

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد- تلمسان

Université Aboubakr Belkaïd- Tlemcen –

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER**

En : **Génie Biomédical**

Spécialité : **Imagerie Médicale**

Par : **ELEZAAR Roufaïda Et FAKHAR HadjHammou**

Sujet

**Un réseau neuronal convolutif pour la segmentation
d'images IRM Cérébrale**

Soutenu publiquement le : 25/06/2023, devant le jury composé de :

Mr. Boukli Hacen Ismail	MCA	Univ Tlemcen	Président
Mr. Hadj Ahmed Ismail	MAB	Univ Tlemcen	Examineur
Mm.KaziTani Lamia Fatiha	MCA	Univ Tlemcen	Encadrante

Année Universitaire :2022-2023

Remerciement

Nous remercions tout d'abord **Allah** le tout-puissant de nous avoir donnés le courage, la patience et sur tout la Volonté de mener à bien réaliser ce travail.

Nous remercions avec appréciation notre encadreur et prof **Mm. KAZI TANI Lamia Fatiha**, pour sa personnalité honnête qui nous fait apprendre beaucoup et sa confiance vers nous a propos de ce projet, ses encouragements et conseils, qui nous a donné l'esprit pour rester toujours motivés.

Nos sincères remerciements à M. Boukli Hacem Ismail et M. Hadj Ahmed Ismail, qui ont accepté de faire partie du jury et de consulter nos travaux. Nous les remercions pour leurs corrections, commentaires et critiques.

Je remercie ma chère mère qui m'a soutenu et m'a appuyé moralement.

Enfin, nous tenons à remercier tous ceux qui ont contribué directement ou indirectement à la réalisation de ce travail.

Dédicaces

*Avec un grand cœur plein de chaleur, je dédie ce mémoire, Aux êtres qui me sont
les plus Chers :*

*A mon cher père en signe de ma profonde reconnaissance pour sa veille sur
mon éducation, ses encouragements et pour m'avoir permis de réaliser mes études
dans les meilleures conditions.*

*A ma chère mère, la source d'amour qui me soutient dans toutes les années
scolaires et universitaires toutes les 17 ans et surtout m'encourage tout au long de
ce travail.*

A mes sœurs Khouloud et Rahaf

*A mes tantes et surtout ma chère tante Sara et mon oncle AEK qui m'ont beaucoup
aidé dans la période de ma maladie, je les remercie pour leur soutien*

A mes chers amis Sifou, Wafaa et Hichem

Roufaïda

Dédicaces

J'exprime tout d'abord mes remerciements à Dieu Tout-Puissant pour m'avoir accordé le succès pour accomplir cet humble travail que je dédie

- A mes chers parents qui ont consacré leur vie à mon éducation et qui ont tout sacrifié pour moi et ont tracé mon chemin avec leur soutien, que Dieu Tout-Puissant les protège

- A toute ma famille, avec qui je partage les bons et difficiles moments de ma vie. Pour son soutien moral tout au long de mon travail

- Mes frères et sœurs : Ibtissam, Aicha, Salman, A mes frères et amis qui ont partagé avec moi les moments de doute et tous les moments d'espoir dans la longue de ces cinq années

Ma future femme

À tous ceux que j'aime et j'ai oublié de citer

A tous ceux qui ont contribué de près ou de loin dans ma réussite

Hadj Hammou.

Résumé :

L'imagerie par résonance magnétique (IRM) est une technique en médecine permettant la localisation précise des tissus et des lésions invisibles dans différents plans spatiaux. Notre travail implémente un système d'aide au diagnostic médical utilisant des réseaux de neurones d'architectures variées pour localiser et détecter des tumeurs cérébrales en imagerie par résonance magnétique (IRM) en nous basant sur l'apprentissage profond. Une base de données contenant 256 images a été explorée. Nous avons ensuite utilisé les modèles VGG16, RESNET50 et VGG19 pour la classification et CNN pour la détection. Des résultats d'apprentissage, de test et de validation satisfaisants ont été ainsi obtenus.

Mots clés : Détection des tumeurs cérébrales ; Deep Learning ; Classification d'IRM ; CNN ; ResNet ; VGG ; Intelligence Artificielle ; Imagerie Médicale ; cloud ; Segmentation Images IRM cérébrales.

Abstract:

Magnetic resonance imaging (MRI) is one of the newest medical imaging techniques that allows the precise localization of tissue structures and invisible lesions in various spatial planes. Our work implements a medical diagnostic aided system using neural network so various architectures to detect brain tumors in magnetic resonance imaging (MRI) by "deep learning". A database containing 256 images. We then used the VGG 16, VGG19 and RESNET 50 networks for classification and CNN for detection. Satisfactory learning, testing and validation results have been achieved.

Keywords: Brain Tumor Detection; Deep Learning; Transfer learning; MRI classification; CNN; ResNet; VGG; Artificial intelligence; Medical imaging; Segmentation, Brain MRI images.

ملخص

التصوير بالرنين المغناطيسي (MRI) هو أحد أحدث تقنيات التصوير الطبي التي تسمح بالتوطين الدقيق لهياكل الأنسجة والآفات غير المرئية في مختلف المستويات المكانية. ينفذ هذا المشروع نظامًا طبيًا مساعدًا للتشخيص باستخدام شبكات عصبية منبني مختلفة للكشف عن أورام الدماغ في التصوير بالرنين المغناطيسي (MRI) عن طريق "التعلم العميق". قاعدة بيانات تحتوي على 256 صورة. ثم استخدمنا شبكات VGG 16 و RESNET 50 و VGG19 للتصنيف و CNN للكشف. تم تحقيق نتائج مرضية في التعلم والاختبار والتحقق.

الكلمات الرئيسية: تشريح الدماغ ، اكتشاف أورام الدماغ ؛ التعلم العميق ؛ نقل التعلم ؛ تصنيف صور الرنين المغناطيسي ؛ زيادة البيانات ؛ الذكاء الاصطناعي ؛ التصوير الطبي.

Tables des matières

<i>Remerciment</i>	I
<i>Dédicaces</i>	I
Résumé :	V
<i>Tables des matières</i>	IX
<i>Liste des figures :</i>	XI
<i>Liste des tableaux :</i>	X
Listes des acronymes et abréviations.....	XIII
<i>Introduction générale :</i>	2
Chapitre I:	
I.1 Introduction :	Error! Bookmark not defined.
I.2 Le cerveau :	5
I.2.1 Le cerveau droit et gauche :	7
I.2.2 Les lobes du cerveau :	7
I.2.3 Cortex :	8
I.2.4 La structure profonde :	9
I.2.5 Le crâne :	9
I.2.6 Les nerfs crâniens :	11
I.2.7 Les cellules de cerveau :	12
I.2.8 Les cellules de la glie :	13
I.3 Les Tumeurs du cerveau :	14
I.3.1 Les symptômes :	14
I.3.2 Les facteurs à risque :	15
I.4 Imagerie par résonance magnétique (IRM) :	16
I.4.1 Le fonctionnement d'une IRM :	16
I.4.2 Les différents IRM :	16
I.4.3 Formation d'IRM :	17
I.4.4 Avantage d'IRM :	18
I.5 Conclusion :	19

Chapitre II:

II.1 Introduction:	21
II.2 Définition de segmentation:	21
II.3 Objectifs de la segmentation :	21
II.4 Filtrage des images medical :	21
II.5 Les méthodes de segmentation	23
II.5.1 Approche par détection de contours:	23
II.5.1.1 Méthodes dérivatives	23
II.5.2 Approche par régions	24
II.5.2.1 Segmentation par seuillage	25
II.5.2.2 Croissance de région	26
II.5.2.3 Méthode de fusion-diffusion	26
II.5.3 Approche par classification de pixels:	27
II.5.3.1 principe de classification de pixels	28
II.5.3.2 Classification non Supervisée	29
II.5.3.3 Classification supervisée	29
II.5.3.3.1 Les support vector machine (SVM):	30
II.5.3.3.2 Les réseaux de neurones convolutifs (CNN):	31
II.5.3.3.2.2 Architecteur de réseaux de neurones convolutifs	36
II.5.3.3.2.3 Avantages de réseaux de neurones convolutifs :	37
II.5.3.3.2.4 Inconvénients de réseaux de neurones convolutifs :	37
II.6 conclusion :	38

Chapitre III:

III.1 Introduction :	40
III.2 L'environnement de travail	40
III.2.1 Hardware :	40
III.2.2 Software :	40
III .2.2.1 Langage de programmation :	40
III .2.2.2 visual studio code:	41
III.3 Base de Données :	41
III .4 Augmentation d'images (Data Augmentation):	42

III .5	Le Transfer learning vs. le Deep Learning:	43
III .6	Structures modifiées des modèles utilisés :	44
III .6.1	Le modèle VGG 16 :	44
•	VGG16	44
•	VGG19	45
III.6.2	Le modele ResNet 50:	46
III .7	Implémentation générale des réseaux de classification:	47
III .7.1	Préparer l’environnement:	47
III .7.2	Les bibliothèques utilisées:	48
III .7.3	Créer les dossiers de la base de données:	49
III.7.4	Diviser la base de données et faire le pre-processing:	49
III .7.5	Visualisation de la base de données :	51
III .7.6	Le prétraitement de la base de données et les réseaux de neurones :	52
III .7.7	L’augmentation des données :	55
III .7.8	Appliquer l’augmentation de données pour toute la base de données	58
III .7.9	Appliquer le Transfer Learning :	59
III .8	L’entrainement des réseaux de classification:	62
III .8.1	Résultats et discussion:	65
III.8.2	La phase de test	67
III .8.2.1	Enregistrer le modèle:	67
III .8.3	Entrez le chemin d'accès à l'image :	67
III .8.3.1	Le résultatde détection avec le nouveau model :	68
III .9	Conclusion	69
	Conclusion	71
	Références bibliographiques	74

Liste des figures :

Figure I. 1 : Les composants du cerveau.....	5
Figure I. 2 :un plan transversal représentant la substance grise et blanche	6
Figure I. 3 : les lobes de cerveau [5].....	8
Figure I. 4 :Le cortex cérébrale.....	9
Figure I. 5 :les os de crane	10
Figure I. 6 :les nerfs craniens.....	11
Figure I. 7 :Le scanner du cerveau.....	16
Figure I. 8 :IRM du cerveau.....	16
Figure I. 9 :Signal FID recueilli par les bobines réceptrices.	18
Figure II. 1 : Illustration de la détection de contours par une méthode dérivative[1]	24
Figure II. 2 :Exemples d’histogrammes[1]	26
Figure II. 3 :La segmentation par division-Fusion[1]	27
Figure II. 4 : principe de classification[1].....	29
Figure II. 5 :Structure d’un réseau de neurones convolutifs[2].....	33
Figure II. 6 :Opération de convolution[3].....	34
Figure II. 7 :Architecteur de CNN [3]	37
Figure III. 1 : Le logo de langage de programmation Python.....	41
Figure III. 2 :Le logo de programme de visual studio code.....	41
Figure III. 3 :deux échantillons de notre base représentantes chaque classe de la maladie	42
Figure III. 4 :étapes d’obtention d’une nouvelle base de données à partir de la base de données originale	43
Figure III. 5 :la différence entre le transfer learning et le machine learning.....	43
Figure III. 6 :Visualisation de l’architecture VGG[4]	44
Figure III. 7 :Le réseau de neurones VGG16 [3].....	44
Figure III. 8 :Les réseaux convolutionnels très profonds pour la reconnaissance d’images à grandeéchelle [3].....	45
Figure III. 9 :Architecture du modèle ResNet 50 [6].....	46
Figure III. 10 :Les étapes d’implémentation des réseaux de classification.....	46
Figure III. 11 :code de préparation des réseaux et de l’environnement	47
Figure III. 12 :Les bibliothèques utilisées	47
Figure III. 13 :code d’importation des bibliothèques et les paramètres d’entraînement	48
Figure III. 14 :Division de la base de données	48
Figure III. 15 :code de division de la base de données	49
Figure III. 16 :Lecture des chemins de la base de données.....	49
Figure III. 17 :Chargement de la base de données.....	50
Figure III. 18 :code de préparation de la visualisation de la base de données	50
Figure III. 19 :Visualisation de quelques échantillons de notre base de données avec leurs classes	51
Figure III. 20 :code de Preprocessing de la base de données.....	52
Figure III. 21 :code de Preprocessing de la base de données.....	52
Figure III. 22 :Définir les nouveaux dossiers et chemins virtuels de la base de données	53

Figure III. 23 :code de préparation à visualiser nos nouvelles images	53
Figure III. 24 :Visualisation de nos nouvelles images classifiées.....	54
Figure III. 25 :Schéma synoptique du principe de la technique d'augmentation des données.....	55
Figure III. 26 :code d'implémentation de l'augmentation de données	56
Figure III. 27 :Exemple d'une image ayant subit l'augmentation de données.....	56
Figure III. 28 :code d'implémentation de l'augmentation de données pour toute la base	57
Figure III. 29 :code de génération de la base de données augmentée.	58
Figure III. 30 :Chargement des réseaux de neurones qui vont etre utilisés pour le Transfer Learning.....	59
Figure III. 31 :Visualisation du résumé du modèle.....	60
Figure III. 32 :Organigramme de l'optimiseur ADAM	61
Figure III. 33 :code de configuration de la fonction d'activation et l'optimiseur ADAM.....	61
Figure III. 34 :Visualisation de l'entrainement.....	62
Figure III. 35 :La courbe de perte (loss) et la courbe de précision (accuracy) du réseau VGG16	62
Figure III. 36 :La courbe de perte (loss) et la courbe de précision (accuracy) du réseau VGG 19	63
Figure III. 37 :La courbe de perte (loss) et la courbe de précision (accuracy) du réseau Resnet 50.....	63
Figure III. 38 :code d'Enregistrer le modèle.....	64
Figure III. 39 :le code de détection avec le nouveau model	65
Figure III. 40 :Exemple d'image avec ou sans tumeur	65

Liste des tableaux :

Tableau I. 1 : Le nom et la fonction principale des douze nerfs crâniens11

Listes des acronymes et abréviations

IRM : Imagerie par Résonance Magnétique

TEP : Tomographie par Émission de Positons

TDM : Tomodensitométrie

TR : Temps de Répétition

LCR : Liquide céphalorachidien

CNN : Convolutional Neural Network

R-CNN : Region-based Convolutional Neural Network.

