité de classification en font des outils puissants. Les réseaux de neurones peuvent être utilisés pour la classification supervisée et non supervisée, avec le perceptron multicouche étant le modèle le plus couramment utilisé. L'apprentissage, souvent basé sur la rétro-propagation du gradient, ajuste les poids du réseau pour minimiser les erreurs. En résumé, les réseaux de neurones sont des outils puissants pour la classification d'images, bien que leur conception et leur paramétrage nécessitent une expertise considérable pour obtenir des résultats optimaux[27].

2.4 La Classification des images par CNN :

La classification des images reposait initialement sur des méthodes traditionnelles où les experts définissaient manuellement les caractéristiques à analyser. Bien que fonctionnelles, ces techniques montrent des limites en termes de précision et de traitement de données complexes, peinant à capturer des motifs subtils, surtout dans des domaines comme l'imagerie médicale.

Les réseaux de neurones convolutifs (CNN) ont transformé ce processus. Contrairement aux méthodes traditionnelles, les CNN apprennent automatiquement des caractéristiques complexes directement à partir des images, rendant la classification plus précise et adaptable. Ils sont particulièrement efficaces pour des tâches telles que la détection de tumeurs cérébrales sur des images IRM.

Un exemple comparatif peut illustrer cette différence(voir figure(2.6)) : dans la classification traditionnelle, l'extraction manuelle des caractéristiques est suivie d'une réduction de dimensionnalité et de l'utilisation d'un classificateur traditionnel, tel que le kNN, pour réaliser une prédiction. En revanche, les CNN automatisent ces étapes, utilisant des couches de convolution pour apprendre les caractéristiques, des couches de pooling pour réduire la dimensionnalité, et des couches entièrement connectées suivies d'une prédiction via Softmax pour obtenir une classification plus précise.

Dans le domaine médical, cette automatisation a considérablement amélioré la précision des diagnostics, permettant une meilleure prise en charge des patients. De nombreuses études récentes illustrent l'efficacité des CNN dans la classification des tumeurs cérébrales, mettant en évidence leurs avantages par rapport aux méthodes traditionnelles.

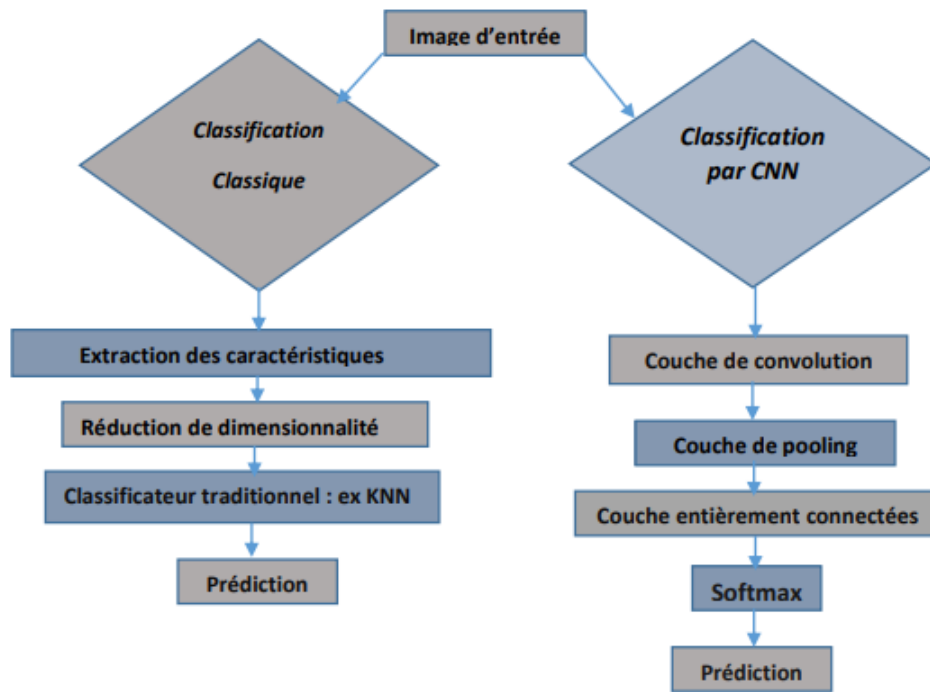


FIGURE 2.6 – Un exemple comparatif peut illustrer la différence la classification classique et par CNN.

Voici quelques contributions notables dans ce domaine :

Nayak et al.[42] (2022) ont utilisé le transfert d'apprentissage avec EfficientNet pour distinguer les images de cerveau en quatre catégories : normal, méningiome (MEN), gliome et tumeur pituitaire (PT). Ils ont atteint une précision et un F1-score de 98,75 %.

Ahammed et al. [35] (2019) ont employé VGG19 pour classifier les tumeurs cérébrales en quatre grades. Ils ont obtenu une précision de 94,71 % , une sensibilité de 92,72%, une spécificité de 98,13 % et un F1-score de 93,71 % .

Xiao et al. [14] ont appliqué le transfert d'apprentissage avec ResNet50 pour classifier MEN, gliome et PT, utilisant des validations croisées de 3, 5 et 10 plis, avec une précision atteignant 98,02 % .

Rizwan et al.[47] (2022) ont développé un CNN personnalisé pour classifier MEN, gliome et PT, obtenant une précision remarquable de 99,8 % .

Laraimi et al. [15] (2021) ont appliqué le transfert d'apprentissage avec AlexNet, VGG16 et GoogLeNet pour classifier les images de tumeurs cérébrales en (MEN), gliome et (PT). Avec AlexNet, ils ont obtenu une aire sous la courbe (AUC) de 0,976 et une

précision de 94,4% . En utilisant VGG16, l'AUC a atteint 0,981 avec une précision de 100% . Pour GoogLeNet, l'AUC était de 0,986 % et la précision de 98,5 %.

Noreen et al.[43] (2020) ont utilisé le transfert d'apprentissage avec Inception-v3 et DenseNet201 pour classifier les images de tumeurs cérébrales (MEN), gliome et PT. Avec Inception-v3, ils ont obtenu une précision de 99,34% . En utilisant DenseNet201, la précision a atteint 99,51 %.

Bulla et al. [17] (2020) ont utilisé le transfert d'apprentissage avec InceptionV3 pour classifier les images de tumeurs cérébrales en (MEN), gliome et (PT). En appliquant différentes méthodes de validation, ils ont obtenu une précision (PRE) de 97,57 % , une sensibilité (RE) de 99,47 % , un F1-score de 98,40 % , et une aire sous la courbe (AUC) de 0,995. La précision globale du modèle était de 99,82 % .

2.5 Conclusion :

En résumé, la classification d'image est un processus complexe mais essentiel, utilisant diverses méthodes d'apprentissage automatique pour analyser et catégoriser les données visuelles. Que ce soit par des méthodes supervisées ou non supervisées, chaque technique présente ses propres avantages et défis. Les avancées récentes, notamment avec les réseaux de neurones convolutifs (CNN), ont significativement amélioré la précision et l'efficacité des diagnostics, particulièrement dans le domaine médical. La compréhension et l'amélioration continue de ces méthodes sont cruciales pour le développement futur de la vision par ordinateur et d'autres applications liées.

Chapitre 3

Les Réseaux de Neurones Convolutifs :

3.1 Introduction :

Les réseaux neuronaux et l'apprentissage en profondeur donnent de nos jours de bonnes solutions à plusieurs problèmes de reconnaissance et de classification d'image, raison pour laquelle ils sont devenus sujets importants en informatique et dans l'industrie économique. Inspirés de la structure du cerveau de l'être humain, les réseaux de neurones artificiels font partie des moyens qui permettent de rendre les ordinateurs plus humains et aider ainsi les machines à raisonner comme des êtres humains. Le cerveau humain apprend d'une manière essentielle par l'expérience. Il s'agit donc d'une preuve naturelle que certains problèmes peuvent dépasser la portée des ordinateurs.

3.2 Deep learning :

3.2.1 Qu'est ce que c'est le deep learning :

Le deep learning, ou apprentissage profond, est un sous-domaine de l'intelligence artificielle et un sous-ensemble de l'apprentissage automatique (machine learning). Contrairement aux approches traditionnelles, où les caractéristiques des données sont extraites manuellement, le deep learning permet à la machine d'apprendre de manière autonome en utilisant des réseaux de neurones artificiels. Ces réseaux, conçus pour imiter le fonctionnement du cerveau humain, détectent automatiquement les caractéristiques des données et s'améliorent en analysant de vastes ensembles de données. Les progrès récents en puissance de calcul et en analyse du big data ont permis de créer des réseaux de neurones plus vastes et sophistiqués, capables de traiter des situations complexes plus rapidement que les humains. Le deep learning a permis des avancées significatives dans des domaines tels que la classification d'images, la traduction linguistique et la reconnaissance vocale." [23] [37][41].

3.2.2 Historique de deep learning

Le Deep Learning trouve ses origines en 1943 avec le modèle précurseur de Walter Pitts et Warren McCulloch, inspiré des réseaux neuronaux du cerveau humain. Cependant, le concept moderne se développe réellement dans les années 1980, lorsque Rina Dechter popularise le terme et que Yann LeCun utilise un réseau de neurones profonds pour reconnaître les codes postaux manuscrits en 1989.

Le développement des réseaux de neurones convolutifs (CNN), crucial pour le Deep Learning, est avancé par LeCun, Kunihiko Fukushima, et Georey Hinton. En 2007,

Fei-Fei Li lance ImageNet, une base de données d'images étiquetées, facilitant l'entraînement des modèles de Deep Learning.

L'année 2009 marque une avancée majeure avec l'utilisation des GPU par Nvidia, accélérant considérablement l'entraînement des modèles. En 2012, Andrew Ng et son équipe créent une architecture de réseau neuronal profonde, entraînée avec 10 millions de vidéos YouTube, permettant de reconnaître des images de chats et signalant une étape clé dans l'évolution du Deep Learning.

L'année 2014 voit l'introduction des Réseaux Neuronaux Antagonistes Génératifs (GAN) par Ian Goodfellow, où deux réseaux neuronaux rivalisent pour générer des images réalistes. Cette même année, les systèmes de reconnaissance d'images basés sur le Deep Learning atteignent une précision surpassant celle des humains, grâce à l'amélioration continue des algorithmes et à l'augmentation des capacités de calcul[29].

En 2016, de nouveaux produits d'apprentissage automatique puissants ont été lancés. Par exemple, Cray Inc a proposé des solutions avancées en matériel pour l'apprentissage profond, et Microsoft a utilisé des superordinateurs avec des processeurs graphiques pour traiter des données rapidement.

D'autre part Le potentiel du marché se révèle énorme. En 2018, McKinsey estimait la création de valeur annuelle entre 3 500 et 5 800 milliards de dollars. L'entreprise O'Reilly indique que l'apprentissage non supervisé aurait augmenté de 172% en 2019 et représentait 22% des applications d'intelligence artificielle[51].

3.3 Les Réseaux de Neurones :

3.3.1 Perceptron :

Le perceptron simple, un modèle de neurone artificiel créé par Frank Rosenblatt, est une brique fondamentale de l'apprentissage automatique. Il fonctionne en calculant une somme pondérée de ses entrées, puis en appliquant une fonction d'activation pour déterminer une sortie. Cette sortie correspond à une classification binaire : oui ou non, 1 ou 0. Bien que simple, le perceptron a ses limites et ne peut résoudre tous les problèmes. Malgré cela, il a ouvert la voie à des modèles plus complexes et performants comme les réseaux de neurones multicouches, qui sont aujourd'hui à la base de nombreuses applications d'intelligence artificielle.

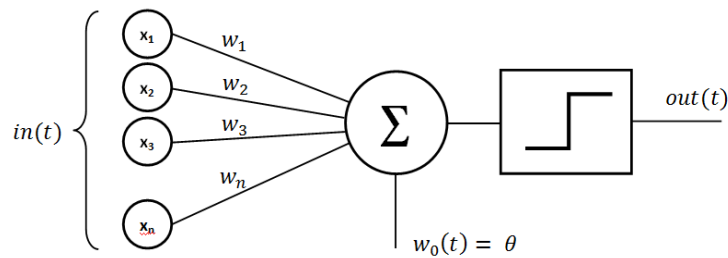


FIGURE 3.1 – Schéma de perceptron simple

ou :

Les valeurs (x_i) sont les données d'entrée (les variables que l'on souhaite prendre en compte).

$W_0(t)$: poids synaptique

$Out(t)$: sont les données de sorties

$In(t)$: entrées

3.3.2 Perceptron multi-couches :

Le perceptron multicouche (MLP) est un modèle plus complexe qui utilise plusieurs couches de neurones pour apprendre à classer des données. Les couches cachées permettent de créer des représentations plus abstraites des données, améliorant ainsi la capacité du modèle à apprendre. L'apprentissage d'un MLP se fait généralement par une méthode d'optimisation comme la descente du gradient, qui ajuste les poids en fonction de la fonction d'erreur entre la sortie prédite et la sortie attendue. Les MLP sont couramment utilisés pour des tâches de classification multiclass et de régression .

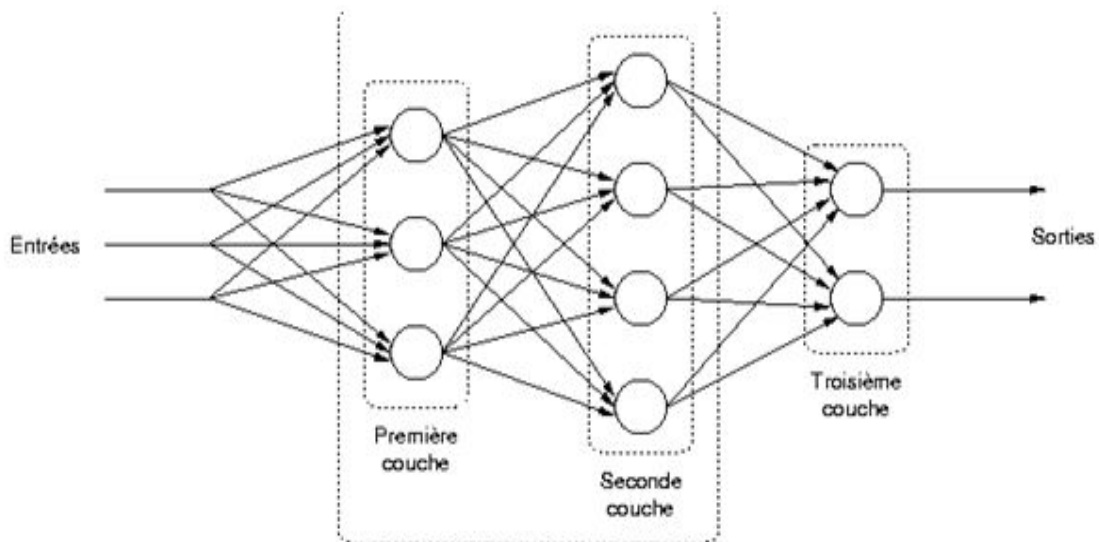


FIGURE 3.2 – Schéma simple d'un MLP

3.3.3 L'architecture d'un réseau de neurones :

On distingue deux types d'architectures de réseaux de neurones : les réseaux de neurones « non bouclés » et les réseaux de neurones « bouclés »[25].

- **les réseaux de neurones bouclés** : les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant des boucles qui ramènent aux entrées d'un neurone la valeur de sa sortie, éventuellement par l'intermédiaire d'autres neurones. Pour qu'un tel système soit causal, il faut évidemment qu'à toute boucle soit associé un « retard » : un réseau de neurones bouclé est donc un système dynamique, régi par des équations différentielles.
- **les réseaux de neurones non bouclés** :

Un réseau de neurones non bouclé réalise une ou plusieurs fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun de ses neurones. Il est représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière. Un « Perceptron multicouche » est un réseau de neurones non bouclé qui a une structure particulière, très fréquemment utilisée : il comprend des entrées, une couche de neurones intermédiaires, dits neurones cachés, et des neurones de sortie. Les neurones de la couche cachée ne sont pas connectés entre eux.

Les réseaux de neurones non bouclés sont statiques : si les entrées sont indépendantes du temps, les sorties le sont également. Ils sont utilisés principalement pour effectuer des tâches d'approximation de fonction non linéaire, de modélisation statique de processus non linéaires ou de classification.

3.4 Les Réseaux de Neurones Convolutifs

: Les réseaux de neurones convolutifs (CNN, ou ConvNet) sont l'un des algorithmes les plus populaires dans le domaine de l'apprentissage en profondeur. Comme les autres réseaux de neurones, un CNN est composé d'une couche d'entrée, d'une couche de sortie et de nombreuses couches cachées. Ils utilisent des opérations de convolution pour extraire automatiquement des caractéristiques hiérarchiques à partir d'images. Ces caractéristiques, de plus en plus complexes au fur et à mesure de l'approfondissement du réseau, permettent de réaliser des tâches de classification, de détection d'objets et de segmentation d'images avec une précision remarquable.

3.4.1 L'architecture d'un réseau de neurones convolutifs :

Un réseau de neurones convolutif (CNN) comporte plusieurs étapes clés : convolution, ReLU et éventuellement regroupement (pooling). La fonction de correction (ReLU) doit être appliquée après chaque étape de convolution pour obtenir une réponse non-linéaire, tandis que le regroupement (pooling) est facultatif. Après avoir traversé ces étapes, l'image passe à la phase de classification, où tous les pixels sont envoyés dans un réseau de neurones multicoluches. Grâce à la condensation des parties importantes

de l'image, cette phase de classification est plus performante qu'un réseau de neurones artificiels sans convolution.

Un CNN est une succession de couches de convolution, regroupement, correction et entièrement connectées. Chaque image d'entrée est filtrée, réduite et corrigée plusieurs fois pour former un vecteur. Dans les problèmes de classification, ce vecteur contient les probabilités d'appartenance aux classes. Tous les CNN commencent par une couche de convolution et se terminent par une couche entièrement connectées, avec des couches intermédiaires qui doivent avoir des sorties compatibles avec les entrées des couches suivantes. Par exemple, une couche entièrement connectées ne peut pas précéder une couche de regroupement.

En général, un CNN empile plusieurs couches de convolution et de correction ReLU, ajoute une couche de regroupement (facultative), et répète ce schéma plusieurs fois avant d'ajouter des couches entièrement connectées. La profondeur d'un réseau de neurones augmente avec le nombre de couches, ce qui relève du deep learning.

Les premières couches de convolution apprennent des features simples comme les contours et les coins. Plus on monte dans les couches, plus les features apprises deviennent complexes, intégrant des éléments des couches précédentes. Par exemple, dans la reconnaissance de chats, des features plus complexes pourraient correspondre à des oreilles, un nez ou des yeux, constituées de contours et de coins[38].

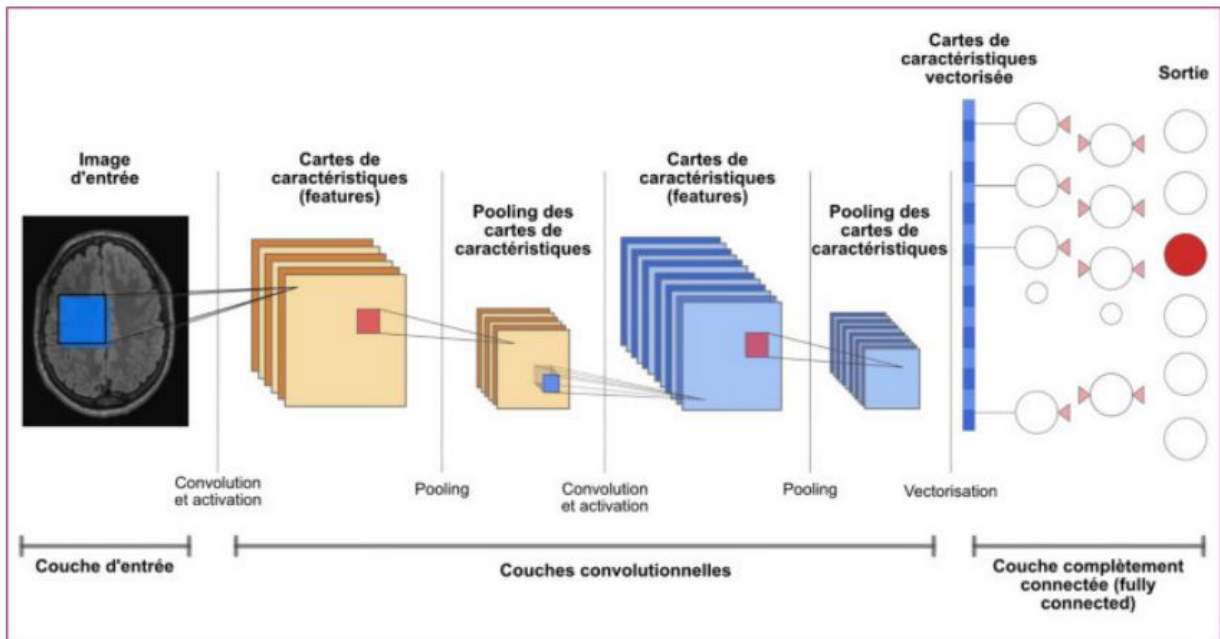


FIGURE 3.3 – Schéma d'un CNN

3.4.1.1 Couche de convolution :

La couche de convolution est une composante fondamentale des réseaux de neurones convolutifs, étant souvent leur première couche. Elle est constituée de filtres, qui sont des

matrices de valeurs appelées cartes de caractéristiques (feature maps). Chaque filtre est appliqué à l'image en entrée pour produire une carte de caractéristiques correspondante. Par exemple, si l'image est de taille 5x5 et que le filtre est de taille 3x3, la carte de caractéristiques sera de taille 3x3. Ainsi, la couche de convolution prend en entrée plusieurs images et effectue la convolution de chacune d'entre elles avec chaque filtre. Les filtres sont spécialement conçus pour détecter des caractéristiques spécifiques dans les images, correspondant souvent à des motifs ou des formes d'intérêt[52].

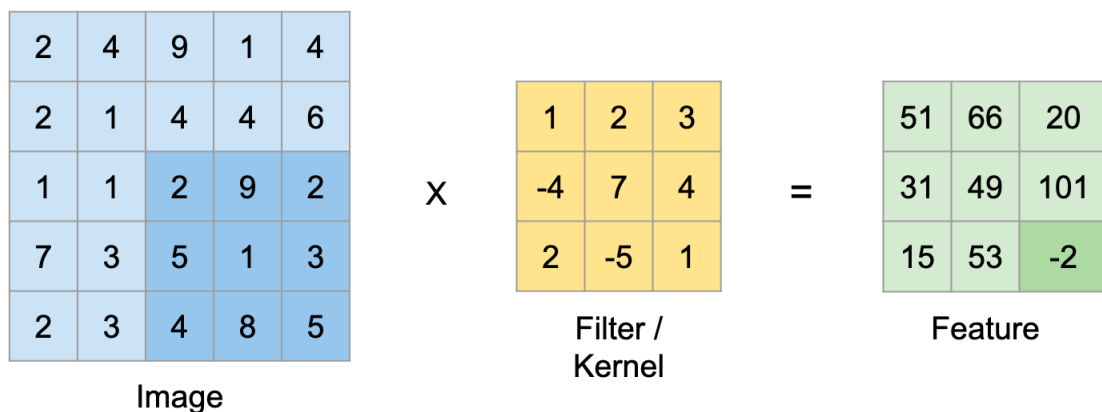


FIGURE 3.4 – Convolution

3.4.1.2 Couche de regroupement(pooling) :

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. Une couche de pooling, agit comme une couche de réduction. Elle divise l'image en blocs et ne garde par exemple le maximum de chaque bloc. Cela permet de réduire la dimension de l'image tout en conservant les caractéristiques les plus importantes. On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites[52].

Il existe différents types de pooling :

- (1) **Pooling moyen** qui prend la moyenne de tous les pixels de la sélection(voir Figure 3.5.(b)).
- (2) **Pooling maximal** qui prend le pixel qui a la valeur maximale entre tous les pixels de la sélection(Voir Figure 3.5.(a)).

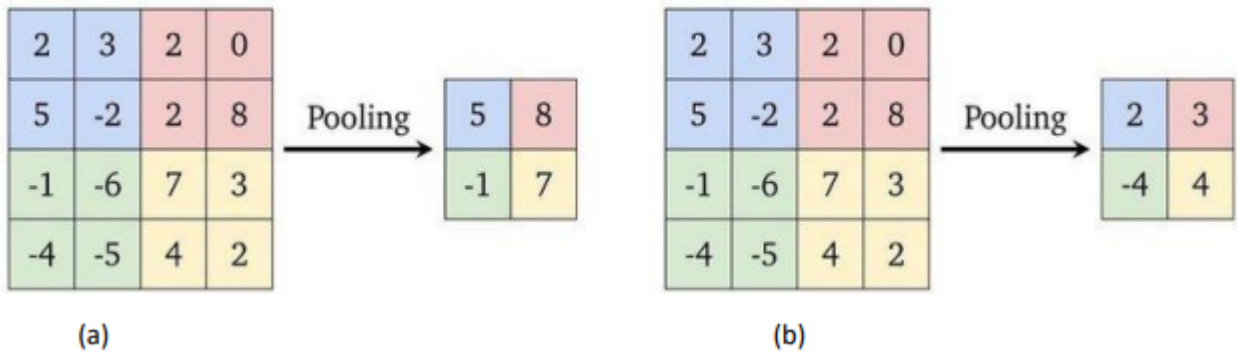


FIGURE 3.5 – (a)pooling maximal,(b) pooling moyen

3.4.1.3 Couche complètement connectées (fully-connected) :

La couche complètement connectées est une composante essentielle des CNN (Convolutional Neural Networks). Elle suit généralement plusieurs couches de convolution et de pooling. Sa fonction principale est de combiner les caractéristiques extraites par les couches précédentes pour classifier l'image en entrée. Cette couche produit un vecteur de taille N, où N représente le nombre de classes dans le problème de classification d'images donné. Chaque élément de ce vecteur indique la probabilité que l'image appartienne à une classe spécifique[52].

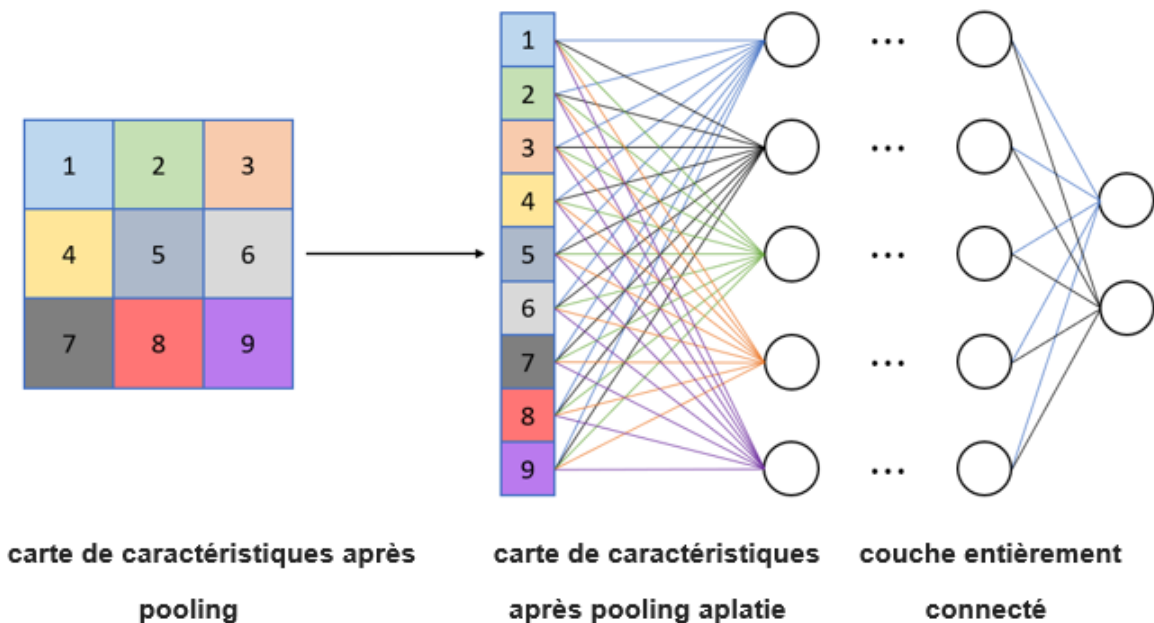


FIGURE 3.6 – Couche complètement connectées

3.4.2 Fonction d'activation :

Dans un réseau de neurones artificiels, les fonctions d'activation jouent un rôle crucial dans l'apprentissage et la modélisation de relations complexes entre les entrées et les sorties. Une fonction d'activation transforme la sortie linéaire d'un neurone (c'est-à-dire la somme pondérée des entrées) en une forme non linéaire, permettant ainsi au réseau de capturer des structures plus complexes dans les données.

L'objectif principal des fonctions d'activation est d'introduire des non-linéarités dans le modèle. Sans ces non-linéarités, un réseau de neurones, même avec plusieurs couches, se comporterait comme un modèle linéaire simple. Les non-linéarités permettent donc au réseau de neurones de modéliser des fonctions plus sophistiquées et de résoudre des problèmes plus complexes, tels que la reconnaissance d'images, la traduction de langue, et bien d'autres tâches en apprentissage automatique.

Il existe plusieurs types de fonctions d'activation, chacune avec ses propres caractéristiques, avantages, et inconvénients. Les fonctions les plus couramment utilisées incluent la sigmoïde, la tangente hyperbolique (tanh), la fonction ReLU (Rectified Linear Unit), et la fonction softmax. Le tableau suivant présente un comparatif de ces différentes fonctions d'activation pour une entrée x , en mettant en évidence leurs définitions, leurs dérivées, et les intervalles dans lesquels elles opèrent.