VGG : Visual Geometry Group

Conv : Convolution

ResNet : Residual Network

SVM : Support Vector Machine (Machine à vecteurs de support)

HOG : Histogram of Oriented Gradients (histogramme de gradient orienté)

GPU : Graphics Processing Unit (unité de traitement graphique).

CPU : Central Processing Unit (unité centrale de traitement).

VIA: VGG Image Annotator

VAL: Validation

Introduction générale

Introduction générale :

L'imagerie médicale amplement connue sous le nom de Radiologie est un axe en médecine où des personnes compétentes dans le domaine mettent en image des parties du corps humain dans le but de diagnostiquer chaque partie en détail et en profondeur. Ceci inclut des tests de diagnostic précoce de lésions. Cette technologie a impliqué plusieurs disciplines comme la physique qui a contribué grâce à de nouvelles technologies à la relance et le développement de ce domaine. Ceci offre un nouvel espoir aux malades pour l'identification exacte et précise des lésions, diminuant ainsi l'acte chirurgical, et facilitant les solutions thérapeutiques pour certaines maladies.

Les radiologues, à l'aide des techniques de l'IRM, et des différentes techniques de modalités d'observation anatomique (IRM, TDM, X-Ray...) et fonctionnelle (TEP, TEMP, IRM...) permettent aux chirurgiens de réaliser les interventions chirurgicales complexes et délicates, d'une manière moins invasive.

Le but de notre travail est de cibler des lésions voir des tumeurs cérébrales grâce à l'outil que nous proposerons au dernier chapitre. Cependant, Les tumeurs cérébrales sont plus fréquentes chez les hommes que chez les femmes. Néanmoins, les **méningiomes** sont généralement bénins et sont fréquents chez les femmes. Ce type de tumeur a l'opportunité de se développer à tout âge, le glioblastome est le type de tumeur cérébrale le plus grave, attaquant les personnes âgées. Les tumeurs cérébrales, cancéreuses ou non, causent réellement des soucis de santé du fait que le crâne est rigide, ce qui empêche la croissance de la tumeur. Ces problèmes peuvent se présenter comme une faiblesse, des difficultés à marcher, une perte d'équilibre, une perte partielle ou complète de la vision, des troubles de la compréhension ou de l'utilisation du langage, ainsi que des problèmes de mémoire.

D'un côté technique, la segmentation d'images est importante en traitement d'images, le but de notre travail est de mettre en place un outil d'aide au diagnostic de

tumeur cérébrales, il se base d'abords sur le detection été la reconnaissance de tumeur sur l'image pour qu'ensuite être capable de distinguer la présence ou non de la tumeur sur l'image. Les découpes cérébrales IRM des patients atteintes du gliome représentent environ un tiers des tumeurs cérébrales fréquentes, basé sur l'apprentissage profond (deeplearning), nootre but est d'aider les neuroradiologues à identifier les diverses formes des lésions du gliome.

Dans notre travail, nous utilisons trois modèles de réseau de neurones convolutifs CNN différents pour la segmentation des gliomes à partir des volumes IRM 3D, en essayant d'augmenter la précision des résultats en faisant varier les paramètres qui constituent le modèle.

Notre mémoire est divisé en trois chapitres, le premier présente le domaine médical de base de notre étude, le second présente les différentes techniques utilisées dans l'étude des tumeurs cérébrales, et le dernier chapitre explique en détail les différentes étapes avec la programmation respective de notre étude.

Chapitre I :

La tumeur cérébrale et l'imagerie

Par résonance magnétique

I.1 Introduction :

Selon qu'elle est bénigne ou maligne, une tumeur est une masse anormale qui se développe à la suite de la prolifération aberrante et le fonctionnement des cellules. Les tumeurs cérébrales étant particulièrement fréquentes, c'est l'une des maladies les plus répandues dans le monde.

Le contrôle musculaire, l'intellect, la créativité, la cognition, la mémoire et d'autres processus cruciaux sont tous contrôlés centralement par le cerveau. Le cerveau est constitué du cerveau, du cervelet et du tronc cérébral, qui sont enfermés dans le cadre protecteur du crâne (figure I.1).

Les cinq sens la vue, l'odorat, le toucher, le goût et l'ouïe travaillent souvent ensemble pour transmettre l'information au cerveau. Ces entrées sensorielles sont traitées par le cerveau, qui les transforme ensuite en signaux utiles et les stocke dans notre mémoire. Il contrôle notre pensée, la rétention de la mémoire, la parole, les mouvements volontaires des membres et le fonctionnement efficace de plusieurs fonctions corporelles [1].

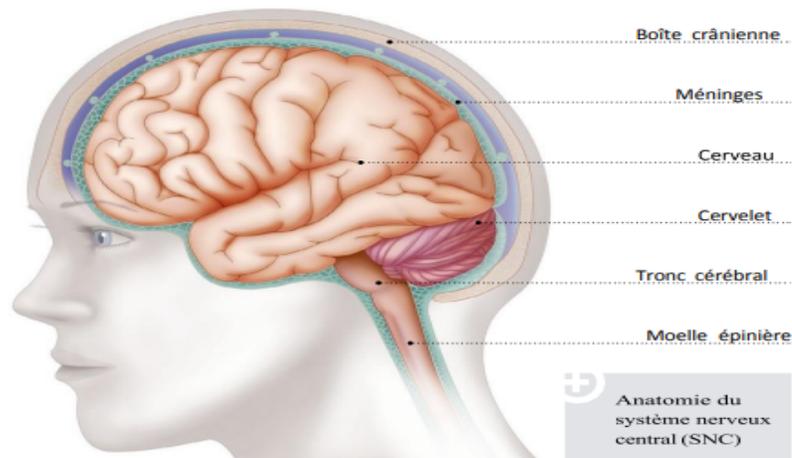


Figure I. 1 : Les composants du cerveau [1]

I.2 Le cerveau :

Est composé des éléments suivants :

- ✓ **Le cerveau :** La plus grande région du cerveau sert de chef d'orchestre pour les organes du corps, les conduisant comme un orchestre. C'est un organe qui donne un but à notre existence. L'hémisphère gauche du cerveau, qui est divisé en deux hémisphères, est responsable des processus cognitifs supérieurs, y compris la compréhension du toucher, de la vision et de l'ouïe. Il est également en charge de la communication, de la logique, des émotions, de l'apprentissage et de la régulation des mouvements intentionnels [4].
- ✓ **le cervelet :** Le cervelet, situé au-dessus et à l'arrière du cerveau, est crucial pour maintenir l'équilibre et la posture et pour réguler les mouvements musculaires. Le cervelet, composé de matière grise et de matière blanche [3] (Fig. I.2), apporte une contribution considérable à la précision et à la synchronisation des fonctions motrices.
 - ✓ **La matière grise** : Correspond au corps des cellules nerveuses, recouvre les hémisphères et le cervelet (cortex) et se trouve également disséminés sous forme de noyaux dans la substance blanche.
 - ✓ **La matière blanche** : Correspond aux fibres nerveuses (axones et dendrites recouvertes d'une gaine de myéline, est située au-dessous du cortex cérébral et autour des noyaux gris.

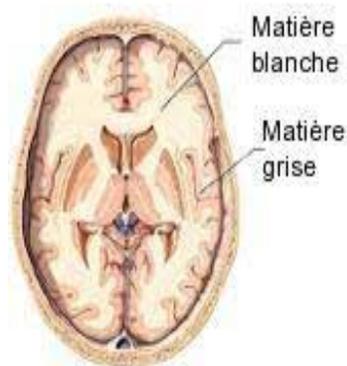


Figure I. 2 : un plan transversal représentant la substance grise et blanche [16]

- ✓ **Tronc cérébral :** Il agit comme un lien crucial entre le cerveau et le cervelet et la moelle épinière, orchestrant les processus clés, y compris la respiration, le contrôle de la température corporelle, le contrôle de la fréquence cardiaque, le maintien de la tension artérielle et l'aide du système digestif avec la transformation des aliments.

I.2.1 Le cerveau droit et gauche :

Les deux hémisphères du cerveau sont l'hémisphère droit et l'hémisphère gauche. Les messages sont transmis d'un côté à l'autre par un faisceau de fibres appelé corps calleux qui les relie. Le côté opposé du corps est occupé par chaque hémisphère du cerveau. Le bras ou la jambe gauche peuvent être faibles ou paralysés après un AVC sur le côté droit du cerveau. Les fonctions de chacun des hémisphères ne sont pas partagées.

La parole, la compréhension, l'arithmétique et l'écriture sont généralement occupées par l'hémisphère gauche du cerveau. La créativité, la capacité spatiale, les compétences artistiques et musicales jouent un rôle important dans l'hémisphère du cerveau. Environ 92 % des personnes utilisent les mains et le langage dans l'hémisphère gauche. Les fibres nerveuses du corps calleux relient les deux côtés.

I.2.2 Les lobes du cerveau :

Les fissures qui divisent le cerveau en lobes se trouvent dans les différents hémisphères cérébraux. Ils ont quatre lobes : le frontal, le temporel, le pariétal et l'occipital (figure 1.3.1). Il est important de se rappeler que chaque lobes du cerveau ne fonctionne pas seul. Les lobes du cerveau et les hémisphères droit et gauche sont liés de manière complexe.[5]

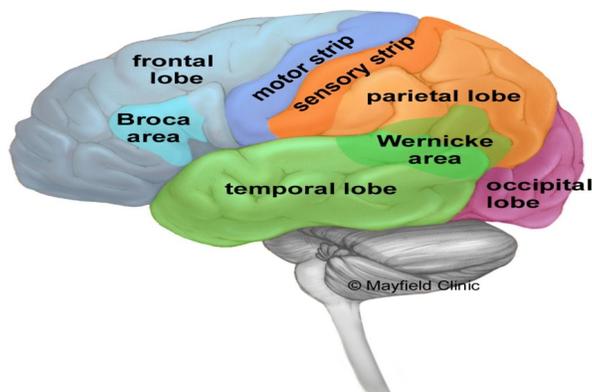


Figure I. 3 : leslobes de cerveau [5]

Les principales fonctions de chaque lobesont les suivantes :

- **Les lobesfrontaux** : contrôlent la parole, le langage, le raisonnement, la mémoire, la prise de décision, la personnalité, le jugement et les mouvements. Les mouvements du côté gauche du corps sont gérés par le lobefrontal, et inversement, les mouvements du côté droit sont gérés par le lobefrontal gauche.
- **Les lobes pariétaux** :contrôlent la lecture, le repérage dans l'espace, la sensibilité. Là aussi, le lobe pariétal droit gère la sensibilité du côté gauche du corps et réciproquement.
- **Les lobesoccipitaux** : contrôlent la vision.
- **Les lobestemporaux** :contrôlent le langage, la mémoireet les émotions.