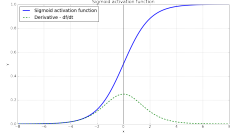
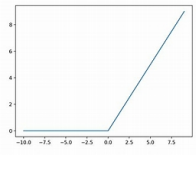
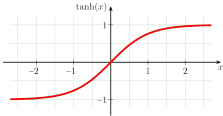
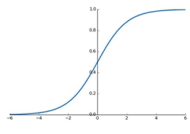
Fonction	Définition	Graphe	Dérivée	Intervalle
Sigmoïde	$\sigma(x) = \frac{1}{1+e^{-x}}$		$\sigma'(x) = \sigma(x)(1 - \sigma(x))$	$[0, 1]$
ReLU	$f(x) = \max(0, x)$		$f'(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases}$	$[0, \infty)$
Tanh	$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$		$\tanh'(x) = 1 - \tanh^2(x)$	$[-1, 1]$
Softmax	$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}}$		Varie	$[0, 1]$

TABLE 3.1 – Tableau comparatif des fonctions d'activation

3.4.3 L'optimisation et l'hyperparamètres d'un CNN :

L'optimisation des réseaux de neurones convolutifs (CNN) repose sur le réglage attentif de plusieurs hyperparamètres et l'emploi de techniques spécifiques pour accroître leur efficacité et robustesse. Voici une analyse détaillée de ces éléments essentiels[31] :

a. Taux d'apprentissage (Learning Rate) Le taux d'apprentissage détermine la rapidité avec laquelle le réseau ajuste ses poids pour minimiser l'erreur. Un taux d'apprentissage trop élevé peut provoquer une instabilité et empêcher la convergence du modèle, tandis qu'un taux trop bas peut ralentir l'apprentissage et piéger le modèle dans des minima locaux.

b. Nombre et Taille des Filtres Le nombre et la taille des filtres dans chaque couche convolutive influencent la capacité du modèle à détecter des caractéristiques à divers niveaux de granularité. Des filtres plus nombreux et de plus grande taille permettent de capturer des caractéristiques plus complexes mais augmentent également le coût computationnel.

c. Choix de l'Optimiseur L'optimiseur affecte directement la vitesse et la qualité de la convergence du modèle. Parmi les options courantes :

- **SGD (Stochastic Gradient Descent)** : Cet optimiseur met à jour les paramètres du modèle en utilisant un sous-ensemble aléatoire des données à chaque itération. Son rôle est de permettre des ajustements rapides et efficaces, ce qui est particulièrement utile pour les grands ensembles de données..
- **Adam (Adaptive Moment Estimation)** : Adam ajuste automatiquement le taux d'apprentissage pour chaque paramètre du modèle en fonction des gradients passés. Son rôle est de combiner la rapidité de convergence avec une meilleure stabilité, ce qui en fait un choix populaire pour de nombreux types de modèles.

d. Techniques d'amélioration de la Robustesse

- **Dropout** Le dropout est une technique de régularisation qui aide à prévenir le surajustement en désactivant aléatoirement des neurones pendant l'entraînement. Cela force le réseau à ne pas dépendre excessivement de caractéristiques spécifiques, améliorant ainsi sa capacité à généraliser.
- **Normalisation par Lots (Batch Normalization)** La normalisation par lots ajuste et met à l'échelle les activations pour stabiliser l'apprentissage. Cela aide à réduire les problèmes liés à la disparition ou à l'explosion des gradients en maintenant les activations dans une plage stable, ce qui accélère l'entraînement et améliore la performance globale du modèle.

3.4.4 Modèles pré-entraînés :

3.4.4.1 Un CNN pré-entraînés :

Un CNN pré-entraîné est un modèle de réseau de neurones convolutif qui a été entraîné sur un grand jeu de données pour une tâche spécifique, puis utilisé pour une autre tâche similaire. Ce modèle a déjà appris à détecter des caractéristiques importantes dans les images et peut être utilisé comme base pour des applications spécifiques, comme la reconnaissance d'images, la détection d'objets, la segmentation sémantique, et même dans des domaines tels que la reconnaissance de la parole et la traduction automatique.

3.4.4.2 Transfert learning (apprentissage par transfert) :

Le transfert learning permet d'exploiter la puissance des CNN pré-entraînés, évitant ainsi la complexité de la création de nouveaux réseaux. Cette technique utilise des modèles pré-entraînés pour des problèmes de classification généraux et les adapte à des problèmes particuliers avec une quantité raisonnable de données annotées. ces méthodes[34] :

- **a. Extraction Automatique de Caractéristiques** L'extraction automatique de caractéristiques utilise uniquement la partie convolutive d'un réseau pré-entraîné comme extracteur de caractéristiques. Par exemple, le réseau VGG-16 est tronqué pour ne garder que les couches convolutives, gelées pour exprimer l'absence d'entraînement supplémentaire. Le processus consiste à utiliser le réseau convolutif tronqué pour prendre en entrée une image et produire un vecteur de caractéristiques (code CNN), qui alimente ensuite un nouveau classifieur. Cette méthode présente plusieurs avantages, notamment la transformation des images en vecteurs de petites dimensions, facilitant ainsi la classification, et la liberté d'utiliser différents classifieurs selon les besoins. De plus, elle permet une exploitation possible sur CPU, évitant l'investissement dans des GPU.
- **b. Fine tuning** : Le fine tuning consiste à utiliser un modèle pré-entraîné comme point de départ et à le ré-entraîner sur un nouveau problème. Ce processus permet de tirer parti de l'architecture optimisée et des capacités d'extraction de caractéristiques d'un modèle existant. Il inclut l'adaptation de la dernière couche du modèle pour correspondre au nouveau problème de classification (par exemple, remplacer la couche finale de VGG-16 pour passer de 1000 à 10 catégories), et éventuellement l'ajout d'une couche intermédiaire ou la modification de la taille des couches existantes. Les couches initiales sont gelées, et les couches finales sont ré-entraînées avec un faible taux d'apprentissage (de l'ordre de 10^{-3}). Les avantages de cette méthode incluent l'utilisation d'une architecture éprouvée et optimisée, ainsi que l'amélioration de la performance finale par rapport à l'extraction automatique de caractéristiques seule.

3.4.5 Les avantages et les inconvénients d'un CNN :

Avantages des CNN	Inconvénients des CNN
Apprentissage à partir des données brutes : Les CNN peuvent apprendre directement des pixels d'image sans nécessiter d'ingénierie manuelle des caractéristiques.	Exigences computationnelles élevées : Les CNN sont coûteux à entraîner et à exécuter en raison du grand nombre de paramètres à mettre à jour.
Découverte automatique des caractéristiques : Les CNN peuvent découvrir automatiquement les caractéristiques saillantes des images comme les bords, les formes, les couleurs, etc.	Besoin de grandes quantités de données : Les CNN nécessitent de grands ensembles de données étiquetées pour s'entraîner efficacement, ce qui peut être coûteux et long.
Exploitation de la structure spatiale : Les CNN préservent la connectivité locale et le contexte des pixels, permettant des représentations abstraites et hiérarchiques.	Surajustement : Les CNN peuvent mémoriser le bruit et les détails des données d'entraînement, échouant à généraliser sur de nouvelles données.
Réduction de la dimensionnalité : Les CNN réduisent la complexité des données d'entrée, rendant la formation et l'inférence plus rapides et plus efficaces.	Difficulté d'interprétation : Les CNN sont souvent considérés comme des boîtes noires, rendant l'interprétation et l'explication difficiles.
Bonne généralisation : Les CNN peuvent capturer la variabilité des images et généraliser aux données nouvelles et invisibles.	Techniques de régularisation nécessaires : Pour éviter le surajustement, des techniques comme l'abandon, la normalisation par lots et l'augmentation des données doivent être appliquées, augmentant la complexité[3].

TABLE 3.2 – Avantages et inconvénients des réseaux neuronaux convolutifs (CNN)

3.4.6 Quelques réseaux de neurones convolutifs :

L'architecture des CNN a connu une évolution rapide depuis leurs premières incarnations. La figure 3.7 présente un aperçu chronologique de quelques modèles CNN emblématiques qui ont marqué cette évolution. De LeNet-5, un pionnier dans le domaine, à des modèles plus récents comme EfficientNet et DenseNet, chaque itération a introduit des innovations significatives en termes de profondeur, de complexité et de

performance. Ces avancées ont permis aux CNN de s'imposer comme l'état de l'art dans de nombreuses tâches de vision par ordinateur.

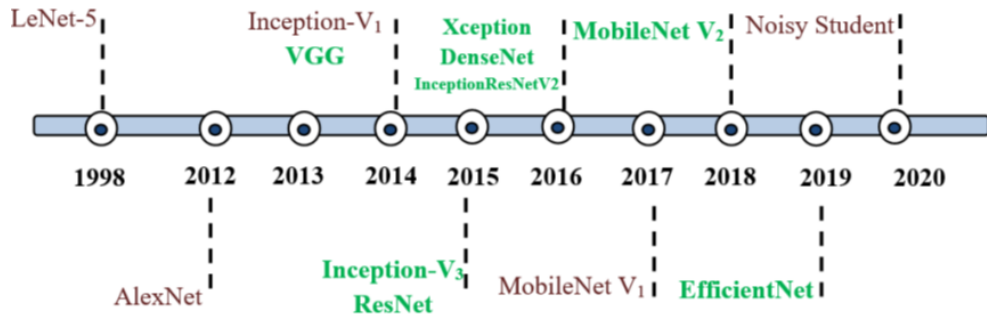


FIGURE 3.7 – Historique des CNN

3.4.6.1 Densenet :

DenseNet est une architecture de réseau neuronal convolusionnel (CNN) introduite en 2016 par Gao Huang, Zhuang Liu, Laurens van der Maaten et Kilian Q. Weinberger. DenseNet, qui signifie "Densely Connected Convolutional Networks" (réseaux convolusionnels densément connectés), se distingue par sa connexion unique entre les couches : chaque couche est connectée à toutes les couches précédentes (voir Figure 3.8). Cette connectivité permet une meilleure réutilisation des caractéristiques et améliore l'efficacité du réseau.[21]

Deux éléments clés distinguent DenseNet des autres architectures CNN. Premièrement, l'utilisation de blocs denses où chaque couche reçoit comme entrée les cartes de caractéristiques de toutes les couches précédentes. Deuxièmement, les couches de goulot d'étranglement, qui réduisent le nombre de paramètres tout en maintenant la capacité du réseau à apprendre efficacement[21].

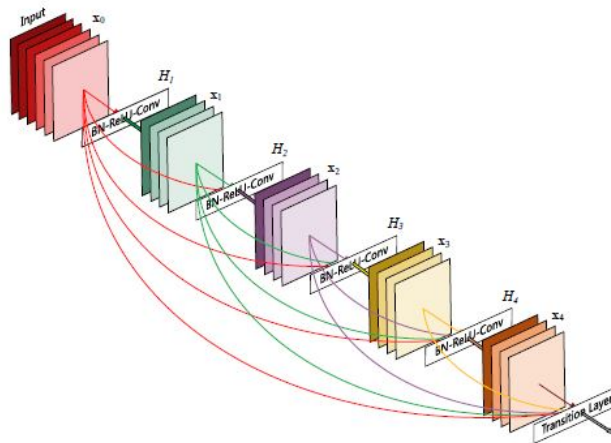


FIGURE 3.8 – La connectivité entre les couches

- **Les composants :** Les composants de DenseNet incluent :

Connectivité : Dans chaque couche, les cartes de caractéristiques de toutes les couches précédentes ne sont pas additionnées, mais concaténées et utilisées comme entrées. Par conséquent, les DenseNets nécessitent moins de paramètres qu'un CNN traditionnel équivalent, ce qui permet la réutilisation des fonctionnalités car les cartes de fonctionnalités redondantes sont supprimées. Ainsi, la l ème couche reçoit les cartes de caractéristiques de toutes les couches précédentes, x_0, \dots, x_{l-1} , en entrée :

$$x_l = H_l[x_0, x_1, \dots, x_{l-1}]$$

où $[x_0, x_1, \dots, x_{l-1}]$ est la concaténation des cartes de caractéristiques, c'est-à-dire la sortie produite dans toutes les couches précédant l ($0, \dots, l-1$). Les multiples entrées de H_l sont concaténées en un seul tenseur pour faciliter la mise en œuvre. (H_l est la fonction appliquée aux cartes concaténées pour générer la sortie de la première couche.)

Blocs denses : (Dense block en anglais) Ces blocs permettent de construire des réseaux profonds tout en restant efficaces sur le plan computationnel. Dans chaque bloc dense, la taille des cartes de caractéristiques reste constante, et chaque couche applique une normalisation par lots, une activation ReLU et une convolution 3×3 . Entre les blocs denses, des couches de transition, composées d'une convolution 1×1 suivie d'un pooling moyen 2×2 , réduisent la dimensionnalité spatiale des cartes tout en augmentant le champ réceptif des neurones (voir Figure 3.9).

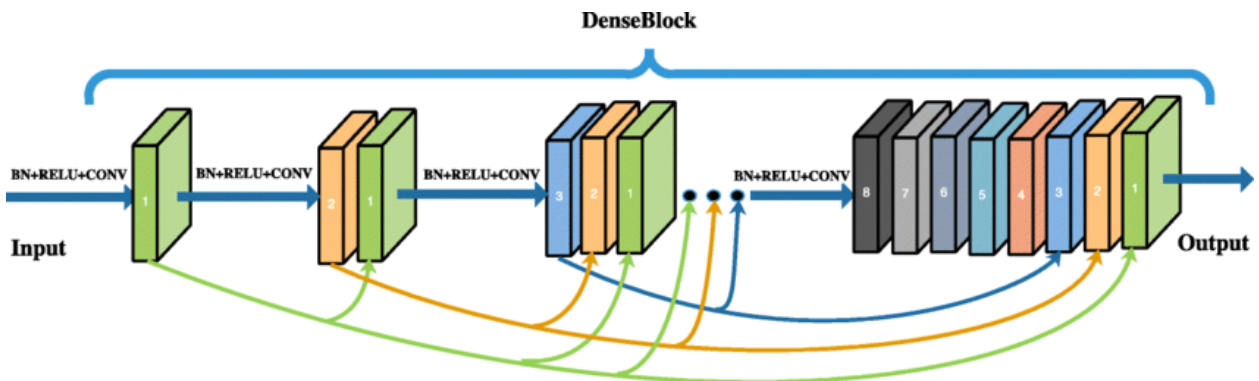


FIGURE 3.9 – Blocs denses

Taux de croissance :

Le taux de croissance, noté k , définit le nombre de nouvelles caractéristiques ajoutées à chaque couche. Contrairement aux autres architectures, DenseNet permet d'utiliser des couches étroites, offrant ainsi une flexibilité dans la conception du réseau. La relation entre les paramètres est décrite par la formule :

$$k_l = k_0 + k \times (l - 1)$$

où k_l est le nombre total de caractéristiques après l couches, k_0 est le nombre initial de caractéristiques.

Couches goulot d'étranglement : Bien que chaque couche ne produise que k cartes de caractéristiques de sortie, le nombre d'entrées peut être assez élevé, en particulier pour les couches supplémentaires. Ainsi, une couche de convolution 1×1 peut être introduite comme couche de goulot d'étranglement avant chaque convolution 3×3 pour améliorer l'efficacité et la vitesse des calculs[1].

- **Densenet121 :** DenseNet121 est une variante spécifique de l'architecture DenseNet, composée de 121 couches. Cette architecture se distingue par ses connexions denses entre les couches, où chaque couche reçoit les caractéristiques de toutes les couches précédentes. Cette approche permet une propagation plus efficace de l'information et résout des problèmes comme celui du gradient qui disparaît dans les réseaux profonds. La Figure 3.10 illustre l'architecture globale de DenseNet121 composé de 4 block denses.