I.2.3 Cortex :

Le cortex du cerveau est la surface de ce corps. 16 milliards de neurones sont présents sur le site, tandis que le cervelet en compte entre 70 et 86 milliards. Ils sont disposés sur des couches particulières. La matière grise est la couche gris-brune que les corps des cellules nerveuses colorent le cerveau. La matière blanche se trouve au dessous du cerveau, où se trouvent des longues fibres nerveuses (axones) qui relient les zones du cerveau entre elles. Le repli du cœur augmente la surface du cerveau, ce qui permet à un plus grand nombre de neurones de tenir à l'intérieur du crâne et d'assurer des fonctions supérieures. Chaque gyrus et chaque sulcus entre les plis sont des termes (figure.1.4).

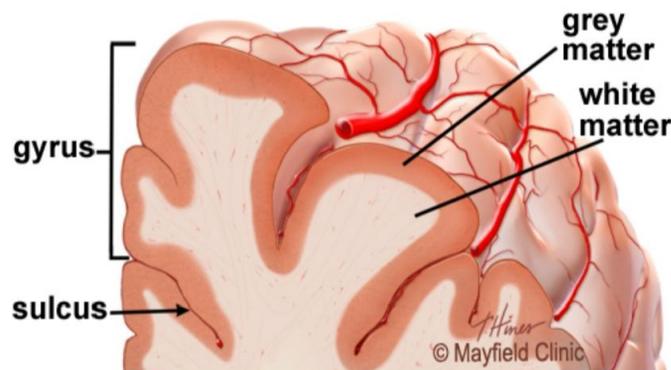


Figure I. 4 : Lecortex cérébrale [17]

I.2.4 La structure profonde :

- *Hypothalamus* : Il se trouve sur le plancher du troisième ventricule et sert de moteur du système autonome. Il est chargé de gérer les symptômes tels que la faim, la soif, le sommeil et l'activité sexuelle. De plus, il contrôle la température du corps, la pression sanguine, les émotions et la sécrétion d'hormones.
- *Hypophyse* : se trouve à la base du crâne dans une petite pièce appelée sellaturcica. La tige pituitaire relie l'hypophyse à l'hypothalamus du cerveau. Elle est connue sous le nom de "glande maîtresse" et remplit le rôle des autres glandes endocrines du corps. Elle aide à la croissance des os et des muscles, réduit le stress et sécrète des hormones qui entravent le développement sexuel.
- *Glande pinéale* : Elle se trouve après le troisième ventricule. En sécrétant de la mélatonine, elle aide à réguler l'horloge interne et les rythmes circadiens de la corp. Elle a un rôle particulier dans le développement sexuel.
- *Thalamus* : sert de relais pour presque toutes les informations qui font des allers-retours vers le cortex. Il joue un rôle dans la sensation de la douleur, l'attention, la vigilance et la mémoire.
- Les ganglions de la base : comprennent le caudate, le putamen et le globus pallidus. Ces noyaux travaillent avec le cervelet pour coordonner les mouvements fins, tels que les mouvements du bout des doigts.
- *Système limbique* : est le centre de nos émotions, de l'apprentissage et de la mémoire. Ce système comprend les gyres cingulaires, l'hypothalamus, l'amygdale (réactions émotionnelles) et l'hippocampe (mémoire).[7]

I.2.5 Le crâne :

Le crâne osseux, composé de huit os qui se combinent dans le long des lignes de suture, protège le cerveau des blessures. Certains d'entre eux s'occupent du front, des pariétaux, des temporaux, des sphénoïdes, des sphénoïdes occipitaux, des sphénoïdes et des sphénoïdes. Il y a 14 os appariés sur la face : le maxillaire, le

zygmatique, le nasal, le palatin, le lacrymal, les cornets nasaux inférieurs, la mandibule et le vomer.[6]

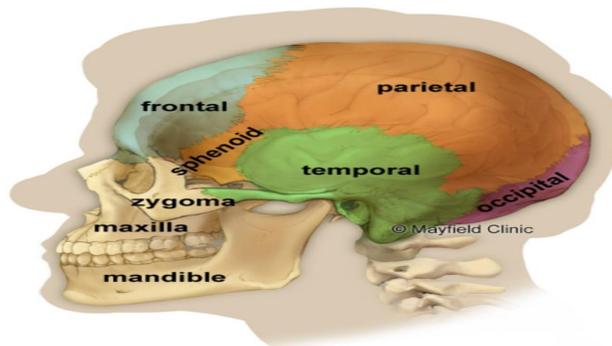


Figure I. 5 : lésions de crâne [18]

Trois zones distinctes se trouvent à l'intérieur du crâne : la fosse antérieure, la fosse moyenne et la fosse postérieure. Les nerfs crâniens partent du tronc cérébral, se déplacent vers les parties du crâne qu'ils innervent à travers des trous appelés foramina. Les fosses antérieure, moyenne et postérieure sont les trois régions de la base du crâne.

Les médecins font référence à l'emplacement d'une tumeur en s'appuyant sur ces termes, par exemple, méningiome fosse moyenne.

La foramina est un tissu qui relie toutes les artères, les veines et les nerfs au bas du crâne. Aussi le grand trou au milieu (foramen magnum) qui comprend la moelle épinière.

I.2.6 Les nerfs crâniens :

La moelle épinière et douze paires de nerfs crâniens permettent au cerveau de communiquer avec le cœur. (Figure.1.6). Les dix paires de nerfs crâniens qui affectent l'audition, les mouvements des yeux, les sensations faciales, le goût, les mouvements du visage, du cou, des épaules et de la langue ont leur origine dans le cerveau. Le cerveau est responsable des nerfs crâniens du thorax et de la vision [10].

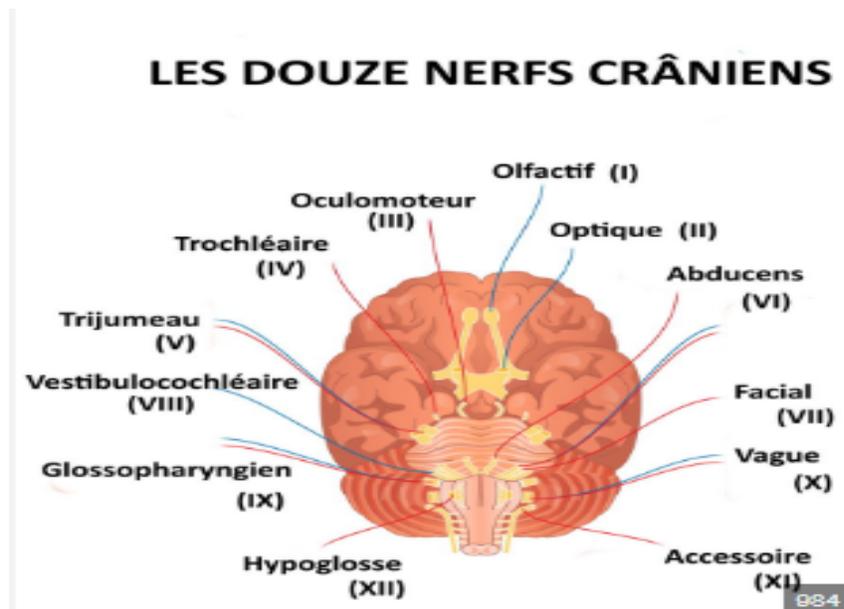


Figure I. 6 : les nerfs crâniens[19]

Tableau I. 1 : Le nom et la fonction principale des douze nerfs crâniens

N°	Nom	Fonction
I	Olfactif	odeur
II	Optique	La vue
III	Oculomoteur	déplace l'œil, la pupille
IV	Trochléaire	déplace l'oeil

V	Trijumeau	sensation du visage
VI	Abducens	déplace l'oeil
VII	Visage	bouge le visage, salive
VIII	Vestibulocochléaire	audition et équilibre
IX	Glossopharyngeal	goût, déglutition
X	Vagus	rythme cardiaque, digestion
XI	Accessoire	bouge la tête
XII	Hypoglosse	déplace la langue

I.2.7 Les cellules de cerveau :

Le cerveau est composé de deux types de cellules : les cellules nerveuses (neurones) et les cellules gliales.

I.2.7.1 La cellule nerveuse : Le neurone est une cellule comme une cellule du cœur, mais ils n'ont pas la même fonction car le neurone est principalement chargé de transmettre des informations sûres à long terme. Les informations peuvent être redirigées dans différentes zones corticales en série ou en parallèle grâce à un réseau complexe de connexions créé par les neurones. Les informations sont transmises de deux manières : électriquement à l'intérieur du neurone et, plus précisément, chimiquement d'un neurone à l'autre.

I.2.7.2 Les neurones Il existe plusieurs types :

➤ **Distinction selon le nombre de neurites :**

- ✓ 1 neurite : neurone unipolaire
- ✓ 2 neurites : neurone bipolaire
- ✓ N neurites : neurone multipolaire.

➤ **Distinction anatomique (3 exemples) :**

- ✓ Les cellules pyramidales se trouvent dans la matière grise, ou cortex cérébral.
- ✓ Les neurones en étoiles se trouvent également dans le cortex cérébral.

- ✓ Les cellules de Purkinje sont des cellules aux ramifications très nombreuses que l'on retrouve exclusivement dans le cervelet.
- **Distinction selon la fonction :**
 - ✓ Les neurones sensoriels sont directement reliés aux organes des sens et sont responsables de faire transiter l'information sensorielle (connues aussi comme « information ascendante ») vers le cerveau.
 - ✓ Les motoneurones ou neurones moteurs se chargent de faire transiter les ordres provenant du cerveau (connues aussi comme « information descendante ») en direction des muscles.
 - ✓ Les interneurones sont représentés par tous les neurones qui ne sont ni sensoriels ni moteurs mais qui font la jonction entre ces deux types de neurones.[9]

I.2.8 Les cellules de la glie :

Les cellules du cerveau qui fournissent aux neurones de la nourriture, les protègent et leur fournissent un soutien structurel sont appelées cellules de la glie, un terme qui vient du mot grec "cellule". Les cellules gliales sont dix à cinquante fois plus nombreuses que les cellules nerveuses. C'est le type de cellules qui est le plus présent dans le cerveau.

- Les astroglies ou astrocytes sont les gardiens. Ils contrôlent l'homéostasie, la défense et la réparation des neurones, la formation des cicatrices, et affectent les impulsions électriques.
- Les cellules oligodendrogiales créent une substance grasse appelée myéline, qui isole les axones, permettant ainsi les messages électriques de voyager plus rapidement.
- Les cellules épendymaires tapissent les ventricules et sécrètent le liquide céphalo-rachidien (LCR).
- Les microglies sont les cellules immunitaires du cerveau, le protégeant des envahisseurs et nettoyant les débris. Elles élaguent également les synapses.

I.3 Les Tumeurs du cerveau :

Les tumeurs cérébrales sont des masses de cellules qui progressent rapidement, de façon incontrôlable. Elles sont de deux types :

- Les tumeurs primitives : Les tumeurs bénignes, qui représentent deux tiers des cas de tumeurs primitives, sont constituées de cellules non cancéreuses, et les tumeurs malignes, qui se sont transformées en cellules cancéreuses. Les tumeurs primaires, qu'elles soient bénignes ou malignes, peuvent provoquer des symptômes graves, en particulier lorsqu'elles se trouvent dans les zones fonctionnelles clés du cerveau.
- Les tumeurs secondaires : sont des métastases d'un cancer préexistant. Elles naissent lorsque des cellules d'une tumeur initiale située hors de l'encéphale parviennent à migrer, via la circulation sanguine, vers le tissu cérébral. [8]

I.3.1 Les symptômes :

La pression exercée sur les tissus cérébraux provoque les symptômes d'une tumeur cérébrale. Ils sont plus dépendants de la localisation et de la rapidité de croissance de la tumeur que de sa nature. En général, la tumeur peut être localisée en analysant les troubles fonctionnels subis.[12]

- Les maux de tête sont le symptôme initial le plus fréquent des patients atteints de tumeurs primaires. Ils résultent principalement d'une croissance tumorale qui affecte les tissus environnants.
- Les nausées et vomissements dus à la pression dans la boîte crânienne.
- Les vertiges et troubles de l'équilibre.
- L'effet de masse peut indiquer des signes de tumeur cérébrale en fonction de l'emplacement de la tumeur dans le cerveau. Par exemple, les tumeurs de la région lombaire peuvent entraîner des changements de corps, tandis que les tumeurs de la région lombaire dominante peuvent entraîner des problèmes d'électricité réceptive.
- La perte d'appétit

- Les crises d'épilepsie
- Les changements d'humeur, de comportement et de personnalité.
- Les hallucinations visuelles, auditives ou olfactives
- La sensation de la faiblesse, des difficultés à marcher

I.3.2 Les facteurs à risque :

Certains facteurs peuvent augmenter le risque d'atteinte d'une tumeur cérébrale. Parmi ces facteurs de risque :[12]

- ✓ L'âge.
- ✓ L'exposition aux radiations.
- ✓ Les antécédents familiaux

I.3.3 Le diagnostic :

Le diagnostic d'une tumeur du cerveau se fait en plusieurs étapes :

- Un examen clinique et neurologique complet.
- Des examens d'imagerie (Scanner, IRM...)
- Un examen anatomopathologique, qui consiste à analyser un échantillon de tumeur. [11]



Figure I. 7 : Le scanner du cerveau [11]

Figure I. 8 : IRM du cerveau [11]

I.4 Imagerie par résonance magnétique (IRM) :

La technique de l'imagerie par résonance magnétique (IRM) a été introduite en 1973 par Lauterbur et Damadian. C'est une technique d'imagerie médicale récemment développée, non invasive et sans effets secondaires qui permet d'obtenir une imagerie 2D ou 3D d'une partie du corps, en particulier du cerveau. Elle aide au diagnostic du cancer, à l'évaluation de l'efficacité d'un traitement et/ou au suivi après le traitement. Grâce à l'utilisation d'un aimant et d'ondes électromagnétiques, une IRM permet d'obtenir des images précises de l'intérieur du corbeau. Vous êtes allongé sur un lit (ou table d'examen) qui se déplace à l'intérieur du tunnel pendant l'examen, et la zone de trois corbeaux est examinée au centre de l'aimant. Souvent, il est nécessaire d'injecter un produit appelé « de contrace » dans la circulation sanguine pour améliorer la visibilité de certains organes et la détection d'anomalies.[14]

I.4.1 Le fontionnement d'une IRM :

L'imagerie par résonance magnétique (IRM) utilise un aimant très puissant qui fait « vibrer » les atomes d'hydrogène, éléments qui composent les molécules d'eau des notrecorps. Plus de 65 % de l'organisme est composé d'eau. Chaque atome d'hydrogène contient une particule encore plus petite appelée proton, qui se trouve à l'origine du phénomène. Les protèges « vibrent » à différentes vitesses en fonction du type de tissu dans lequel ils se trouvent, et une antenne sera placée à côté de la région à examiner pour les capter. Cela permet de bien différencier les différentes structures de l'organisme sur les images présentées. Les signaux reçoivent des millions de protons dans le corpssont pour créer des images en coupes détaillées de l'intérieur du corps. Un ordinateur reconstruit ensuite des images en 2 ou 3 dimensions avant qu'un radiologue les interprète.[14]

I.4.2 Les différents IRM :

Il existe deux techniques pour l'acquisition de l'IRM :

- IRM anatomique : En médecine, les noyaux d'hydrogène, un élément présent en abondance dans l'eau et les graisses des tissus biologiques, sont fréquemment utilisés pour la résonance magnétique nucléaire (RMN). La visualisation de la structure anatomique est appelée IRM anatomique.
- IRM fonctionnelle : Il est maintenant possible de produire des images RMN en des temps suffisamment rapides pour suivre certains aspects du métabolisme grâce au développement de techniques d'acquisition et de traitement de données ultrarapides. On parle également de l'IRM fonctionnelle. En imagerie cérébrale, la résonance magnétique de l'hémoglobine modifie légèrement les caractéristiques magnétiques de la molécule lorsqu'elle est liée à l'oxygène. En créant des images qui montrent le contraste entre les régimes riches en oxygène et les régimes du flux sanguin, on peut accéder à l'activité cérébrale.[14]

I.4.3 Formation d'IRM :

Le patient est exposé à un champ magnétique initial B_0 avant d'être exposé à un champ magnétique onde électromagnétique B_1 . Les atomes d'hydrogène entrent en résonance avec l'onde de cette dernière. Les atomes d'hydrogène commencent leur relaxation au terme de la phase d'excitation. Lorsqu'ils arrivent à l'équilibre, ils émettent un champ connu sous le nom de « décharge d'induction libre ».

Les capteurs de la machine recevront ce signal et le transformeront en un signal électrique qui sera analysé et transformé en une image numérique. Il est important de savoir distinguer les signaux provenant des différentes zones examinées lors de l'utilisation de la technologie de l'imagerie. Le couplage de trois gradients différents permet de réaliser ce couplage :

- Le gradient de coupe sélective sélectionne le plan de coupe, celui-ci pouvant être quelconque dans l'espace,
- Le gradient de codage de phase sélectionne les lignes dans le plan de coupe sélectionné,

- Le gradient de fréquence sélectionne les colonnes dans le plan de coupe sélectionné.

L'orientation des coupes, qui dépend en particulier du gradient de coupe sélective, varie en fonction de la pathologie recherchée. On retrouve

Cependant trois orientations principales : axiale, coronale et sagittale. Nous effectuons notre travail sur des images de type axial.[13]

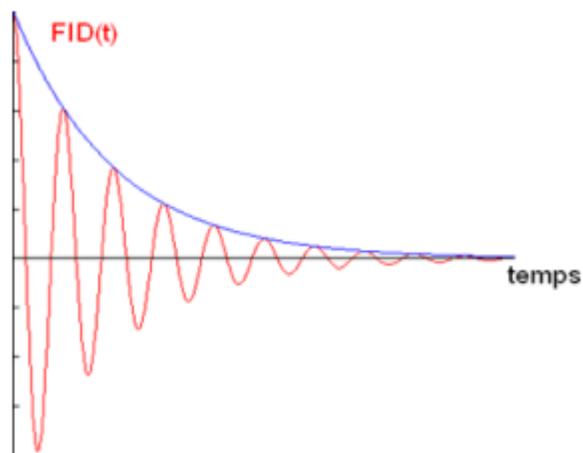


Figure I. 9 : Signal FID recueilli par les bobines réceptrices.[47]

I.4.4 Avantage d'IRM :

Grâce à l'utilisation de l'IRM, l'étude du cerveau a connu un développement impressionnant ces dernières années. Le développement permettra de protéger le cerveau et ses maladies à une échelle plus précise et représentative des phénomènes qui l'animent. L'une des méthodes les plus courantes et les plus efficaces pour diagnostiquer le cerveau d'un patient est cette méthode. Ce succès découle de plusieurs facteurs :

- Un contraste excellent pour les tissus mous grâce aux temps de relaxation T1 et T2 et la densité des spins.
- Versatilité dans la modification du signal avec des agents de contraste.
- Bonne résolution spatiale (de l'ordre de 1 mm) et temporelle (de l'ordre de 50 ms pour une seule coupe à quelques minutes pour les images 3D).

- L'examen IRM peut durer jusqu'à une heure, mais La précision est plus élevée que celle du scanner [15]

I.5 Conclusion :

Dans ce chapitre, les cellules gliales sont utilisées pour décrire brièvement l'anatomie du cerveau. Ensuite, nous avons défini et fourni une mine d'informations en utilisant les principes de l'IRM pour définir les tumeurs cérébrales à l'étude.

Le sujet de notre recherche est la détection automatique des tumeurs cérébrales, dont nous parlerons dans le chapitre suivant...

Chapitre II:

Les techniques de segmentation

II.1 Introduction :

Les techniques de segmentation des images par classification La discipline de la classification des données multidimensionnelles a été développée, et nombreuses techniques ont été ajustées pour essayer de trouver celles qui fonctionnent le mieux pour segmenter les images. Dans cette situation, la classification permet de diviser les pixels en groupes clairement définis et de regrouper des pixels ayant des caractéristiques similaires. Qu'elle soit supervisée ou non, la classification est depuis longtemps une méthode populaire dans le domaine de l'analyse et du traitement d'images numériques, que ce soit pour des raisons de segmentation ou d'interprétation.

Les réseaux neuronaux convolutifs (CNN) et le support de la machine vectorielle (SVM) sont le cœur de ce chapitre. Le potentiel pour CNN et SVM pour révolutionner l'imagerie médicale et améliorer les résultats des patients est énorme. Dans ce chapitre, nous présentons quelques idées théoriques et les techniques qui ont été utilisées pour obtenir les résultats.

II.2 Définition de segmentation :

Il est difficile de trouver une définition unique de la segmentation car elle est souvent confondue avec la classification ou l'étiquetage, cependant nous essayons de donner une définition pour lever l'ambiguïté.

La segmentation est l'extraction d'indices visuels à partir d'images. Où il permet de diviser l'image en un groupe de points appelés régions, homogènes dans une ou plusieurs propriétés (densité, couleur, texture, etc.) et différentes de régions voisines dans au moins une de leurs propriétés formellement.

Définition : La segmentation définie par Horowitz et Pavlidis [20], consiste à considérer un domaine Ω de l'image I et une fonction f qui associe à chaque pixel une valeur $f(i, j)$. On définit alors le prédicat p sur l'ensemble des parties de Ω qui permet de tester l'homogénéité des ensembles disjoints R_i .

La segmentation de l'image I est définie comme une partition de Ω en n sous-ensembles $\{R_1, R_2\}$ tel que :

$$1/ I = \bigcup_{i=1}^n R_i.$$

2/ $\forall i \in \{1, \dots, n\} p(R_i)$ vraie.

3/ $\forall i, j \in \{1, \dots, n\}^2 R_i$ est adjacent à R_j et $i \neq j \implies p(R_i, R_j) = \text{Faux}$.

Une segmentation est donc une décomposition de l'image en sous-ensembles de régions homogènes, chaque région correspond à un objet de l'image.

II.3 Objectifs de la segmentation :

- Fournir des régions homogènes (selon un critère donné).
- Réduire le bruit.
- Etudier et Interpréter des structures anatomiques.
- Localiser de manière précise les contours des régions.

II.4 Filtrage des images medical :

Améliorer et restaurer des images médicales pour augmenter leur qualité et leur clarté est une technique connue sous le nom de filtrage d'images médicales. Il est appliqué pour réduire le bruit et les artefacts, améliorer le contraste des caractéristiques locales et mettre en évidence les bordures et les limites. Les radiologues, les ingénieurs et les médecins traitent les images médicales pour mieux comprendre l'anatomie d'individus ou de populations spécifiques. La première étape du traitement des images médicales consiste à extraire les données brutes des images CT ou IRM et à les reconstruire dans un format pouvant être utilisé par le logiciel nécessaire. Pour l'analyse d'images biologiques, de nouvelles techniques de filtrage, telles que les techniques de filtrage dépendant de l'image, ont été créées. La compréhension a priori du type de bruit entachant l'image est cruciale pour la construction de filtres. Le lissage réduit les artefacts bruyants avec des filtres tels que moyen, gaussien et Wiener, mais ce processus lisse également souvent les

bords. Aux fins du post-traitement des images médicales, diverses méthodes de filtrage d'image sont disponibles, telles que les filtres à réponse impulsionnelle finie, qui préservent les caractéristiques haute et basse fréquence tout en offrant le meilleur lissage possible de l'image bruitée.

II.5 Les méthodes de segmentation :

La segmentation est essentielle dans le traitement d'image. Ainsi de nombreux algorithmes ont été proposés au cours des dernières décennies [21]. Généralement Ils reposent sur des principes différents. Nous proposons de discuter, des différentes approches trouvées dans la littérature. Dans ce travail, nous nous limiterons aux trois catégories suivantes : les approches basées sur le contour, les approches basées sur la région et les approches basées sur la classification.

II.5.1 Approche par détection de contours :

La méthode de contour est l'une des méthodes les plus classiques de la division. Ceci est généralement effectué par l'application de masque sur l'image pour détecter les changements locaux dans la force des pixels [22]

On peut catégoriser ces méthodes en plusieurs classes :

- Les méthodes dérivatives (opérateurs de Sobel, Robert,...).
- Méthodes des contours actifs.
- Les méthodes morphologiques (Erosion, Dilatation...).
- Ces méthodes sont adaptées aux images qui présentent des régions uniformes au sens des niveaux de gris.