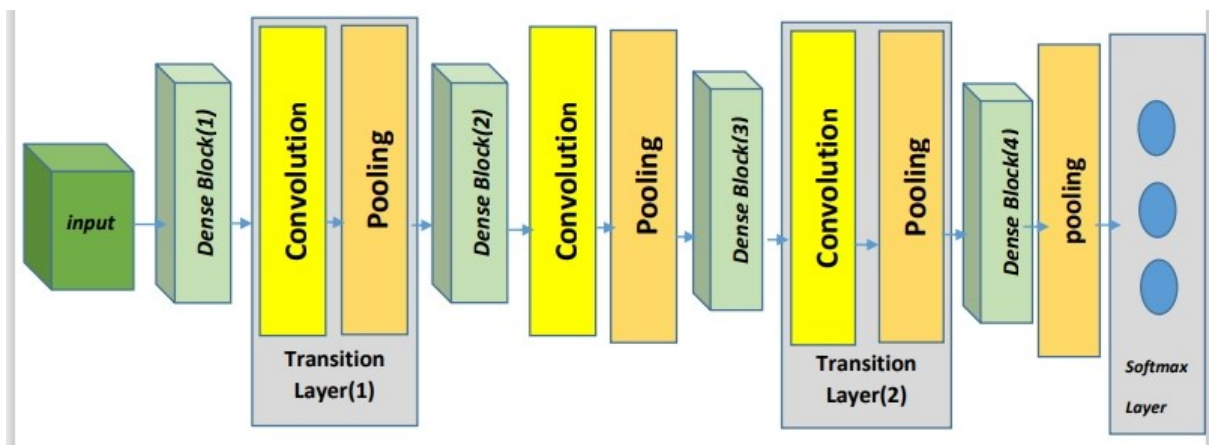


FIGURE 3.10 – Architecture de densenet121

3.4.6.2 Resnet :

ResNet est l'abréviation de RESidual NETwork (Réseaux résiduels). Cette architecture définit véritablement la profondeur des réseaux d'apprentissage en profondeur. Les ResNets sont constitués de plusieurs modules résiduels empilés, qui forment le bloc de construction de base de l'architecture.. ResNet peut aller jusqu'à 152 couches et comprend environ 25 millions de paramètres. Il prend en charge des centaines de couches convolutives et est l'un des premiers à utiliser la normalisation par lots. Les blocs de construction de base de ResNet incluent les blocs convolutifs et d'identité, utilisant la fonction d'activation ReLU et la normalisation des lots[18].

ResNet50 est une variante du modèle ResNet qui comporte 48 couches de convolution ainsi qu'une couche MaxPool et une couche Average Pool. Il comporte $3,8 \times 10^9$ opérations à virgule flottante. Il s'agit d'un modèle ResNet largement utilisé et la figure 3.11 illustre l'architecture ResNet50 .

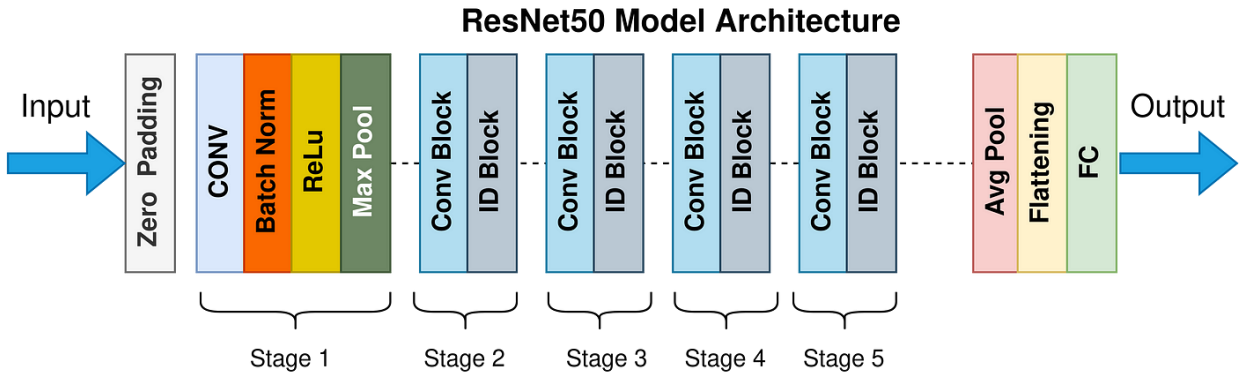


FIGURE 3.11 – Architecture de Resnet50

3.4.6.3 VGGnet :

VGG signifie Visual Geometry Group ; il s’agit d’une architecture standard de réseau neuronal convolutif profond (CNN) avec plusieurs couches. Le « profond » fait référence au nombre de couches avec VGG-16 ou VGG-19 composé de 16 et 19 couches convolutives.

L’architecture VGG constitue la base de modèles de reconnaissance d’objets révolutionnaires. Développé comme un réseau neuronal profond, le VGGNet dépasse également les références sur de nombreuses tâches et ensembles de données au-delà d’ImageNet . De plus, elle reste aujourd’hui l’une des architectures de reconnaissance d’images les plus populaires [16].

Le modele VGG19 se compose de 19 couches dont 16 couches de convolution ainsi 3 couches connectes ,et 5 couche de pooling la figure 3.12 illustre son architecture.

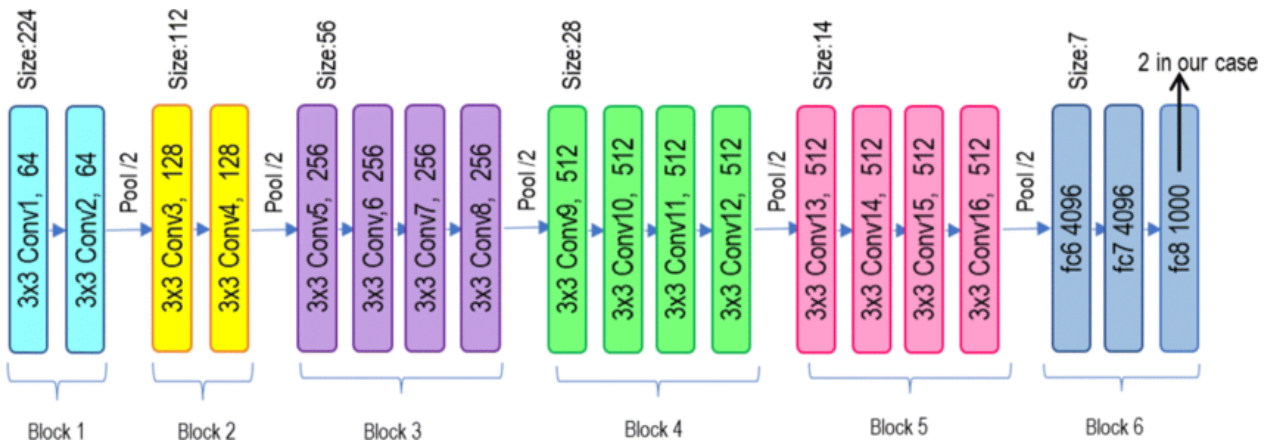


FIGURE 3.12 – Architecture de Vgg19

3.4.6.4 LeNet :

L'architecture LeNet la plus efficace et la plus connue et elle a été utilisée pour lire les codes postaux, les chiffres, ..etc. Cette architecture contient 4 couches convolutive (CONV) et mise en commun (Pooling) alternées, suivies de 3 couches entièrement connectées (fully-connected). LeNet était la première architecture CNN, qui non seulement réduit le nombre de paramètres mais a pu apprendre les caractéristiques de pixels bruts automatiquement [39].

LeNet-5 se compose de 7 couches principales, dont des couches convolutives (CONV), de mise en commun (Pooling), et des couches entièrement connectées (fully-connected). Cette architecture a été l'une des premières à démontrer l'efficacité des CNN pour l'apprentissage des caractéristiques.

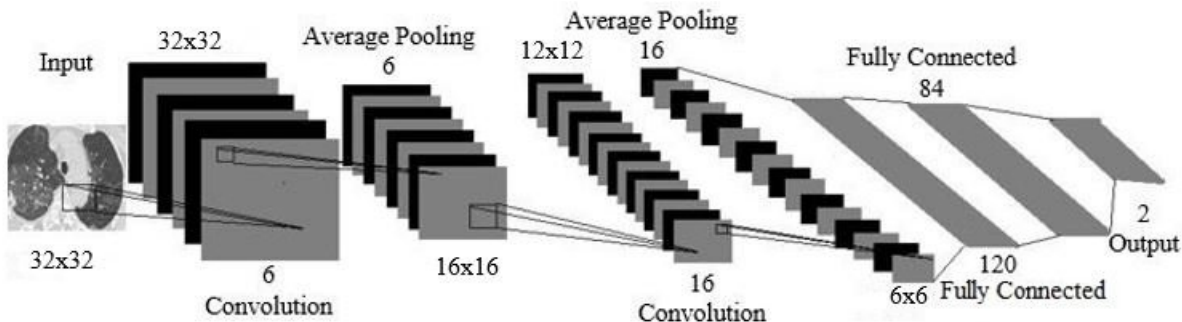


FIGURE 3.13 – Architecture de Lenet-5

3.5 Conclusion :

Les réseaux de neurones convolutifs (CNN) sont des outils puissants pour l'analyse d'images, permettant une reconnaissance précise et efficace des objets, des visages et des motifs au sein de données visuelles. Leur capacité à apprendre automatiquement à partir de données brutes les rend particulièrement adaptés à des tâches complexes de vision par ordinateur et de traitement d'images.

Les CNN ont révolutionné divers domaines, allant de la reconnaissance faciale à la conduite autonome, en passant par le diagnostic médical. Leur impact continuera à croître à mesure que la recherche progresse et que de nouvelles applications innovantes sont développées. Pour illustrer ces avancées, nous allons réaliser une implémentation concrète de ces concepts, en nous concentrant sur l'analyse d'images IRM pour la classification des tumeurs cérébrales.

Chapitre 4

Implémentation , Résultats et Discussions

4.1 Introduction :

Dans ce chapitre, nous présentons l'implémentation et les résultats de notre travail qui consiste à classifier les images IRM en trois tumeurs : gliomes, méningiomes et tumeurs pituitaires. Nous détaillons les outils et environnements utilisés, notamment Python et Google Colab, ainsi que les étapes de développement du système de classification.

Trois modèles de réseaux de neurones ont été expérimentés : DenseNet121, VGG19, un modèle hybride que nous avons proposé, qui combine les forces des deux premières architectures. Ce chapitre aborde en profondeur la préparation des données, la construction des modèles et l'analyse des performances obtenues, mettant en lumière l'efficacité et la pertinence de notre approche pour le diagnostic des tumeurs cérébrales.

4.2 Environnement et outils de développements

4.2.1 Environnement matériel

L'apprentissage profond, ou deep learning, est un ensemble de techniques d'apprentissage automatique reposant sur des méthodes mathématiques pour modéliser des données. Cette approche nécessite des calculs intensifs et dépend fortement des ressources matérielles, en particulier des GPU (unités de traitement graphique), qui excellent dans l'exécution rapide de tâches spécifiques grâce à leur capacité de parallélisme massif.

Pour notre projet, nous utilisons la configuration matérielle suivante :

- **PC portable** : Lenovo ThinkPad X280 avec 8 Go de RAM.
- **Microprocesseur** : Intel(R) Core(TM) i5-7300U CPU @ 2.60 GHz (2.71 GHz).
- **Disque dur** : SSD de 512 Go.

- **Système d’exploitation** : Windows 11, 64 bits, processeur x64.
- Nous avons également utilisé Google Compute Engine avec une ressource GPU, comprenant 1.2 Go de RAM système sur un total de 12.7 Go, et 24.1 Go d’espace disque utilisé sur un total de 78.2 Go.

4.2.2 Environnement logiciel

4.2.2.1 Latex :

LaTeX est un langage et un système de composition de documents. particulièrement apprécié dans les domaines techniques et scientifiques pour sa capacité à automatiser la mise en page et à composer des formules mathématiques complexes. Développé par Leslie Lamport au début des années 1980 et maintenu depuis 1989 par une équipe de bénévoles du projet LaTeX3, il est utilisé pour divers types de documents, allant des articles et thèses aux lettres et présentations. De plus, de nombreux sites web utilisent un sous-ensemble de LaTeX pour afficher des formules mathématiques[12].

4.2.2.2 Python :



Python est devenu le langage de programmation open source le plus utilisé dans le domaine de l’intelligence artificielle. Sa popularité s’explique par sa capacité à simplifier le développement des modèles d’IA, permettant aux chercheurs et ingénieurs de se concentrer sur les algorithmes et les résultats plutôt que sur la complexité du code. Grâce à sa syntaxe simple et à ses nombreuses bibliothèques dédiées à l’IA, comme TensorFlow, Keras ou PyTorch, Python facilite et accélère la création de solutions d’intelligence artificielle par rapport aux langages plus anciens [13].

Les bibliothèques : La Figure 4.1 illustre quelques bibliothèques utilisées dans ce projet. Ces outils jouent un rôle crucial dans le développement et l’analyse des modèles de classification des images de tumeurs cérébrales. Voici un aperçu des principales bibliothèques employées :



FIGURE 4.1 – Quelques bibliothèque utilisées

- **Tensorflow :**



TensorFlow Est une bibliothèque d'apprentissage en profondeur open source développée par Google. Il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance.

- **Keras :**



Keras

Est une bibliothèque populaire d'apprentissage profond qui offre une interface conviviale pour la création rapide de modèles de réseaux de neurones. Elle est compatible avec plusieurs moteurs de backend tels que TensorFlow, Theano et PyTorch, offrant ainsi aux utilisateurs une flexibilité maximale. Son développement initial par François Chollet et son intégration ultérieure à TensorFlow en font un outil incontournable pour les chercheurs et les développeurs souhaitant expérimenter efficacement dans le domaine de l'apprentissage automatique[6].

- **Scikit-learn :**



Est une bibliothèque Python open source qui propose une large gamme d'algorithmes pour la classification, la régression et le regroupement. Elle inclut des méthodes telles que les Support Vector Machines (SVM), les arbres de décision aléatoire (Random Forests), le boosting de gradient, ainsi que des algorithmes de regroupement comme KMeans et DBSCAN. Cette bibliothèque est largement utilisée pour la création et la formation de modèles de machine learning, offrant des outils puissants et une syntaxe cohérente pour explorer et appliquer différents algorithmes.

- **OpenCV :**



Une bibliothèque open source, est l'outil standard pour la vision par ordinateur (Computer Vision) et le traitement d'images. Utilisée dans le domaine du Deep Learning, elle permet aux machines de comprendre et d'analyser le contenu des images et vidéos. OpenCV est également employée pour la retouche photo, la détection de visages, la reconnaissance de formes et de texte. Cette bibliothèque est largement adoptée par les grandes entreprises pour développer des applications avancées de Computer Vision et de Deep Learning.

- **NumPy :**



Abréviation de "Numerical Python", est une bibliothèque Open Source en Python, largement utilisée pour la programmation scientifique. Elle permet de manipuler des tableaux multidimensionnels ("arrays") et d'effectuer des opérations mathématiques et statistiques complexes de manière efficace. NumPy est réputée pour sa rapidité et sa capacité à intégrer du code C/C++ et

Fortran, tout en étant essentielle pour d'autres bibliothèques comme TensorFlow et Scikit-learn. Importée souvent sous l'alias "np", elle simplifie considérablement le traitement des données en Python[33].

- **Matplotlib :**



Est une bibliothèque destinée à tracer et visualiser des données sous formes de graphiques, c'est-à-dire donner une représentation visuelle des résultats. Nous l'utilisons quand nous devons tracer quelques images pour faire calcul scientifique NumPy et SciPy.

- **pandas :**



Est une bibliothèque Python conçue pour l'analyse et la manipulation de données structurées, offrant des fonctionnalités puissantes et une syntaxe simple..[48].

4.2.2.3 Google colab :



Google Colab est un produit de type carnet de notes Jupyter de Google Research. Un développeur de programmes Python peut utiliser ce carnet pour écrire et exécuter des codes de programmes Python aléatoires à l'aide d'un simple navigateur web, Pour utiliser Colab, vous n'avez pas besoin d'installer de runtime ni de mettre à niveau votre matériel informatique pour répondre aux exigences de charge de travail intensive de Python en termes de CPU/GPU[22].