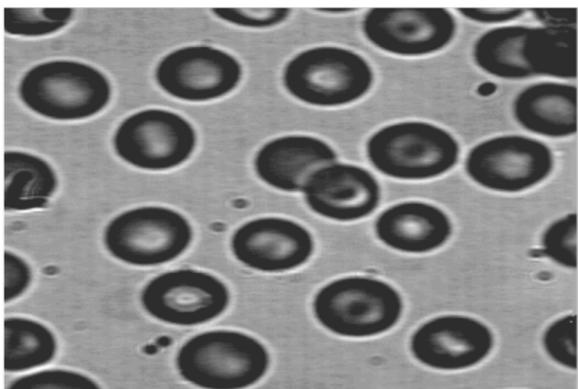
II.5.1.1 Méthodes dérivatives :

Ces méthodes facilitent la détection des ruptures dans une image et sont les plus couramment utilisées pour détecter ces transitions. Pour de meilleurs résultats, il est

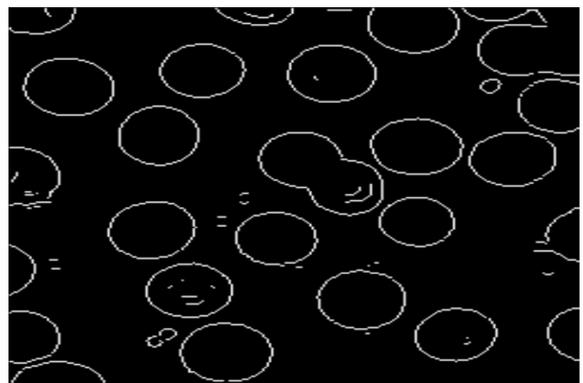
recommandé de lisser l'image avant le masquage, en raison de sa grande sensibilité au bruit.[22].

On classe les méthodes dérivatives en deux approches :

- Approche gradient : détermination des extrema locaux dans la direction de gradient.
- Approche laplacien : détermination des passages par zéro du Laplacien.



(a) Image originale



(b) contour détectés

Figure II. 1 : Illustration de la détection de contours par une méthode dérivative [23]

II.5.2 Approche par régions :

La segmentation consiste à faciliter l'interprétation automatique des images d'une manière qui ressemble à l'interprétation humaine. Historiquement influencé par le système de perception visuelle humaine, qui localise et délimite les choses dans une scène en utilisant les idées de ressemblance et de différence [23].

Le processus intermédiaire de segmentation des régions commence par une image originale ou filtrée et tente de reconstruire les régions qui correspondent aux objets visibles dans l'image. La segmentation transforme une image où chaque pixel est représenté par une étiquette correspondant à la région à laquelle il appartient. L'étape

d'analyse vise souvent à définir les régions à l'aide d'attributs afin d'identifier les éléments présents dans l'image une fois qu'ils ont été recréés avec succès.

II.5.2.1 Segmentation par seuillage :

La méthode de segmentation la plus populaire pour séparer les objets de l'arrière-plan d'une image est le seuil [24]. Chaque point de l'image donne une classe de luminosité par l'opérateur. Après avoir comparé les luminosités avec un ou plusieurs seuils, le seuil est complété.

Cette méthode fait une tentative de catégoriser les pixels en utilisant l'analyse d'histogramme unidimensionnel. Typiquement, les seuils sont établis en utilisant ces analyses localement ou globalement pour une image ou un volume [25]. Un seuil simple placé sur la vallée entre les deux modes peut être utilisé pour séparer les régions de l'arrière-plan dans une image composée d'une ou plusieurs zones du même type et d'un fond visuellement distinct lorsque l'histogramme est bimodal (figure II.2 (a)). L'histogramme de l'image est multimodal (figure II.2(b)) si elle comprend des régions avec différentes propriétés, chaque mode correspondant à une région distincte. Le problème de la localisation du minimum entre deux modes adjacents, ou les vallées, est alors décrit comme un seuil. Ensuite, la même région est donnée à tous les pixels avec des niveaux de gris entre deux vallées.

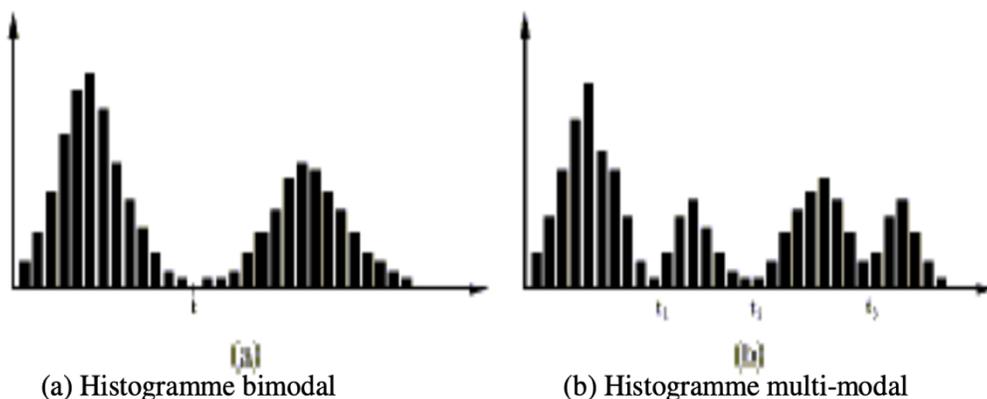


Figure II. 2 : Exemples d'histogrammes [23]

Il existe généralement deux approches distinctes pour choisir les seuils de l'histogramme. Alors que dans le second, les seuils sont dérivés localement à partir de minuscules fenêtres de mise au point, les limites dans le premier sont déterminées globalement pour tous les points de l'image [27].

II.5.2.2 Croissance de région :

Une stratégie connue sous le nom de méthodes de croissance régionale, parfois appelées techniques ascendantes, implique de regrouper les pixels en zones basées sur deux critères : homogénéité et adjacence [22]. Les zones environnantes du germe sont examinées l'une après l'autre ; si elles répondent aux critères d'agrégation, elles sont acceptées dans la région et deviennent ensuite un germe ; sinon elles sont rejetées. Les critères locaux basés sur les niveaux radiométriques des sites sont les plus utilisés comme critères. Lorsque tous les pixels entourant les régions touchées qui ont été affectés, la procédure se termine.

II.5.2.3 Méthode de fusion-diffusion :

Ces techniques ont toutes une chose en commun : elles commencent par une phase initiale, non homogène de l'image, qui est généralement l'image elle-même. Ils continuent à se diviser jusqu'à ce qu'ils atteignent des partitions homogènes [28].

Après le processus de division, il y a souvent encore quelques petits morceaux de zone déconnectés. Ceci est réalisé au cours de la phase de fusion, qui réunit les zones voisines. La fusion continue des zones basées sur les exigences d'homogénéité donne un nombre maximum de segments liés. Pour les images au niveau gris, un algorithme de « division de fusion » exécute cette procédure. Cet algorithme est divisé en deux parties [29] :

1. Chaque bloc (correspondant à un nœud de tétra-arbre) est évalué dans la première étape de telle sorte qu'il devrait être divisé en quatre sous-blocs, et cela se poursuit jusqu'à ce que chaque sub-bloc satisfasse un prédicat d'homogénéité.

2. Une fusion de blocsvoisins qui ont les mêmes propriétés colorimétriques est réalisée dans une étape ultérieure.

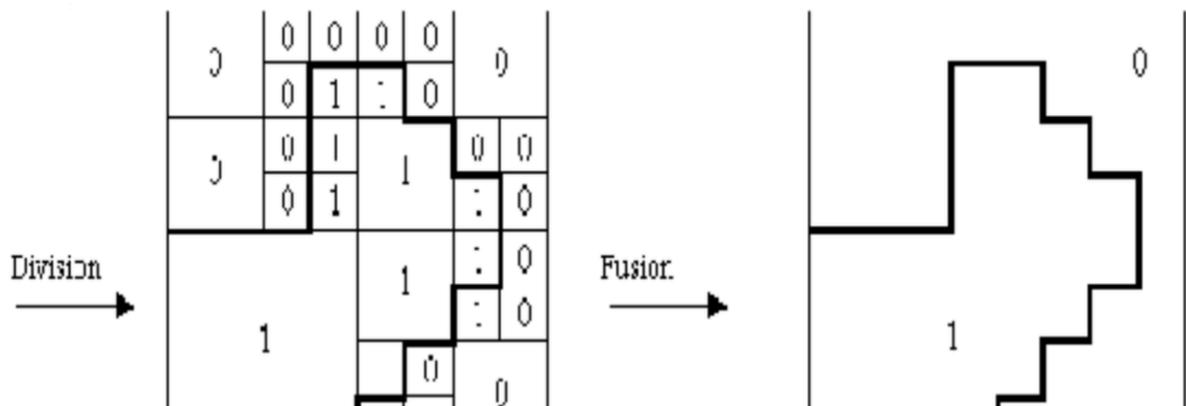


Figure II. 3 : Lasegmentation par division-Fusion [23]

II.5.3 Approche par classification de pixels :

Une étape avant la segmentation, la classification implique de classer les pixels de l'image en groupes en fonction de leurs propriétés colorimétriques comparables. Les propriétés pixel sont utilisées pour créer des classes. Les classes devraient être aussi différentes les unes des autres que c'est pratique [30]. Chaque pixel d'une image couleurou multispectrale sera classé dans une classe différente indépendamment de ses voisins et sera considéré comme un ensemble non trié de vecteur [31].

Les pixels dans une zone doivent appartenir à la même classe, bien qu'il puisse y avoir de nombreuses régions dans l'image qui le font. En extrayant les composants connexes et en

identifiant les pixels appartenant à la même classe, la segmentation est finalement terminée. Il s'agit d'une opération qui implique de donner la même étiquette (valeur intégrale) à chaque pixel d'un composant lié.

Étant donné qu'aucun des critères de segmentation énumérés dans l'introduction n'est strictement respecté, en particulier en ce qui concerne la règle de connexion, la classification est incapable de résoudre tout problème de segmentation.

II.5.3.1 principe de classification de pixels :

Les problèmes de classification impliquent de comprendre comment attribuer un objet à une classe. (Individu). Il y a souvent deux étapes clés dans la construction d'une méthode de classification qui classe automatiquement les choses :

- une phase d'apprentissage visant à décider d'un espace pour la représentation des choses et à rechercher les caractéristiques discriminatoires capables de définir chaque classe d'objets.
- une étape de reconnaissance où chaque élément inconnu dans l'espace de représentation déterminé par l'apprentissage donne une classe.

Selon Bezdek [32], la classification peut être largement divisée en deux types : la soi-disant classification supervisée et la non-supervisée. (Automatique).

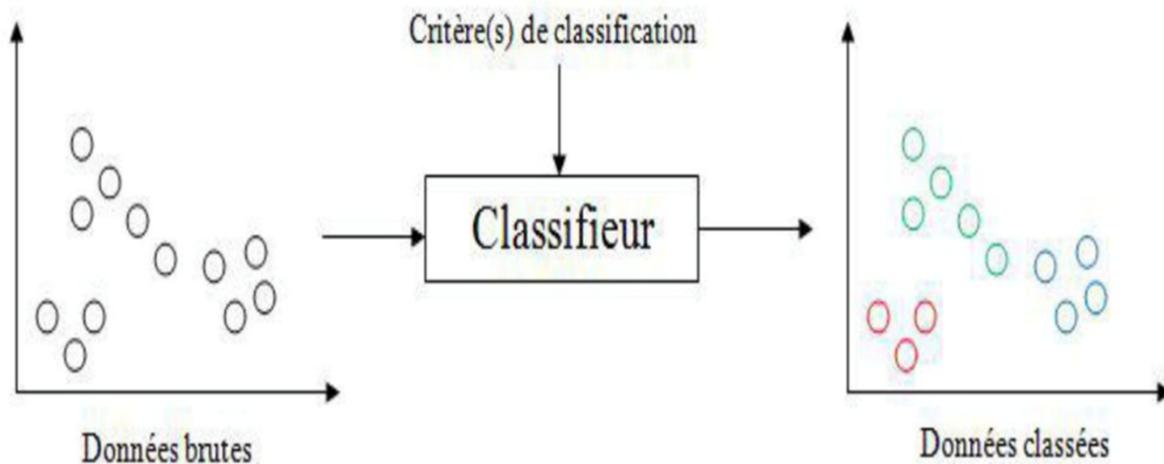


Figure II. 4 : principe de classification [23]

II.5.3.2 Classification non Supervisée :

Pour catégoriser les données, il faut d'abord extraire les caractéristiques qu'un système de classification emploierait. Trouver des partitions d'un groupe d'individus sur la base de critères de proximité à leurs vecteurs d'attribut dans l'espace de représentation est le processus de classification non supervisée, également connu sous le nom de classement pixel sans apprentissage. Ils sont employés pour effectuer une segmentation sans information préalable de l'image ainsi qu'une classification aveugle.

II.5.3.3 Classification supervisée :

La classification de l'image médicale peut être faite de plusieurs façons, telles que la classification floue, la Classification multi-modèle et la Classification vectorielle de la machine. (SVM). Une base d'images étiquetées est nécessaire pour la classification supervisée afin de former un modèle à prédire de nouvelles étiquettes. Utilisant des ensembles flous pour représenter des classes d'image, la classification floue est une méthode de classification supervisée. La classification multi-modèle est une méthode de classification supervisée qui classe les images en utilisant une variété de modèles. La classification SVM est une méthode de classification supervisée qui divise les classes

d'image à l'aide d'un hyperplan. Sont également employés des réseaux neuronaux convolutifs (CNN) pour la classification supervisée des images.

Dans ce qui suit, deux algorithmes de classification de pixels seront détaillés :

1. L'algorithme des supportvector machine (SVM) ;
2. L'algorithme des réseaux de neuronesconvolutifs (CNN).

II.5.3.3.1 Les supportvector machine (SVM) :

Vapnik et Cortes (1995) ont introduit la technique d'apprentissage supervisé connue sous le nom de machine vectorielle de support (SVM). Dans la classification statistique et l'analyse de régression, il a un large éventail d'applications. Il peut être largement utilisé pour résoudre des problèmes d'apprentissage automatique tels que l'ajustement des fonctions et a un certain nombre d'avantages distinctifs dans la gestion des petits échantillons, la reconnaissance des modèles non linéaire et haute dimension. L'idée fondamentale derrière cette approche est de localiser un hyperplan fractal dans l'espace de l'échantillon pour l'ensemble de formation V qui optimisera la séparation des catégories. SVM peut également localiser la solution optimale unique au niveau mondial pour le problème de programmation quadratique, empêchant l'apparition du minimum local. L'idée directrice et l'approche du problème sont les suivantes :

Étant donné un ensemble de données échantillon $V = \{(x_i, y_i) | x_i \in \mathbb{R}^m, y_i \in \{-1, +1\}, i = 1, \dots, n\}$, alors la fonction [33] discriminante de la machine à vecteurs de support est :

$$f(x) = \text{sign}\left(\sum_{i=1}^n A_i y_i K(x, x_i) + B\right)$$

où $K(x, x_i)$ est la fonction noyau, i est le nombre de vecteur support.

Lors de la formation d'un vecteur de support pour combattre avec succès le problème de catastrophe de dimensionnalité, la fonction du noyau est cruciale. La précision de

prédiction du modèle de classification peut être augmentée en utilisant la fonction de noyau appropriée. La fonction de noyau linéaire, la fonction de noyau polynomial, la fonction de base radiale, les fonctions de noyau sigmoïde et d'autres sont fréquemment utilisées.

a. Les Avantages de support vector machine :

- Dans les espaces élevés, il est plus efficace.
- Les machines vectorielles de support fonctionnent également bien lorsqu'il y a une marge de dissociation de classe perceptible.
- Lorsque le nombre de dimensions dépasse le nombre d'exemples, cela fonctionne bien.
- Généralisation : les SVM sont efficaces dans la classification des données fraîches et non testées en raison de leurs puissantes capacités de généralisation.

b. Les inconvénients de support vector machine :

- Pour les grands ensembles de données, l'approche de la machine vectorielle de support est inacceptable.
- Il ne fonctionne pas très bien, lorsque les classes cibles se chevauchent et que l'ensemble de données inclut plus de son.
- Ne convient pas aux grands ensembles de données avec beaucoup de fonctionnalités : lorsque l'ensemble de données dispose d'un grand nombre de caractéristiques, les SVM peuvent s'exécuter lentement et utiliser beaucoup de mémoire.
- Les SVM ne conviennent pas aux ensembles de données présentant des valeurs manquantes, car ils nécessitent des ensembles de données entiers sans lacunes dans les données.

II.5.3.3.2 Les réseaux de neurones convolutifs (CNN):

Un sous-ensemble de réseaux neuronaux est un réseau de neurones convolutif. Les CNN, en revanche, sont créés spécifiquement pour analyser les images entrantes. Leur architecture est plus distincte ; il n'y a que deux de blocs principaux [34].

- 1- Le premier bloc : Puisqu'il remplit la fonction d'un extracteur de caractéristiques, il crée la particularité de ce type particulier de réseau neuronal. L'application des opérations de filtrage de convolution permettra d'y parvenir. Un certain nombre de noyaux de convolution dans la première couche filtrent l'image, produisant des « cartes de caractéristiques » qui sont ensuite normalisées (avec une fonction d'activation) et/ou redimensionnées. Nous filtrons les cartes fonctionnelles acquises avec des noyaux frais, ce qui nous donne de nouvelles fonctionnalités pour normaliser et redimensionner, et ainsi de suite. Ce processus peut être effectué plusieurs fois. Les valeurs des cartes de caractéristiques les plus récentes sont finalement concaténées en un vecteur. Ce vecteur désigne l'entrée du deuxième bloc ainsi que la sortie du premier bloc.
- 2- Le second bloc : Pour produire un nouveau vecteur de sortie, les valeurs de vecteur d'entrée sont transformées (en utilisant un certain nombre de combinaisons linéaires et de fonctions d'activation). L'élément i de ce vecteur final, qui reflète la probabilité que l'image appartient à la classe i , a autant d'éléments qu'il y a de classes.

Similaire aux réseaux neuronaux convolutifs, la rétropropagation de gradient est utilisée pour calculer les paramètres de la couche tout en minimisant l'entropie croisée pendant l'entraînement. Ces facteurs, cependant, identifient avec précision les caractéristiques des photographies dans les instances de CNN tout en minimisant l'entropie croisée pendant la formation. Ces facteurs, cependant, identifient précisément les caractéristiques des photographies dans l'exemple de CNN. Maintenant, considérez les différents types de couches CNN.

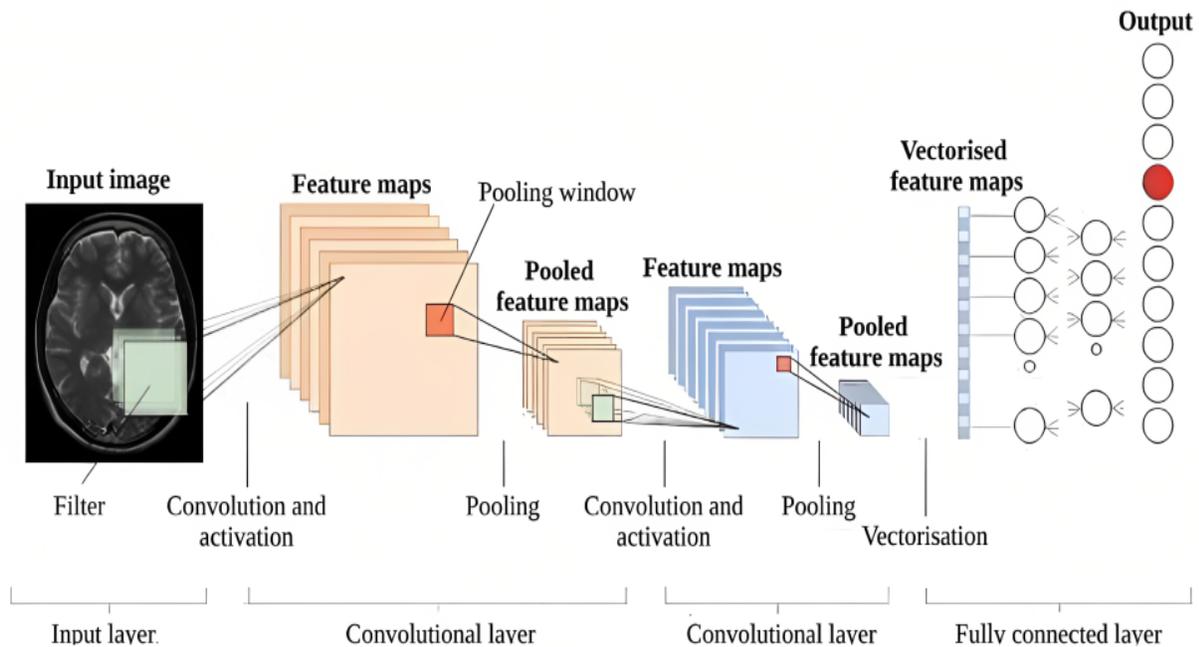


Figure II. 5 : Structure d'un réseau de neurones convolutifs [35]

Il existe quatre types de couches pour un réseau de neurones convolutif :

- La couche de convolution.
- La couche de pooling.
- La couche de correction ReLU.
- La couche fully-connected.

1- La couche de convolution :

La convolution est le processus d'application d'un filtre mathématique sur une image i.e représenter par une matrice l'image et utiliser les multiplications de chaque pixel par la valeur de la matrice. En utilisant cette méthode nous pouvons localiser des zones de l'image qui pourraient nous intéresser. Une fois la première ligne terminée, le filtre doit passer d'une boîte à la prochaine itération. Le même processus est effectué pour chaque ligne et chaque colonne une fois que le filtre "descend" une boîte après que la première ligne est terminée. (Voir Figure II.5).

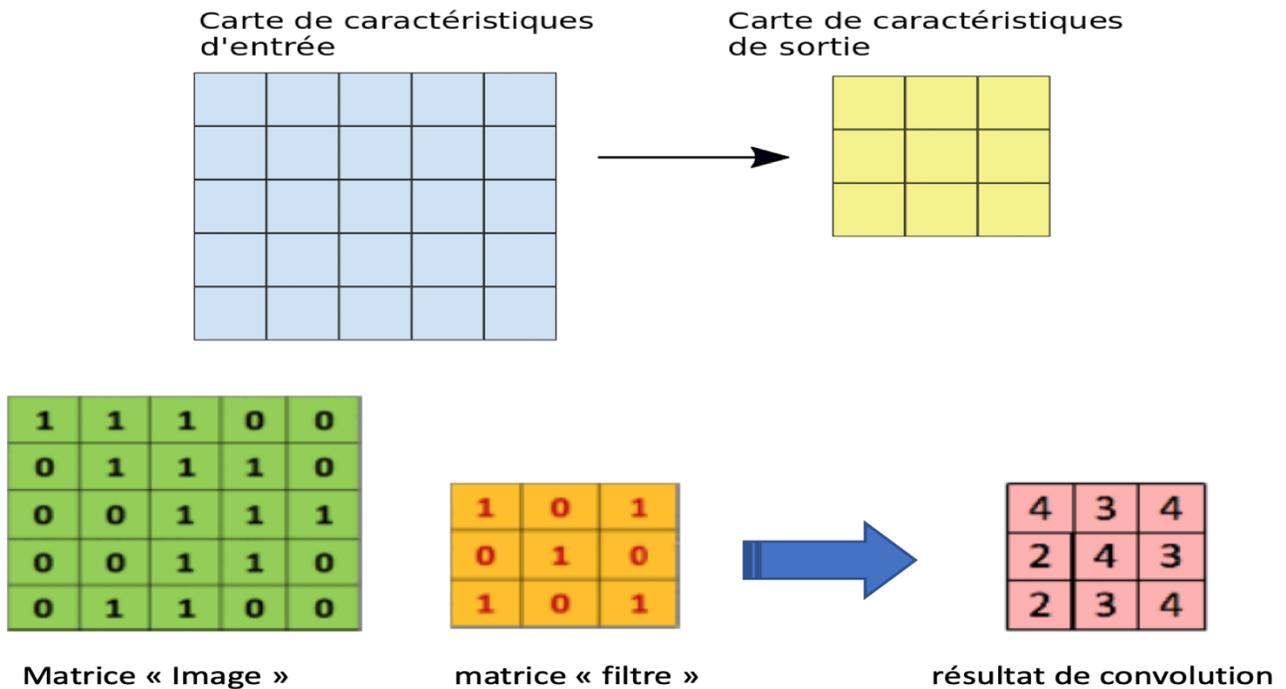
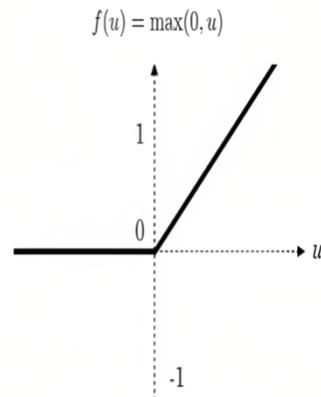


Figure II. 6 : Opération de convolution [36]

2- La couche de correction ReLU :

Après la convolution, une fonction appelée ReLU doit être appliquée à chaque pixel de l'image, remplaçant chaque valeur négative par un 0. Si cette fonction n'est pas utilisée, la fonction produite sera linéaire et le problème XOR existera toujours parce qu'il n'y a pas de fonction d'activation utilisée dans la couche de convolution.