Google a spécialement programmé cet outil de codage Python basé sur le cloud, en gardant à l'esprit les besoins des programmeurs d'apprentissage automatique, des analystes de big data, des scientifiques des données, des chercheurs en IA et des apprenants de Python.

Le plus intéressant, c'est qu'un seul carnet de notes contient tous les éléments nécessaires pour présenter un projet complet d'apprentissage automatique ou de science des données aux superviseurs ou aux sponsors du programme. Par exemple, votre carnet de notes Colab peut contenir des codes exécutables, des codes Python en direct, du texte enrichi, du HTML, du LaTeX, des images, des visualisations de données, des diagrammes, des graphiques, des tableaux, et bien plus encore.

Google Colab rend la science des données, l'apprentissage profond, les réseaux neuronaux et l'apprentissage automatique accessibles aux chercheurs individuels qui n'ont pas les moyens de s'offrir une infrastructure informatique coûteuse.

- **Utilisation de google colab :**

On peut accéder à Google colab en cliquant sur ce lien <https://colab.research.google.com/> la fenêtre se présente comme suit(figure 4.2) :

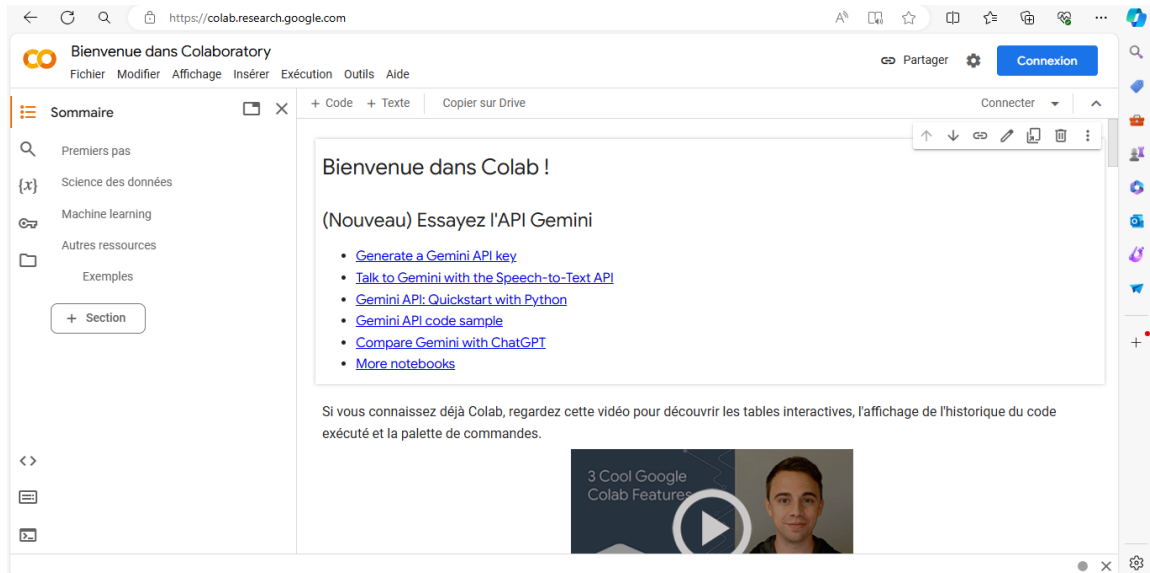


FIGURE 4.2 – Page d'accueil

A partir de cette fenêtre (figure 4.2) on peut créer un nouveau notebook comme indiqué en rouge sur la figure (figure 4.3).

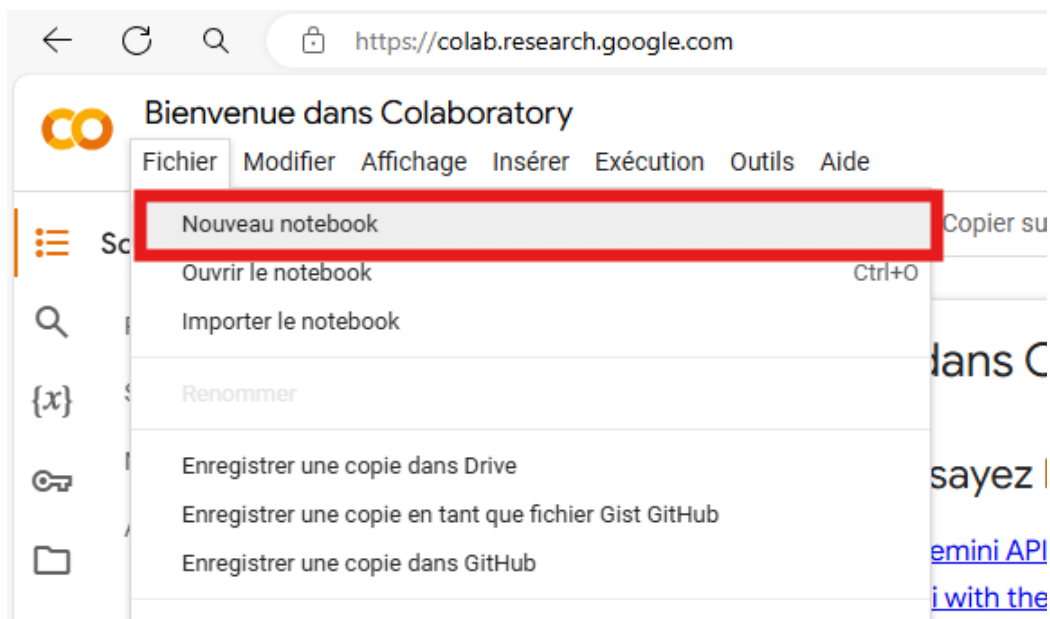


FIGURE 4.3 – Comment ouvrir un nouveau notebook

Afin d'écrire un programme dans cette cloud.

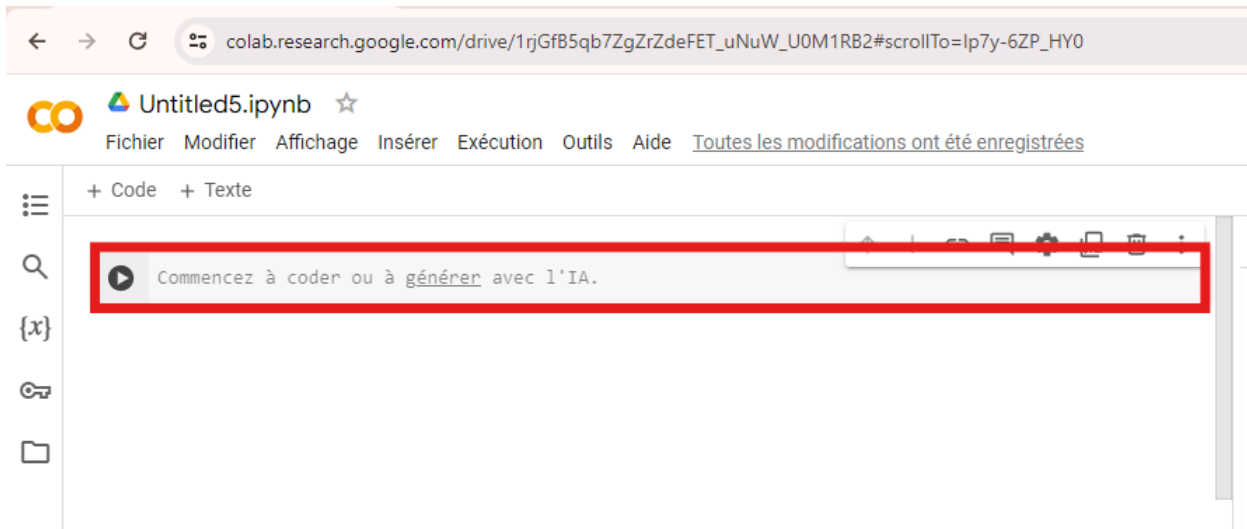


FIGURE 4.4 – Ajouter le programme dans une cellule

Le choix du type d'exécution en GPU est indiqué dans la figure(4.5)et (4.6) .

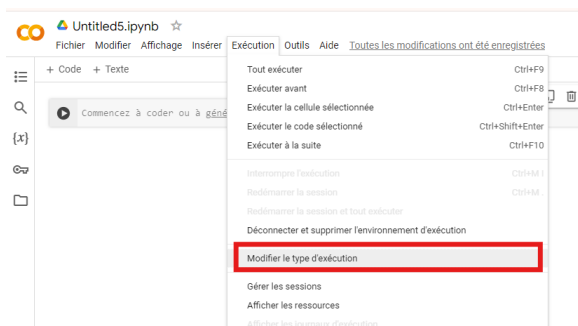


FIGURE 4.5 – Changer le type d'exécution

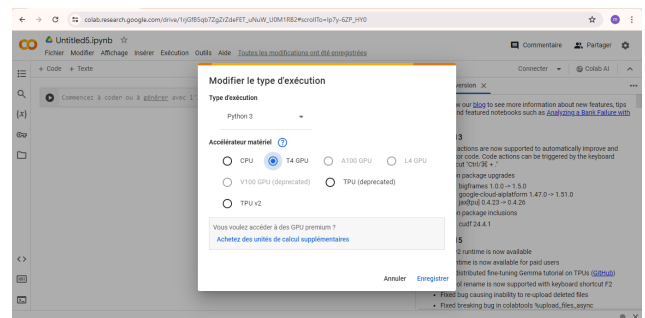


FIGURE 4.6 – Choisir GPU

GPU :Processeurs graphiques ce qui est extrêmement utile pour les tâches gourmandes en calcul, comme l'utilisation de modèles deep learning [28].

4.3 Base de données :

Dans le cadre de ce projet, nous avons utilisé une base de données provenant de Figshare[19], spécifiquement dédiée à l'étude des tumeurs cérébrales. Cette base contient 3064 images pondérées en T1 avec contraste, issues de 233 patients, réparties en trois types de tumeurs (voir figure 4.7) : méningiome (708 images), gliome (1426 images) et tumeur hypophysaire (930 images). Les images sont organisées en quatre fichiers .zip, chacun contenant 766 images. Un point fort de cette base est qu'elle inclut des masques

tumoraux manuels et des méta-données complètes, telles que les labels des tumeurs, les identifiants des patients, ainsi que les coordonnées des bordures tumorales. Cette structure permet une classification précise.

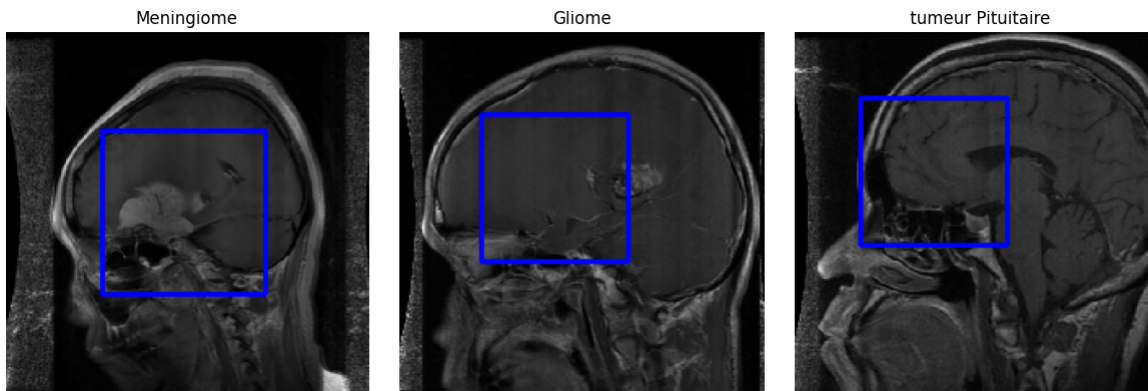


FIGURE 4.7 – Les images de chaque classe de la base de données. La ligne bleu indique la région de la tumeur

4.4 Préparation de données :

- **Normalisation** : Cette étape consiste à ajuster les valeurs des pixels de chaque image pour qu'elles soient dans une échelle cohérente, typiquement entre 0 et 1. Cela se fait en divisant chaque vecteur le long de l'axe spécifié (souvent $axis=1$, correspondant aux canaux ou caractéristiques de chaque image) par sa norme ou une valeur appropriée. La normalisation aide à stabiliser et accélérer l'entraînement du modèle en évitant les grandes variations de valeurs qui pourraient nuire à la convergence.
- **Redimensionnement** : le redimensionnement transforme chaque image d'origine (qui peut avoir des tailles variables) en une image de taille uniforme de 224×224 pixels avant qu'elle ne soit utilisée dans l'entraînement, la validation ou le test du modèle.
- **Division** : Pour garantir une évaluation précise du modèle, les données doivent être séparées en trois ensembles distincts : un jeu d'entraînement (70 %), un jeu de validation (15 %) et un jeu de test (15 %). Avec un total de 3 064 images, cela se traduit par 2451 images pour le jeu d'entraînement, 307 images pour le jeu de validation et 306 images pour le jeu de test.

Le jeu d'entraînement est utilisé pour former les différents modèles. Le jeu de validation permet de sélectionner le modèle le plus performant parmi ceux testés. Enfin, le jeu de test évalue la capacité du modèle à généraliser sur des données nouvelles[8].

- **Étiquetage des données** : Chaque image est associée à une étiquette qui identifie sa classe spécifique. ou l'étiquette de 0 désigne la classe des méningiomes, l'étiquette de 1 correspond aux gliomes, et l'étiquette de 2 représente les tumeurs pituitaires. Ce processus d'étiquetage est essentiel pour la formation supervisée du modèle, car il fournit les repères nécessaires pour que le modèle apprenne à classer correctement les images.

4.5 Modèles CNN utilisées :

4.5.1 Modèle Densenet121 :

Le premier modèle utilisé dans notre travail est le modèle pré-entraîné DenseNet121, ses dernières couches ont été modifiées pour correspondre à notre objectif. Nous appliquons la méthode du transfert d'apprentissage (Transfer Learning) en conservant les couches intérieures et enlevant les dernières couches du modèle. À la place, nous ajoutons une couche de Pooling, suivie d'une couche Dropout (20%) pour la régularisation. Ensuite, nous intégrons une couche entièrement connectée activée par ReLU, puis une couche finale avec Softmax pour la classification. Le nouveau modèle entraînés sur notre base de données. La figure illustre ce processus.