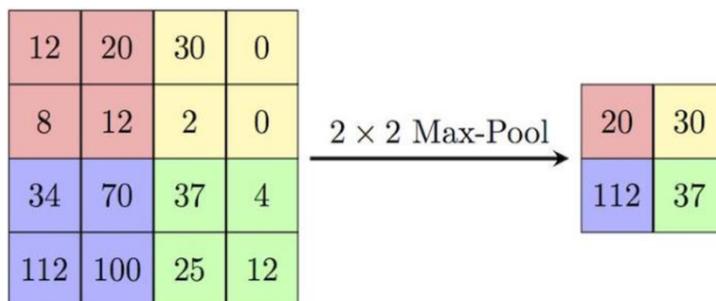
Les réseaux neuronaux convolutifs utilisent souvent ReLU parce que c'est une fonction simple à calculer : $f(u) = \max(0, u)$. En conséquence, il fonctionne mieux que d'autres tâches qui nécessitent des activités coûteuses



3- La couche pooling

Ce type de couche, qui applique l'opération de regroupement à chacune de plusieurs cartes de caractéristiques qu'elle obtient comme input, est souvent utilisé entre deux couches de convolution. La technique de regroupement vise à minimiser la taille des photostout en conservant les détails clés. Pour cela, l'image est divisée en cellules régulières, et la valeur maximale de chaque cellule est conservée.

La technique consiste à visualiser une fenêtre de 2 ou 3 pixels glissant sur une image, similaire à la convolution. Cependant, cette fois-ci, nous prenons deux étapes pour une fenêtre de deux tailles et trois pour un pixel. «kernel size » se réfère à la taille de la fenêtre, et « strides » au nombre d'étapes. La valeur la plus élevée parmi celles de la fenêtre est sélectionnée à chaque étape, et cette valeur crée un nouveau pixel dans une image fraîche. Le terme pour cela est Max Pooling.



4- Couche fully-connected

La couche complètement connectée d'un réseau neuronal est toujours la dernière. Ce type de couche prend un vecteur en tant qu'entrée et crée un nouveau vecteur de sortie. Pour ce faire, il applique une combinaison linéaire et peut-être une fonction d'activation aux données d'entrée.

La couche la plus récente entièrement connectée, qui produit un vecteur de taille N où N est le nombre de classes dans notre problème de classification d'image nous permet de catégoriser l'image comme l'entrée du réseau. Chaque élément vectoriel affiche la probabilité que l'image fournie appartient à une classe donnée.

II.5.3.3.2 Architecteur de réseaux de neurones convolutifs

Plusieurs architectures de CNN ont été présentées au cours des dix dernières années [37,38]. L'architecture du modèle joue un rôle clé dans l'amélioration de la fonctionnalité des différentes applications. De 1989 à nos jours, l'architecture de CNN a subi un certain nombre de modifications. Ces modifications comprennent la régularisation, l'optimisation des paramètres et la reformulation structurelle. D'autre part, il convient de souligner que l'amélioration significative des performances de CNN a été principalement le résultat de la restructuration de l'unité de traitement et de la création de blocs supplémentaires. L'utilisation de la profondeur du réseau a vu les progrès les plus inventifs dans les conceptions de CNN, en particulier.

Nous pouvons procéder à la classification des photos une fois que nous avons terminé la convolution, ReLu et les étapes de regroupement. Envoyer tous les pixels à un réseau de neurones multicouches est la dernière étape. Par rapport à l'utilisation d'un réseau de neurones artificiels sans convolution, la phase de classification sera significativement plus efficace car nous avons pu obtenir les composants importants d'une image que nous avons condensée [34].

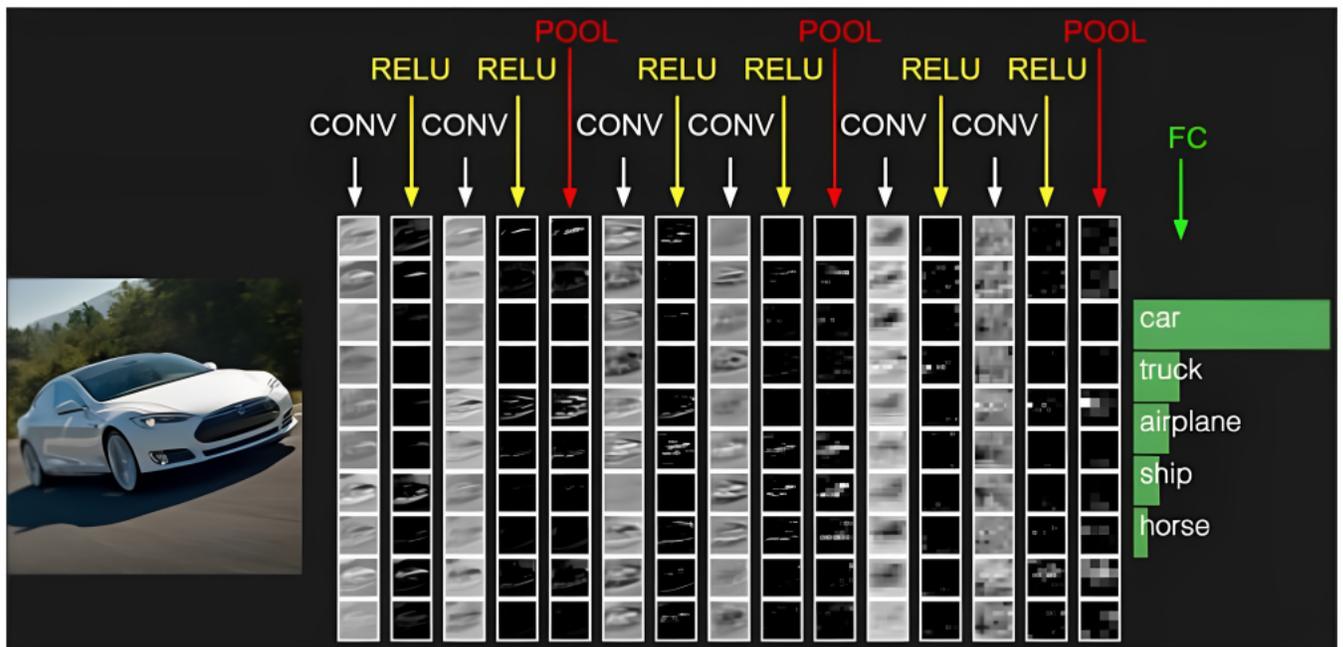


Figure II. 7 : Architecteur de CNN [34]

II.5.3.3.2.3 Avantages de réseaux de neurones convolutifs :

- 1- Les CNN utilisent la convolution pour réduire la quantité d'information nécessaire au traitement des images, les rendant plus rapides et plus efficaces que d'autres algorithmes.
- 2- Les CNN peuvent apprendre à reconnaître des schémas complexes dans les images, ce qui les rend idéales pour la reconnaissance faciale et la détection d'objets.
- 3- Les CNN sont plus résistants au bruit que les autres algorithmes en raison de plusieurs couches de filtres.
- 4- Les CNN sont des outils poly valents pour de nombre uses applications en raison de leurs caractéristiques génériques.
- 5- Les CNN automatisent l'extraction de fonctionnalités, ce qui les rend idéales pour les tâches où les fonctionnalités sont inconnues.

II.5.3.3.2.4 Inconvénients de réseaux de neurones convolutifs :

- 1- Les CNN nécessitent beaucoup de puissance de traitement et de mémoire pour s'entraîner et fonctionner, ce qui les rend impraticables pour des ressources limitées.
- 2- Les CNN nécessitent de grands ensembles de données pour atteindre des taux de précision élevés, mais de petites quantités de données peuvent surpasser.
- 3- Les CNN nécessitent de grands ensembles de données pour atteindre des taux d'exactitude élevés, mais leur manque d'interprétation peut être problématique dans les applications où il est important de comprendre pourquoi des décisions sont prises.
- 4- Les CNN sont vulnérables aux attaques adversaires, potentiellement compromettant la sécurité dans les véhicules autonomes.
- 5- Les CNN ont une capacité limitée à généraliser à de nouvelles situations, ce qui les rend difficiles à utiliser dans les applications.

II.6 conclusions :

Les différentes méthodes de classification supervisées et non-supervisées ainsi que la segmentation de l'images avec les approches basées sur le contour, la région et la classification sont décrites dans ce chapitre. Comme nous l'avons vu, le regroupement permet d'organiser des éléments (tels que des personnes ou des variables) en un certain nombre de groupes ou de classes, de segments ou d'un groupe. Afin de classer une image, des pixels comparables sont regroupés ensemble, souvent au niveau de l'échelle grise et en dessous des ensembles de classes. Les réseaux neuronaux convolutifs (CNN) et le support de la machine vectorielle (SVM) sont deux approches de catégorisation automatique que nous sommes intéressés à utiliser dans notre étude. Dans le prochain chapitre, nous allons appliquer toutes ces techniques aux scans MRI du cerveau humain.

Chapitre III:

Étude expérimentale

III.1 Introduction :

Les différentes approches discutées dans les chapitres précédents sont mises en pratique dans ce chapitre. Nous discutons les bibliothèques installées et l'environnement cloud configuré avant l'exploration de la base de données et des réseaux de neurones. Nous passons ensuite à l'application des différents réseaux développés comme démontré au chapitre 2. Ici, nous nous sommes basés sur les modèles VGG16, VGG19 et ResNet50 pour la détection tumorale. Nous décrivons d'abord les processus de mise en œuvre avant de comparer et d'analyser les résultats.

III.2 L'environnement de travail :

Nous présentons dans ce qui suit l'environnement matériel et logiciel nécessaire à la réalisation de notre travail.

III.2.1 Hardware

- Le matériel utilisé pour l'exécution de ce travail est un Laptop MacBookpro Des caractéristiques suivantes :
- CPU : Intel Core i5 – 2015 2.9GHz
- RAM : 8GB.
- Disque : 512 GB.SSD
- GPU : Intel Iris graphics 6100 -1536 MB.
- MacOS Monterey.

III.2.2 Software :

III.2.2.1 Langage de programmation.

Python est le langage de programmation le plus largement utilisé dans le domaine de l'intelligence artificielle, et en particulier dans l'apprentissage profond, car il contient les bibliothèques les plus fiables et utiles pour l'utilisation des réseaux de neurones et de la vision artificielles, Nous avons utilisé la version 3.11.3 /64 bit de python.



Figure III. 1 : Le logo de langage de programmation Python

III.2.2.2 visualstudiocode :

Visual Studio de Microsoft prend en charge une gamme de langages de programmation, notamment C#, C++ et Python. C'est un IDE, ou environnement de développement intégré. Visual Studio simplifie considérablement le processus de développement logiciel en fournissant une interface facile à utiliser et une variété de fonctionnalités pour augmenter la productivité et la collaboration des développeurs.



Figure III. 2 : Le logo de programme de visualstudiocode [39]

III.3 Base de Données :

La détection des tumeurs cérébrales se base sur des images IRM. Parce que les hôpitaux et les installations d'imagerie n'étaient pas disponibles, une solution a été développée en parcourant des sites Web et des plates-formes pour la recherche médicale spécialisée. "Kaggle" est un dataset open source offrant 250 photos IRM au format JPG présentant les 2 classes en 2 dossiers :

- Classe 1 : présence de la tumeur nommée « Yes »

- Classe 2 : absence de la tumeur nommée « No »

Nous avons utilisé un moyen très pratique pour résoudre le problème de la pénurie de photos IRM car il était évident qu'il n'y avait pas assez d'images pour former nos réseaux neuronaux. Plus de données/images ont été produites à partir des données d'origine en utilisant une approche de traitement d'image. Dans les paragraphes qui suivent, l'approche « Augmentation des données » sera introduite et son fonctionnement sera décrit.