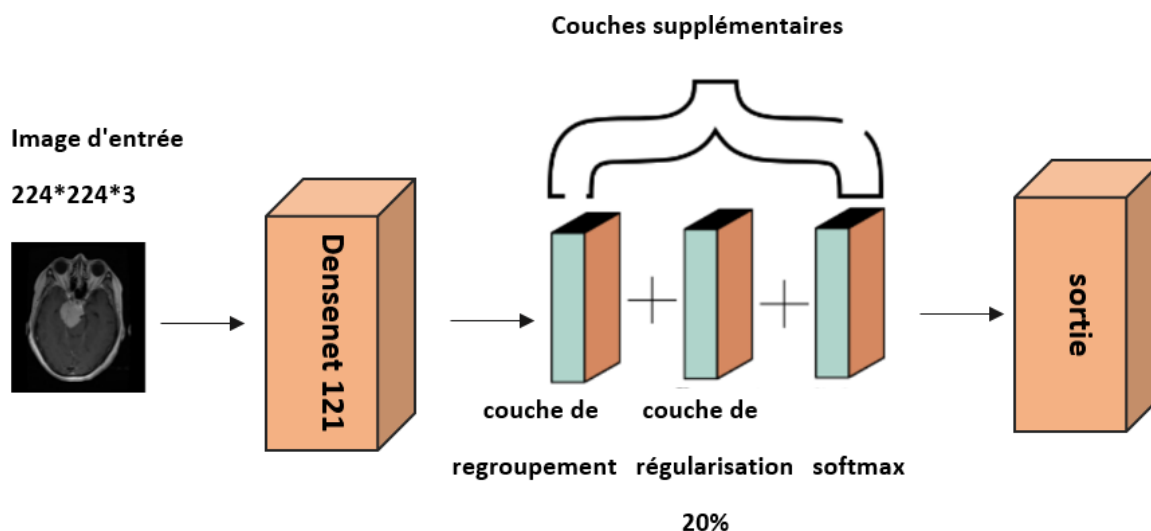


FIGURE 4.8 – Modèle Densnet121

4.5.2 Modèle VGG19 :

Le deuxième modèle utilisé dans notre travail est VGG19 pré-entraîné, dont les dernières couches ont été ajustées pour répondre à nos besoins spécifiques. En appliquant le transfert d'apprentissage, nous avons conservé les premières couches du modèle tout

en supprimant les couches supérieures. Nous avons ensuite ajouté une couche de Pooling, suivie d'une couche Dropout(20%) pour mieux régulariser le modèle. Une couche entièrement connectée activée par ReLU a été ajoutée, et le modèle se termine par une couche Softmax pour la classification finale. Le nouveau modèle sera entraînés sur notre jeu de données. La figure associée illustre cette modification.

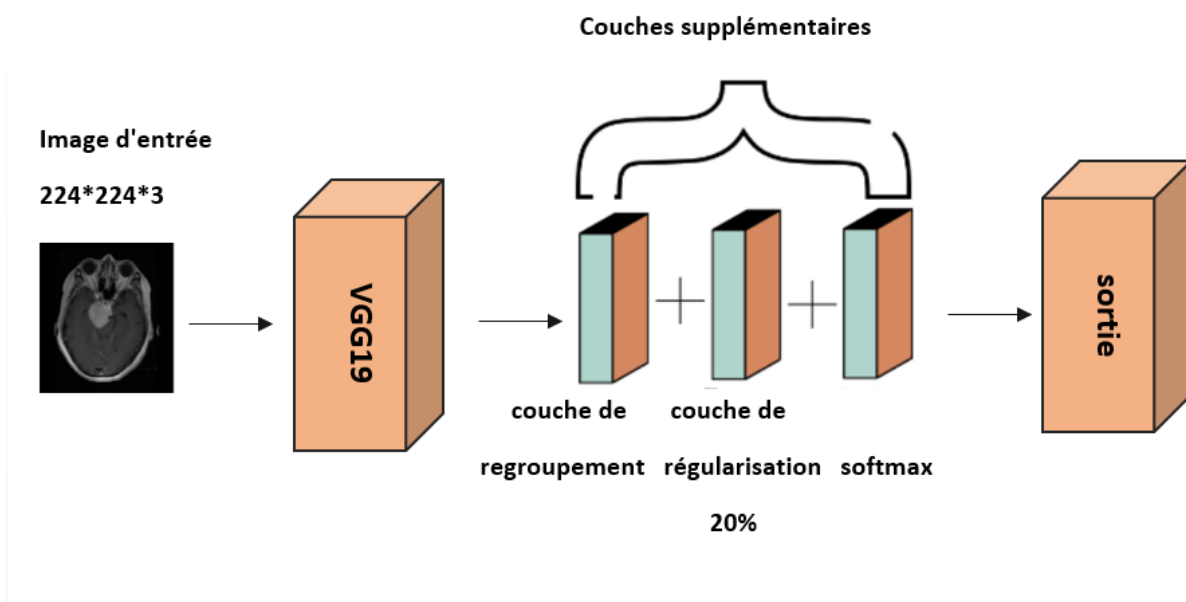


FIGURE 4.9 – Modèle VGG19

4.5.3 Modèle Concaténé :

Dans notre travail ,nous proposons un troisième modèle qui combine les architectures DenseNet121 et VGG19, avec des ajustements apportés aux couches finales pour répondre à nos objectifs spécifiques. En appliquant le transfert d'apprentissage, nous préservons les couches de base tout en enlevant les couches finales de chaque modèle.

Les sorties des couches de DenseNet121 et de VGG19 sont d'abord aplaties, puis combinées pour créer une représentation intégrée unique. À cette représentation, nous ajoutons une couche Dense avec activation ReLU, suivie d'une couche finale Softmax pour la classification. Les couches personnalisées sont optimisés en fonction de notre base de données. La figure correspondante illustre cette architecture concaténée,Cette approche permet non seulement de capturer des caractéristiques complexes, mais aussi de maximiser la complémentarité des deux réseaux, rendant le modèle plus robuste et précis. Elle offre une solution efficace, particulièrement adaptée aux exigences complexes de la médecine numérique.

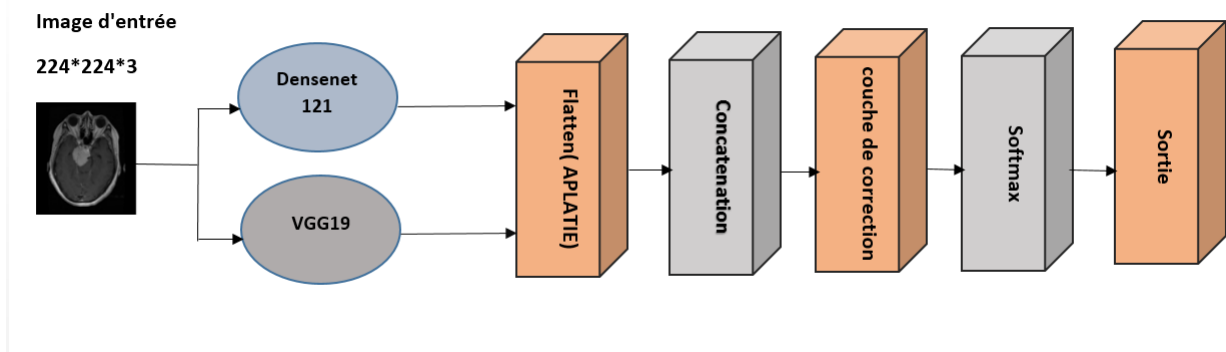


FIGURE 4.10 – Modèle Proposé (La concaténation des deux modèles)

4.6 Métriques de performance :

Les métriques de performance sont cruciales pour évaluer la précision et l'efficacité des modèles de détection d'objets. Elles permettent de mesurer à quel point un modèle est capable d'identifier et de localiser correctement des objets dans des images, ainsi que de comprendre la gestion des erreurs telles que les faux positifs et les faux négatifs. Pour évaluer la performance de la classification d'images, il est essentiel de considérer différentes métriques. Voici un aperçu des métriques pertinentes [7] :

- **Matrice de confusion** : Une matrice de confusion est un tableau qui illustre la performance d'un modèle d'apprentissage automatique en montrant comment les résultats réels se comparent aux résultats prédits. Elle affiche les vrais positifs (TP), les vrais négatifs (TN), les faux positifs (FP) et les faux négatifs (FN) pour chaque classe.
- **Vrai Positif (TP)** : Les instances correctement prédites comme positives.
- **Vrai Négatif (TN)** : Les instances correctement prédites comme négatives.
- **Faux Positif (FP)** : Les instances incorrectement prédites comme positives.
- **Faux Négatif (FN)** : Les instances incorrectement prédites comme négatives.

Par exemple, une matrice de confusion pour un modèle de classification binaire peut ressembler à ceci :

		True Class	
		P	N
Predicted Class	P	TP	FP
	N	FN	TN

FIGURE 4.11 – Matrice de confusion

- **Précision** : La précision mesure le nombre de prédictions correctes faites par le modèle parmi toutes les prédictions positives.

$$\text{Précision} = \frac{TP}{TP + FP}$$

- **Recall (rappel)** : Ou sensibilité, il mesure la proportion de vrais positifs que le modèle a correctement identifiés parmi tous les vrais positifs réels.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Ce métrique est essentiel pour évaluer la capacité du modèle à détecter tous les cas positifs, ce qui est particulièrement important dans des contextes où manquer un cas positif peut avoir des conséquences graves.

- **F1-Score** : Est la moyenne harmonique du rappel et de la précision, et fournit un équilibre entre ces deux métriques.

$$F1\text{-Score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Le F1-Score est utile lorsque l'on souhaite évaluer la performance du modèle en prenant en compte à la fois la précision et le rappel.

- **Accuracy (précision globale)** : La précision globale évalue la proportion totale de prédictions correctes parmi toutes les prédictions faites par le modèle. Elle est définie par :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Cette métrique fournit une vue d'ensemble de l'exactitude du modèle en indiquant combien de prédictions sont correctes par rapport au nombre total de prédictions, ce qui est particulièrement utile lorsque les classes sont équilibrées.

4.6.1 Courbes de performance :

- **Courbe de précision** : Illustre l'évolution de la précision du modèle à travers les époques d'entraînement. Elle permet d'analyser comment la précision des prédictions progresse ou fluctue au fil des itérations, aidant à évaluer l'efficacité du modèle à chaque étape.
- **Courbe de perte** : Affiche la variation de la perte (erreur) du modèle en fonction des époques. Elle montre comment la fonction de perte diminue à mesure que le modèle apprend, ce qui indique une meilleure correspondance aux données d'entraînement. Une stagnation ou une remontée de la perte peut signaler un sur-apprentissage (overfitting) ou une mauvaise convergence.

4.7 Implémentation, Discussions et Comparaison des résultats

4.7.1 Implémentation :

Dans l'implémentation de notre approche, nous avons exploité trois modèles clés : DenseNet121, VGG19, ainsi qu'une méthode de fusion par concaténation. DenseNet121, réputé pour son architecture dense permettant un transfert efficace des informations entre les couches, assure une extraction de caractéristiques profonde et robuste. En parallèle, VGG19, avec sa structure hiérarchique de convolution classique, apporte une autre perspective en capturant des détails plus fins. La fusion des sorties de ces deux modèles par concaténation permet de combiner leurs forces, offrant ainsi une représentation plus complète et diversifiée des données d'entrée. Cette architecture hybride a été optimisée pour maximiser la performance globale et la précision des prédictions. On va suivre les étapes d'implémentation suivantes :

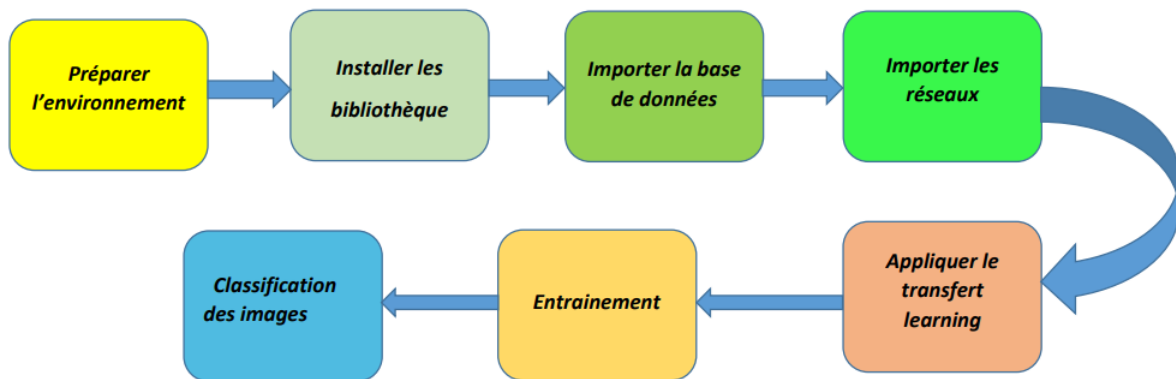


FIGURE 4.12 – Étapes d'Implémentation

Lors de la classification, plusieurs paramètres d'entraînement sont essentiels pour optimiser le modèle. Voici un résumé des principaux paramètres à considérer :

- **Taux d'apprentissage (learning rate)** : Définit la vitesse à laquelle le modèle ajuste ses poids en réponse à l'erreur calculée à chaque itération. Dans notre cas, nous avons utilisé un taux d'apprentissage de 0.00001, permettant ainsi des ajustements plus précis et progressifs des poids pour améliorer la convergence du modèle.
- **Loss** : Évalue l'écart entre les prédictions du modèle et les valeurs réelles.

- **Époques (Epochs)** : Représente le nombre de fois que l'ensemble de données d'entraînement est entièrement parcouru par le modèle. Dans notre cas, le premier modèle a été entraîné sur 200 époques pour favoriser une meilleure convergence. Les deuxième et troisième modèles ont chacun été entraînés sur 100 époques.
- **Taille du lot (Batch size)** : Représente le nombre d'échantillons traités avant chaque mise à jour des paramètres du modèle. Dans notre implémentation, nous avons utilisé une taille de lot de 16, ce qui permet un compromis entre l'efficacité de la mémoire et la stabilité de l'apprentissage.

4.7.2 Résultats et Discussions

4.7.2.1 Résultats de Densenet121 :

Ce modèle a été entraîné avec 200 époques, (voir figure 4.13) lui permettant d'apprendre à identifier des motifs complexes dans les données. Il a atteint une précision d'entraînement de 100 %, démontrant une bonne capacité à réaliser des prédictions correctes sur les données d'entraînement. De plus, il a obtenu une précision de 97,39 % (voir figure 4.14) sur les données de test, avec une erreur de seulement 2,61 %. Bien que ces résultats soulignent la solidité et la performance du modèle, ils ne garantissent pas une fiabilité suffisante pour une utilisation dans le domaine médical, où chaque petite erreur peut avoir des conséquences importantes.

```
Epoch 7/200
154/154 [=====] - 31s 199ms/step - loss: 0.0832 - accuracy: 0.9731 - val_loss: 1.2893 - val_accuracy: 0.7622
Epoch 8/200
154/154 [=====] - 30s 196ms/step - loss: 0.0505 - accuracy: 0.9849 - val_loss: 0.1366 - val_accuracy: 0.9707
Epoch 9/200
154/154 [=====] - 30s 198ms/step - loss: 0.0374 - accuracy: 0.9882 - val_loss: 0.4016 - val_accuracy: 0.8925
Epoch 10/200
154/154 [=====] - 31s 199ms/step - loss: 0.0492 - accuracy: 0.9837 - val_loss: 0.2417 - val_accuracy: 0.9088
Epoch 11/200
154/154 [=====] - 31s 200ms/step - loss: 0.0638 - accuracy: 0.9755 - val_loss: 2.3657 - val_accuracy: 0.6808
Epoch 12/200
154/154 [=====] - 31s 200ms/step - loss: 0.0549 - accuracy: 0.9816 - val_loss: 0.6816 - val_accuracy: 0.8176
Epoch 13/200
...
154/154 [=====] - 30s 197ms/step - loss: 4.4952e-05 - accuracy: 1.0000 - val_loss: 0.2099 - val_accuracy: 0.9739
```

FIGURE 4.13 – Apprentissage du modèle Densenet121

```
Accuracy : 97.39413680781759
Error : 2.6058631921824116
```

FIGURE 4.14 – Accuracy du modèle densenet121

la Figure 4.15 présente la courbe de précision et de perte en fonction des époques d'entraînement. La courbe de précision montre une amélioration continue jusqu'à la convergence, tandis que la courbe de perte diminue progressivement, indiquant que le modèle apprend de manière stable sans risque de sur-apprentissage.

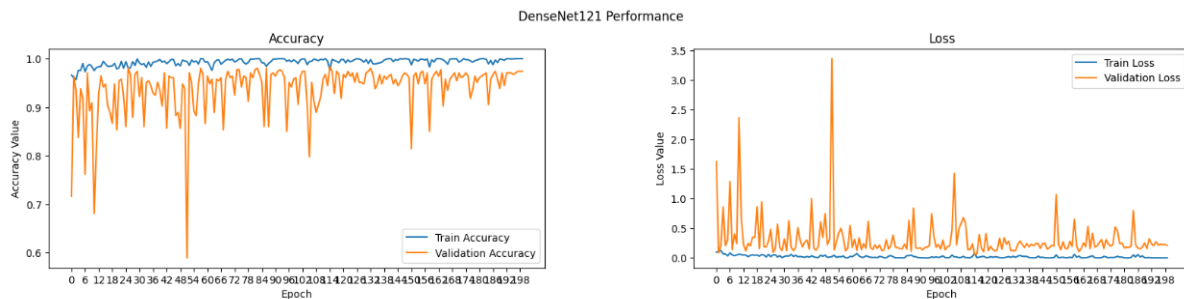


FIGURE 4.15 – Courbe de Precision et Perte du modèle Densnet121

La matrice de confusion présentée en Figure 4.16 indique une très faible incidence d'erreurs de classification. Le modèle a montré une bonne précision pour chaque classe, avec des erreurs minimales, suggérant une capacité de discrimination acceptable entre les différentes catégories. Cependant, cette performance doit encore être améliorée pour répondre aux exigences critiques de la classification des tumeurs cérébrales.

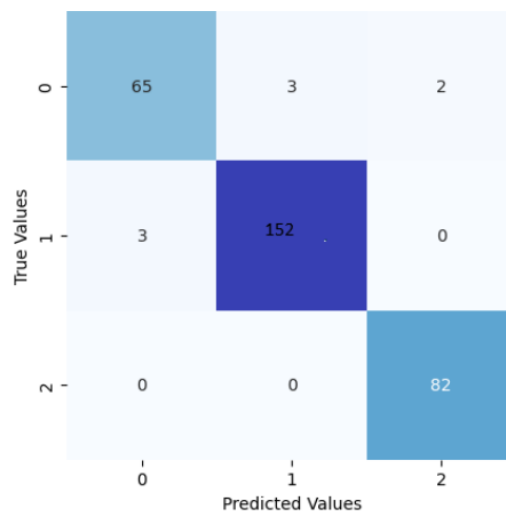


FIGURE 4.16 – Matrice de confusion de Densnet121

Le rapport de performance détaillé en Figure 4.17 confirme que DenseNet121 atteint des valeurs élevées en précision, rappel et F1-score, en particulier pour les classes 1 et 2, illustrant son efficacité globale dans la classification des images..

```

Report :
          precision    recall  f1-score
0         0.96         0.93         0.94
1         0.98         0.98         0.98
2         0.98         1.00         0.99
    
```

FIGURE 4.17 – Rapport de Classification de Densenet121

4.7.2.2 Résultats de VGG19 :

Le modèle VGG19 a été entraîné pendant 100 époques (Figure 4.18), ce qui lui a permis de bien apprendre les caractéristiques des données. Au cours de l'entraînement, la perte a considérablement diminué, atteignant des niveaux très bas, tandis que la précision a atteint 100% sur les données d'entraînement. À la fin du processus, la précision était de 96,74%(voir figure 4.19), avec un taux d'erreur de 3,26%. Bien que ces résultats montrent que le modèle peut faire des prédictions précises, ils ne suffisent pas à garantir une fiabilité adéquate dans le domaine médical, où même de petites erreurs peuvent avoir des conséquences significatives. (À noter que le modèle a également été entraîné pendant 200 époques, avec des résultats similaires) .

```

Epoch 5/100
154/154 [=====] - 43s 276ms/step - loss: 0.0604 - accuracy: 0.9755 - val_loss: 0.1488 - val_accuracy: 0.9511
Epoch 6/100
154/154 [=====] - 44s 283ms/step - loss: 0.0518 - accuracy: 0.9829 - val_loss: 0.1755 - val_accuracy: 0.9316
Epoch 7/100
154/154 [=====] - 43s 278ms/step - loss: 0.0456 - accuracy: 0.9849 - val_loss: 0.1148 - val_accuracy: 0.9577
Epoch 8/100
154/154 [=====] - 43s 279ms/step - loss: 0.0331 - accuracy: 0.9898 - val_loss: 0.1815 - val_accuracy: 0.9414
Epoch 9/100
154/154 [=====] - 44s 284ms/step - loss: 0.0378 - accuracy: 0.9857 - val_loss: 0.1035 - val_accuracy: 0.9544
Epoch 10/100
154/154 [=====] - 44s 284ms/step - loss: 0.0370 - accuracy: 0.9869 - val_loss: 0.2326 - val_accuracy: 0.9381
Epoch 11/100
154/154 [=====] - 43s 278ms/step - loss: 0.0288 - accuracy: 0.9914 - val_loss: 0.1983 - val_accuracy: 0.9511
Epoch 12/100
154/154 [=====] - 44s 285ms/step - loss: 0.0467 - accuracy: 0.9849 - val_loss: 0.2512 - val_accuracy: 0.8860
Epoch 13/100
...
154/154 [=====] - 44s 284ms/step - loss: 6.4226e-06 - accuracy: 1.0000 - val_loss: 0.1866 - val_accuracy: 0.9674
    
```

FIGURE 4.18 – Apprentissage de modèle VGG19

```

Accuracy : 96.74267100977198
Error : 3.257328990228018
    
```

FIGURE 4.19 – Accuracy et Erreur du VGG19

La Figure 4.20 montre les courbes de précision et de perte pendant l'entraînement. La précision augmente rapidement, bien qu'elle connaisse quelques variations au fil des époques, ce qui montre que le modèle continue à s'améliorer. La courbe de perte diminue globalement, mais avec quelques fluctuations, indiquant que l'apprentissage est stable, même si de petits ajustements se produisent sans signe de surapprentissage..

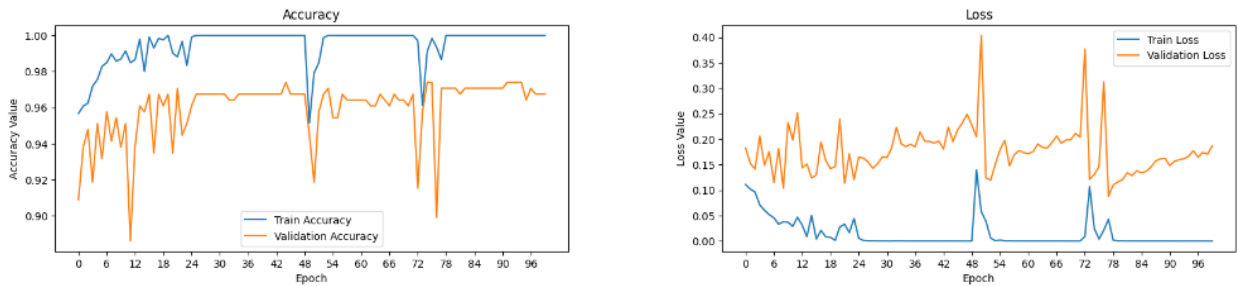


FIGURE 4.20 – Les Courbes du modèle VGG19

La Figure 4.21 présente la matrice de confusion pour VGG19, offrant une vue détaillée des performances du modèle pour chaque classe. Les résultats sont les suivants :

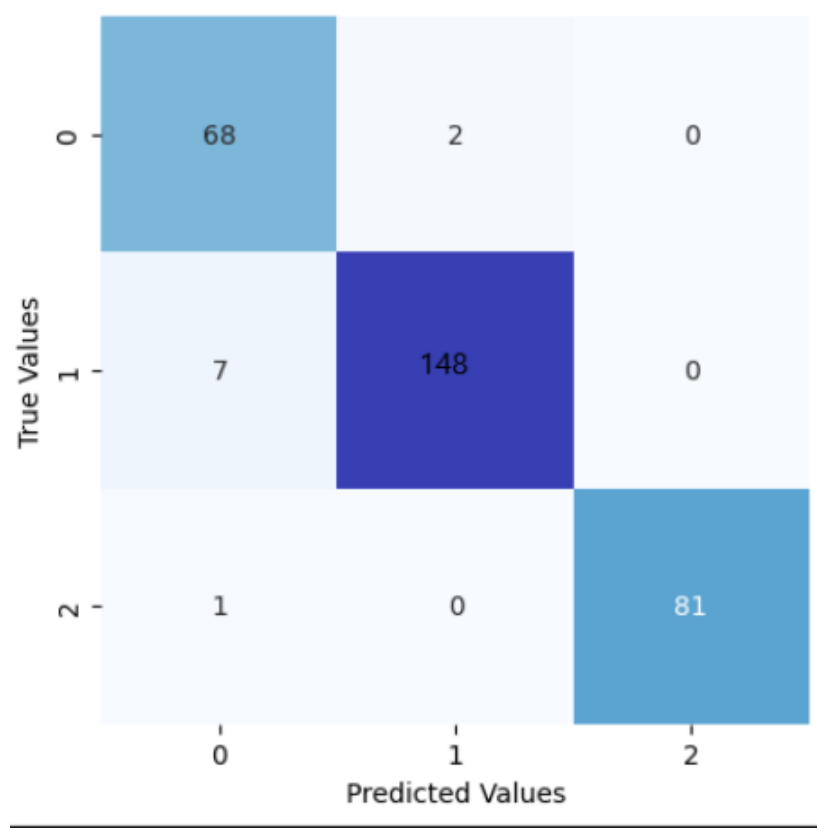


FIGURE 4.21 – Matrice de confusion De VGG19

Le rapport de performance détaillé dans la Figure 4.22 met en avant des scores élevés en précision, rappel et F1-score, bien que ces scores ne surpassent pas ceux de modèle DenseNet121 ou . Cela indique que VGG19 reste efficace dans la classification globale, mais présente quelques limitations dans la minimisation des erreurs.

```
Report :
```

	precision	recall	f1-score	support
0	0.89	0.97	0.93	70
1	0.99	0.95	0.97	155
2	1.00	0.99	0.99	82

FIGURE 4.22 – Le rapport de classification du modèle VGG19

4.7.2.3 Résultats Modèle Proposé :

Le modèle concaténé a été entraîné sur 100 époques (Figure 4.23), démontrant une nette amélioration dans ses performances. En combinant les deux modèles , la perte a considérablement diminué pour atteindre des valeurs très faibles, tandis que la précision a culminé à 100 % à la fin de l'entraînement. La précision s'est établie à 98,70%, avec une perte de 0,1157 (Figure 4.24). Ces résultats soulignent l'efficacité de cette approche combinée pour générer des prédictions précises sur des données non vues, renforçant la robustesse du modèle pour des applications nécessitant fiabilité et performance.(À noter que le modèle a également été entraîné pendant 200 époques, avec des résultats similaires).

```
Epoch 9/100
154/154 - 72s - 469ms/step - accuracy: 0.9890 - loss: 0.0324 - val_accuracy: 0.9804 - val_loss: 0.0526
Epoch 10/100
154/154 - 82s - 534ms/step - accuracy: 0.9943 - loss: 0.0129 - val_accuracy: 0.9641 - val_loss: 0.2175
Epoch 11/100
154/154 - 82s - 531ms/step - accuracy: 0.9951 - loss: 0.0204 - val_accuracy: 0.9869 - val_loss: 0.0462
Epoch 12/100
154/154 - 82s - 531ms/step - accuracy: 0.9988 - loss: 0.0034 - val_accuracy: 0.9837 - val_loss: 0.0556
Epoch 13/100
...
Epoch 99/100
154/154 - 72s - 468ms/step - accuracy: 1.0000 - loss: 3.1663e-06 - val_accuracy: 0.9902 - val_loss: 0.0578
Epoch 100/100
154/154 - 72s - 469ms/step - accuracy: 1.0000 - loss: 3.3800e-06 - val_accuracy: 0.9902 - val_loss: 0.0579
```

FIGURE 4.23 – Appretissage du modèle proposé

```
Test Loss: 0.1157
Test Accuracy: 0.9870
```

FIGURE 4.24 – Accuracy et erreur du modèle proposé

La figure 4.25 illustre des courbes de précision et de perte, montrant une tendance générale à l'amélioration au fil des époques d'entraînement."

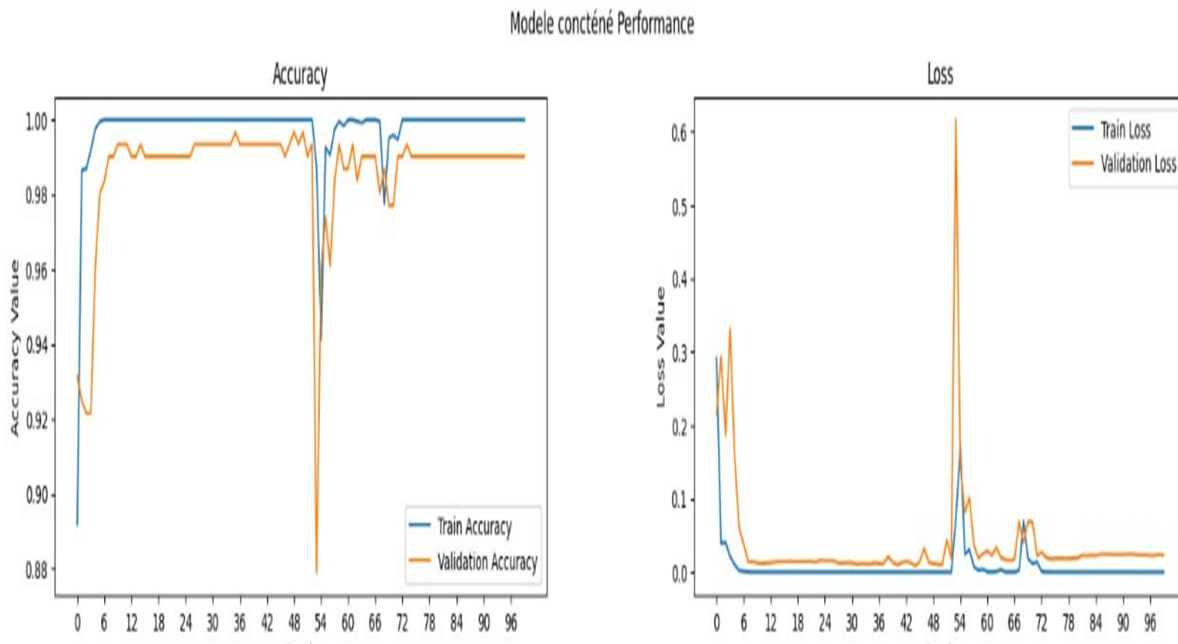


FIGURE 4.25 – Courbes de précision et de perte du modèle proposé

La matrice de confusion (Figure 4.26) montre une faible fréquence d'erreurs de classification. Chaque classe est correctement identifiée, démontrant la capacité du modèle à différencier efficacement les catégories. Cette performance accrue valide l'avantage de combiner les deux réseaux, car elle permet de minimiser les erreurs tout en maximisant la précision pour chaque classe.

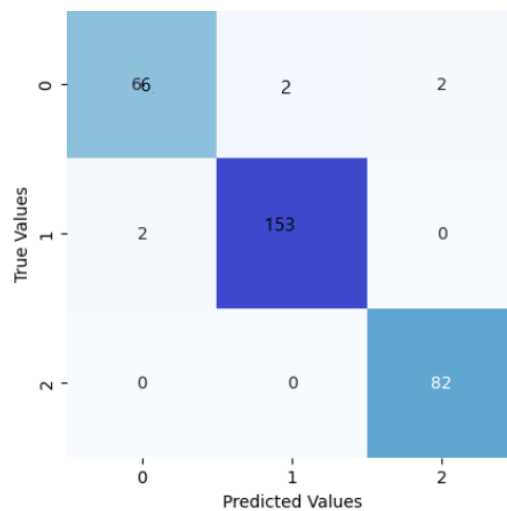


FIGURE 4.26 – Matrice de confusion du modèle proposé

Le rapport de classification en Figure 4.27(voir figure) met en évidence des valeurs élevées pour la précision, le rappel et le F1-score, avec des résultats proches de 100 % pour toutes les classes. Le modèle concaténé se distingue par des performances remarquables, minimisant les erreurs et offrant une classification précise pour l'ensemble des catégories, dépassant ainsi les modèles individuels.

```

Classification Report:
              precision    recall  f1-score
0             1.00         0.94     0.97
1             0.99         1.00     0.99
2             0.98         1.00     0.99
    
```

FIGURE 4.27 – Rapport de Classification du modèle proposé

Voici quelque exemples des images obtenue

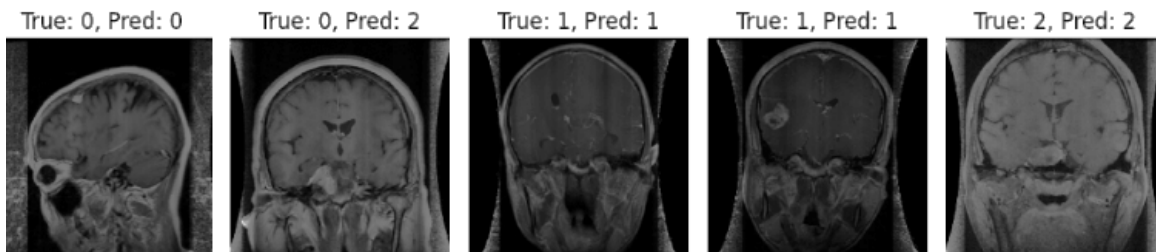


FIGURE 4.28 – Quelques prédictions

4.7.3 Comparaison des Résultats

Modèle	Précision (Accuracy)	Erreur (Error)
DenseNet121	97.39%	2.61%
VGG19	96.74%	3.26%
Concaténé	98.70%	1.15%

TABLE 4.1 – Comparaison des résultats des trois modèles CNN

Les résultats de la comparaison entre les modèles DenseNet121, VGG19 et le modèle concaténé montrent des performances distinctes, reflétant les points forts et les limites de chaque approche.

Le modèle concaténé affiche une précision globale élevée de 98,70% et un taux d'erreur de 1,15%. Cette performance souligne l'efficacité de la fusion des deux architectures, permettant une classification précise dans la plupart des cas. Le faible taux d'erreur indique également une capacité accrue à généraliser les caractéristiques des images d'IRM des tumeurs cérébrales.

Le modèle DenseNet121 suit avec une précision globale de 97,39% et un taux d'erreur de 2,61%. Ce modèle, bien qu'excellent, est légèrement moins performant que le modèle concaténé, mais il offre toujours des résultats fiables et robustes.

Le modèle VGG19, quant à lui, présente une précision de 96,74% avec un taux d'erreur de 3,26%. Bien que ses performances restent compétitives, ce modèle montre une légère baisse en comparaison avec les autres, surtout en ce qui concerne certaines classes spécifiques. Comme dans notre cas la classe 0 qui représente la tumeur la plus difficile à détecter (figure 4.7 Classe : méningiomes).

On constate que l'approche concaténée optimise les performances en combinant l'intégralité des deux architectures, permettant ainsi d'obtenir des prédictions plus précises. Cette fusion met en évidence l'importance d'utiliser différentes architectures dans leur ensemble pour améliorer la robustesse des classifications, soulignant une stratégie efficace dans le domaine de la détection des tumeurs cérébrales. Cela ouvre la voie à des développements futurs, où des modèles encore plus intégrés pourraient être explorés pour maximiser l'efficacité et la fiabilité des résultats.

Bien que cette performance soit impressionnante, il est important de noter que dans le domaine médical, où la précision est cruciale, même de petites erreurs peuvent avoir des conséquences significatives. Ainsi, il est essentiel de continuer à améliorer les performances et la robustesse des modèles pour répondre aux exigences critiques de la détection des tumeurs cérébrales.

4.8 Conclusion :

En conclusion, les résultats révèlent que chaque modèle possède des atouts et des faiblesses distincts. Le modèle concaténé, combinant DenseNet121 et VGG19, se démarque par son excellente précision et sa robustesse dans la classification des tumeurs cérébrales. DenseNet121 suit de près avec des performances solides, tandis que VGG19, bien qu'efficace dans certaines catégories, présente une cohérence moins prononcée. Cependant, malgré ces résultats prometteurs, il est crucial de souligner que, dans le domaine médical, même de petites erreurs peuvent avoir des conséquences graves, ce qui nécessite des améliorations continues pour garantir la fiabilité et la précision des modèles.

Conclusion générale

Ce mémoire avait pour objectif d’explorer l’utilisation des architectures de réseaux de neurones convolutifs (CNN) pour classifier trois types de tumeurs cérébrales : gliome, méningiome et tumeur pituitaire, à partir d’images d’IRM. Nous avons d’abord consacré une partie à la présentation des tumeurs cérébrales et à la classification des images, avant d’explorer les réseaux de neurones convolutifs et leur rôle dans l’analyse des images médicales.

Grâce à l’application de différentes architectures CNN sur une base de données d’images IRM, nous avons réussi à classifier ces trois types de tumeurs cérébrales. Les modèles DenseNet121, VGG19, ainsi qu’un modèle concaténé combinant ces deux architectures, ont été implémentés et testés. Les résultats obtenus montrent que, bien que ces architectures soient différentes, elles offrent des performances solides en termes de classification. Cela souligne l’importance du choix architectural dans l’amélioration des performances des modèles.

En termes de résultats chiffrés, DenseNet121 a atteint une précision de 97.39% avec une erreur de 2.61%, tandis que VGG19 a obtenu une précision de 96.74% avec une erreur de 3.26%. Le modèle concaténé, quant à lui, a surpassé les deux autres en offrant une précision de 98.70% et une erreur de seulement 1.15%. Bien que ces résultats soient encourageants, notamment avec le modèle concaténé qui atteint près de 99% de précision, il est important de noter que dans le domaine médical, une telle précision reste encore insuffisante pour un diagnostic totalement fiable. En effet, toute erreur dans le diagnostic médical peut avoir des conséquences graves, ce qui souligne la nécessité d’une précision proche de la perfection pour l’application clinique.

Toutefois, il est essentiel de rappeler que ces technologies ne visent pas à remplacer les médecins, mais à les assister dans le processus de diagnostic. Les progrès constants dans l’analyse des données biomédicales, notamment grâce à l’apprentissage automatique, ouvrent la voie à des outils diagnostiques combinant imagerie médicale et données génomiques, offrant ainsi des perspectives prometteuses pour des performances encore plus avancées.

En conclusion, cette étude met en lumière le potentiel des réseaux de neurones convolutifs pour le diagnostic médical et montre que, malgré des résultats très positifs, des améliorations sont encore nécessaires pour atteindre un niveau de précision compatible avec les exigences médicales. À l’avenir, l’augmentation des données et l’optimisation des pré-traitements d’images pourraient encore renforcer la performance des modèles et leur capacité à généraliser sur des ensembles de données plus variés.

Bibliographie

- [1] Architecture of densenet-121. <https://iq.opengenus.org/architecture-of-densenet121/>.
- [2] Irm cérébrale. *ELSAN*.
- [3] Quels sont les avantages et les inconvénients de l'utilisation de réseaux de neurones convolutifs pour la reconnaissance d'images? <https://fr.linkedin.com/advice/0/what-advantages-disadvantages-using-convolutional>.
- [4] info cancer. *Arcagy Gineco*, 2023.
- [5] K-means clustering algorithm in machine learning. *Javatpoint*, 2023.
- [6] Keras <https://en.wikipedia.org/wiki/keras>, août 2024.
- [7] Les méthodes d'évaluation des modèles de machine learning. *MLExpertise.com*, 2024.
- [8] Tp sélectionnez le nombre de voisins dans un knn. <https://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308246-tp-selectionnez-le-nombre-de-voisins-dans-un-knn>, 2024. Mis à jour le 06/02/2024.
- [9] Tumeurs cÉrÉbrales. *Docteur 360*, 2024.
- [10] Machine learning : l'algorithme des k plus proches voisins. *MonCoachData*, 25 décembre 2023.
- [11] Anatomie et physiologie du système nerveux. <https://cancer.ca/fr/cancer-information/cancer-types/neuroblastoma/what-is-neuroblastoma/the-nervous>, Dernière révision médicale : janvier 2020.
- [12] Latex. <https://fr.wikipedia.org/wiki/LaTeX>, La dernière modification de cette page a été faite le 10 avril 2024.
- [13] Python : définition et utilisation de ce langage informatique. *JDN*, Mis à jour le 31/03/20.

- [14] H. ; Shen J. ; Chen Z. ; Zhang Z. ; Ge X. Synergy . Xiao, G. ; Wang. Synergy factorized bilinear network with a dual suppression strategy for brain tumor classification in mri. *Micromachines*, 2022.
- [15] S. Alaraimi, K.E. Okedu, H. Tianfield, R. Holden, and O. Uthmani. Transfer learning networks with skip connections for classification of brain tumors. *International Journal of Imaging Systems and Technology*, 31 :1564–1582, 2021.
- [16] Gaudenz Boesch. Réseaux convolutifs très profonds vgg (vggnet) – ce que kou doit savoir. <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>.
- [17] P. Bulla, L. Anantha, and S. Peram. Deep neural networks with transfer learning model for brain tumors classification. *Traitement du Signal*, 37 :593–601, 2020.
- [18] Written by Data Science Team. Réseaux neuronaux résiduels – ce que vous devez savoir. datascience.eu/fr, Published on 03 May 2020.
- [19] Jun Cheng. Brain tumor dataset. 2 avril 2017.
- [20] Walid Chrimni. Support vector machines (svm). *La Revue IA*, September 2020.
- [21] Shuvam Das. Implementing densenet-121 in pytorch : A step-by-step guide. *Deepkapha Notes*, 2023.
- [22] Tamal Das. Google colab : Tout ce que vous devez savoir. *GEEKFLARE*, Dernière mise à jour : mai 17, 2024.
- [23] PAR CÉLINE DELUZARCHE. Deep learning : qu’est-ce que c’est? *futurasciences.com*, LE 14 OCTOBRE 2023.
- [24] Dr Loïc Feuvret Dr Matthieu Peyre. Arc brochure cancer. *Fondation pour la recherche sur le cancer*, 2022.
- [25] Gérard DREYFUS. RÉseaux de neurones. universalis.fr/encyclopedie, 2024.
- [26] Département américain de la Santé et des Services sociaux, Instituts nationaux de la santé. Magnetic resonance imaging (mri). Accédé en septembre 2024.
- [27] Omar el Kharki et Jamila Mechbough et Danielle Ducrot et Mohamed Rouchdi. Panorama sur les méthodes de classification des images satellites et techniques d’amélioration de la précision de la classification. *ResearchGate*, October 2021.
- [28] Denis Eleyehou. Google colab : la force du cloud pour l’apprentissage automatique. *DataScientest*, février 2024.
- [29] Keith D. Foote. Une brève histoire de l’apprentissage profond, 2022. Consulté le 4 février 2022.
- [30] STEVEN A GOLDMAN. presentation des tumeurs cerebrales. mcgill.ca, 2018.

- [31] Antoine Grignola. Qu'est ce qu'un cnn ? un convolutional neural network. *data-bird*, Mis à jour le 13/6/2024.
- [32] Dr Andreas Hottinger. Gliome. *Centres interdisciplinaires d'oncologie*, 2024.
- [33] Raphael Kassel. Numpy : la bibliothèque python la plus utilisée en data science. *DataScientest*, avril 2021.
- [34] Julien Krywyk. Classification d'images : les réseaux de neurones convolutifs en toute simplicité. *blog.octo.com*, 2016.
- [35] V.R. KVA, M.; Rajendran. Glioma tumor grade identification using artificial intelligent techniques. *J. Med. Syst.*, 2019.
- [36] Mlle Hadjer LAGUEL. Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images irm basé sur les systèmes multi-agents. Master's thesis, Faculté d'Électronique et d'Informatique, Département d'informatique, 2010. Mémoire du Projet de Fin d'Études pour l'obtention du diplôme d'Ingénieur d'État en Informatique, soutenu devant le jury composé de Mme F. SOUAMI (Présidente), Mme S. BOUAGAR (Examinatrice), Mme F. DJIDEL (Examinatrice).
- [37] BY MAPASYST. What's the difference between a supervised and unsupervised image classification ? *GEOSPATIAL TECHNOLOGY*, AUGUST 21, 2019 BY MAPASYST.
- [38] Dr. Merzougui. *chapitre3 : Réseau de neurones convolutionnel CNN*. Université de Batna2.DZ. Notes de cours.
- [39] SEKKIL Hicham Mohamed and MEBROUKI Mahmoud. Etude comparative entre les différentes architectures des réseaux de neurones convolutifs (cnns) pour la détection de la fatigue du conducteur. *thesis.essa-tlemcen.dz*, 2021.
- [40] Inès Montenon. Irm : rôle, déroulement, avantages et effets secondaires. *Qare par healthhero*, 2023.
- [41] D. Y. Moualek. *Deep Learning pour la classification des images*. Mémoire de doctorat.
- [42] N. ; Mallick P.K. ; Zymbler M. ; Kumar S Nayak, D.R. ; Padhy. Brain tumor classification using dense efficient-net. *axioms*. 2022.
- [43] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib. A deep learning model based on concatenation approach for the diagnosis of brain tumor. *IEEE Access*, 8 :55135–55144, 2020.
- [44] Réalisé par : Madame FELLAG. Mr BELHADJER Hakim. Mr SAROUER Brahim. Classification des images avec les réseaux de neurones convolutionnels. Master's thesis, université MOUloud Maamri TIZI OUZOU, Promotion 2017-2018.

- [45] Passeport Santé. Matière grise dans le cerveau : quel est son rôle?, 2023. Accédé en mars 2023.
- [46] Par Vonintsoa R. Classification d'image : le guide complet. *INTELLIGENCE-ARTIFICIELLE.COM*, Mis à jour :il y a 2 ans.
- [47] A. ; Javed A.R. ; Shabbir M. ; Baker T. ; Obe D. A. J. Rizwan, M. ; Shabbir. Brain tumor and glioma grade classification using gaussian convolutional neural network. *EEE Access*, 2022.
- [48] Jérémy Robert. Pandas : la bibliothèque python dédiée à la data science. *DataScientest*, janvier 2022.
- [49] Anne-Christel Rolling. Les tumeurs du cerveau. *INSTITUT NATIONAL DU CANCER*, 2023.
- [50] Futura Sciences. Cerveau. *Magazine Français*, 2014.
- [51] La Ryax Team. Comment le deep learning a bouleversé le domaine de l'intelligence artificielle?, 2023. Consulté le 17 septembre 2024.
- [52] Par Équipe Blent Data Scientist. Réseaux convolutifs (cnn) : comment ça marche? *blent.ai/blog*, Publié le 21 juin 2022 Catégorie Machine Learning.