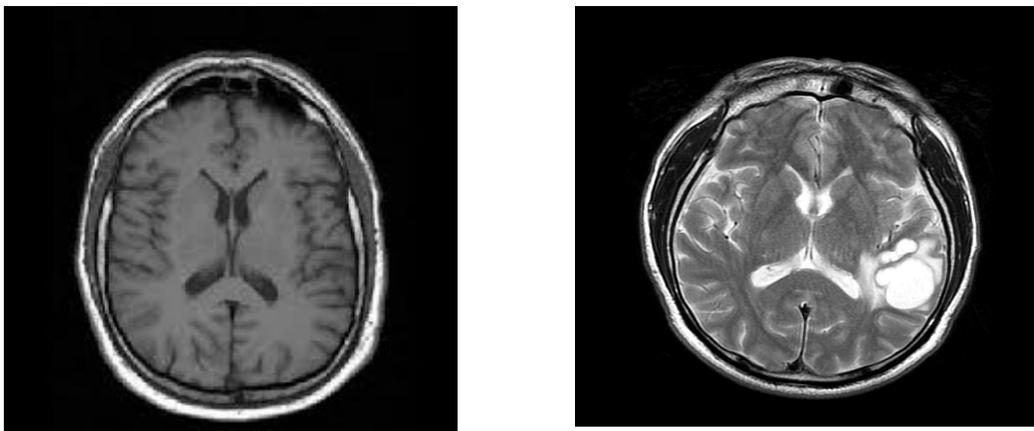


Figure III. 3 : deux échantillons de notre base représentantes chaque classe de la maladie

III.4 Augmentation d'images (Data Augmentation) :

L'augmentation des données est une méthode permettant d'agrandir la quantité de données d'entraînement par la création de nouvelles données à partir des données d'entraînement anciennes. Ceci est réalisé en transformant des photos de l'ensemble de données d'apprentissage qui sont des membres de la même classe de l'image d'origine. Grâce à l'utilisation de techniques d'augmentation, la capacité des modèles à transférer leurs connaissances à de nouvelles données peut être améliorée.



Figure III. 4 : étapes d’obtention d’une nouvelle base de données à partir de la base de données originale

La classe ImageDataGenerator de la bibliothèque « Keras » permet l'utilisation automatisée de l'augmentation de données lors de la formation d'un modèle. La configuration des types d'augmentation de données est d'abord fournie par des paramètres au constructeur de classe, après quoi la classe peut être créée.[40]

III.5 Le Transfer learning vs. le Deep Learning:

Le transfert learning est la pratique d'employer un réseau qui a déjà été entraîné pour identifier des classes d'images ou d'objets sur lesquels il n'a pas été formé. Dans le contexte de l'apprentissage profond, l'extraction de fonctionnalités et le fine-tuning sont deux formes différentes de transfert. Lors de l'extraction de fonctionnalités, le réseau pré-entraîné est utilisé comme n'importe quel autre extracteur de caractéristiques, à la fin l'architecture du modèle est mise à jour.[41]

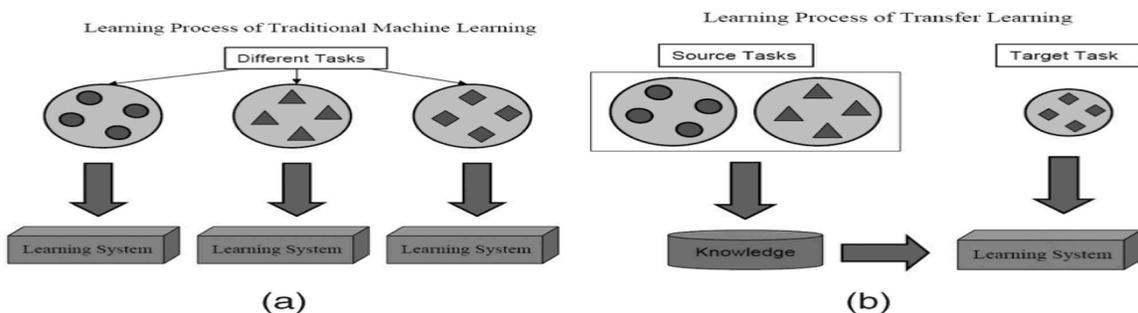


Figure III. 5 : la différence entre le transfert learning et le machine learning

III.6 Structures modifiées des modèles utilisés :

III.6.1 Le modèle VGG 16 :

Puisqu'il n'y a que 3 x 3 couches convolutives utilisées dans ce réseau, il se distingue par sa simplicité. L'articulation maximale contrôle la diminution de la taille du volume. Puis vient un classificateur softmax, qui est suivi de deux couches entièrement connectées avec un total de 4096 nœuds chacune. Dropout fonctionne en séparant irrationnellement les nœuds de la couche actuelle et de la prochaine. Les déconnexions aléatoires qui se produisent pendant l'apprentissage des lots aident dans l'introduction involontaire du modèle de la redondance.[42]

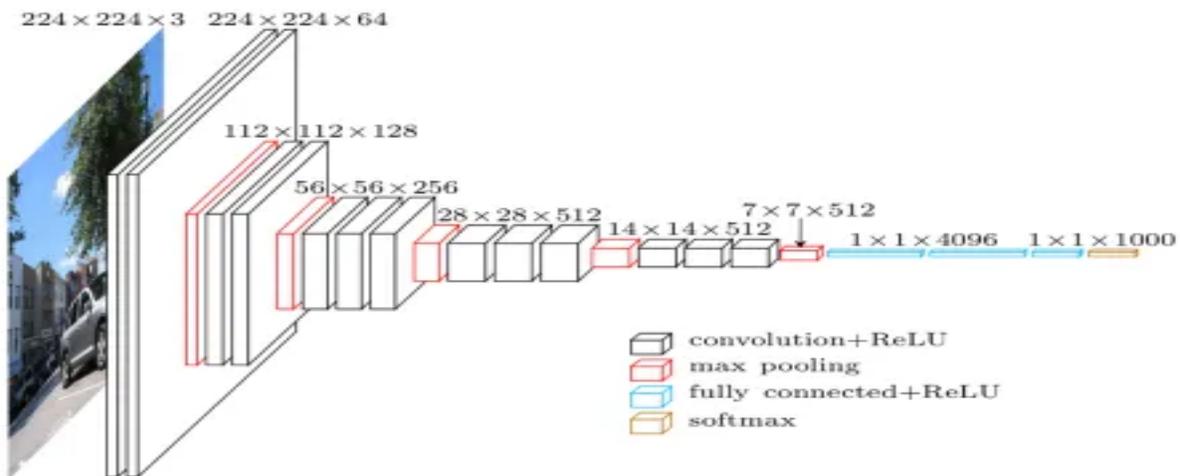


Figure III. 6 : Visualisation de l'architecture VGG [43]

• VGG16 :

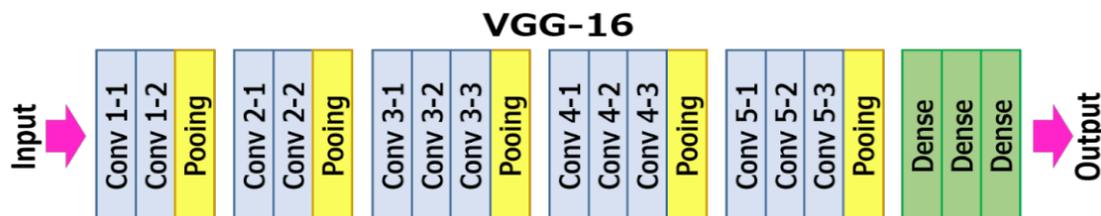


Figure III. 7 : Le réseau de neurones VGG16 [42]

• VGG19 :

VGG-19 se compose de 3 couches complètement liées à la fin de 16 couches de convolution et de non-linéarité ReLU, qui sont séparées par 5 couches jointes.

Le figure 08 répertorie les configurations ConvNet qui ont été examinées pour ce travail, une par colonne. Les filets seront désignés par leurs noms (A-E) à l'avenir. La seule différence entre les configurations est la profondeur, qui va de 11 couches de poids dans le réseau A (8 conv. et 3 couches FC) à 19 niveaux de poids dans le réseau E (16 conv. et 3 couches FC). Toutes les configurations respectent la conception de base. Les couches convolutives ont une largeur (nombre de canaux) relativement petite, commençant à 64 dans la première couche et augmentant d'un facteur 2 après chaque couche avec une mise en commun maximale jusqu'à atteindre 512.[42]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure III. 8 : Les réseaux convolutionnels très profonds pour la reconnaissance d'images à grande échelle [42]

III.6.2 Le modèle ResNet50 :

Ce modèle a pour objectif l'amélioration du problème de gradient, il est composé de 48 Convolution layers avec une couche MaxPooling and une couche Average Pooling. Il a 3.8×10^9 opérations de Floating points.[44]

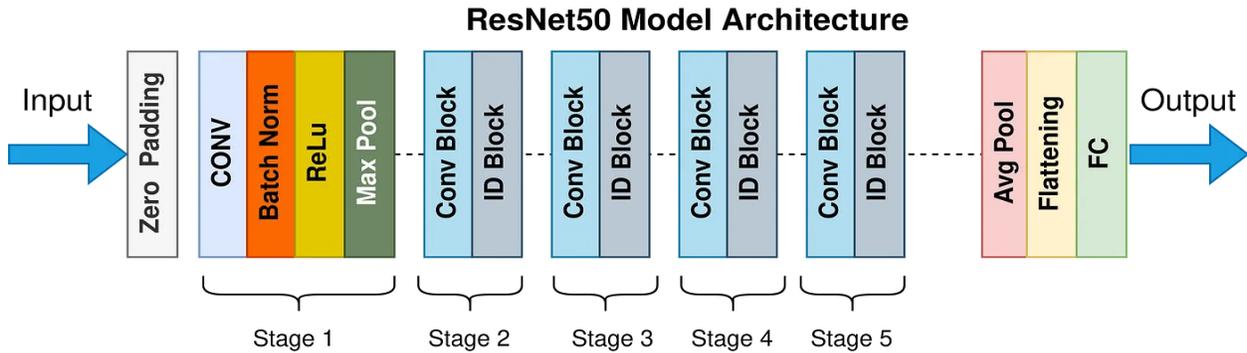


Figure III. 9 : Architecture du modèle ResNet 50 [45]

III.7 Implémentation générale des réseaux de classification :

Afin d'implémenter nos réseaux de neurones pour pouvoir classer les images IRM on va suivre les étapes montrées dans la figure10 :

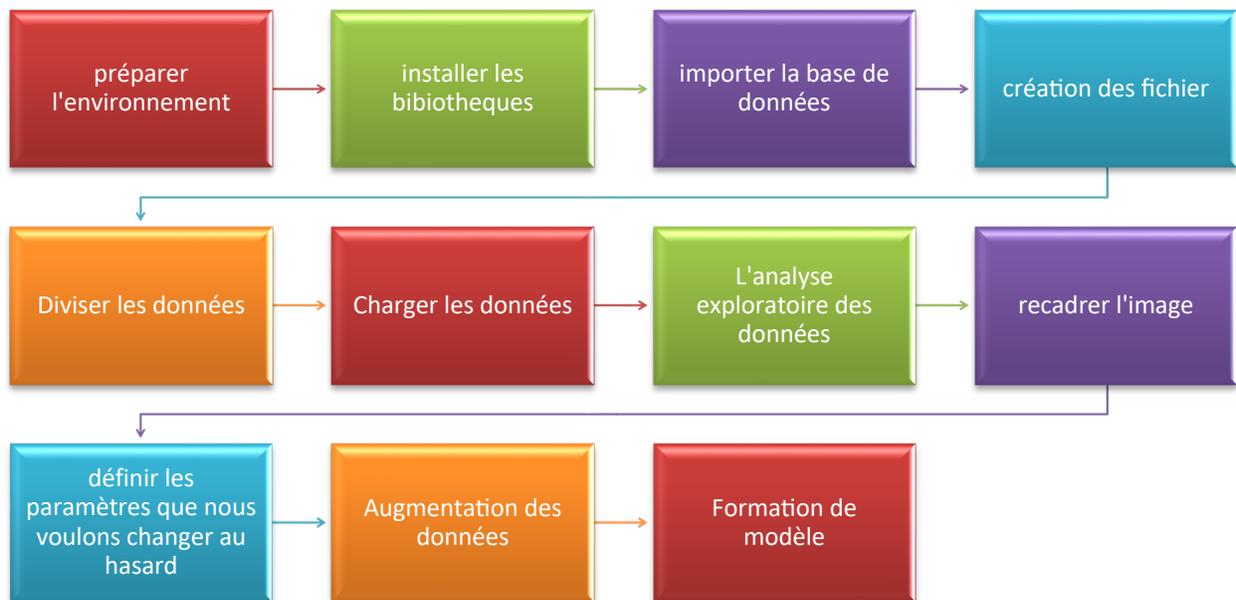


Figure III. 10 : Les étapes d'implémentation des réseaux de classification

III.7 .1 Préparer l'environnement :

D'abord on prépare notre environnement. On va utiliser les modèles VGG 16, VGG 19 et Resnet50 pour la détection.

```
from keras.applications.vgg16 import VGG16, preprocess_input
from keras.applications.vgg19 import VGG19, preprocess_input
from keras.applications import ResNet50
```

Figure III. 11 :code de préparation des réseaux et de l'environnement

III.7 .2 Les bibliothèques utilisées :

Les bibliothèques utilisées dans le programme.

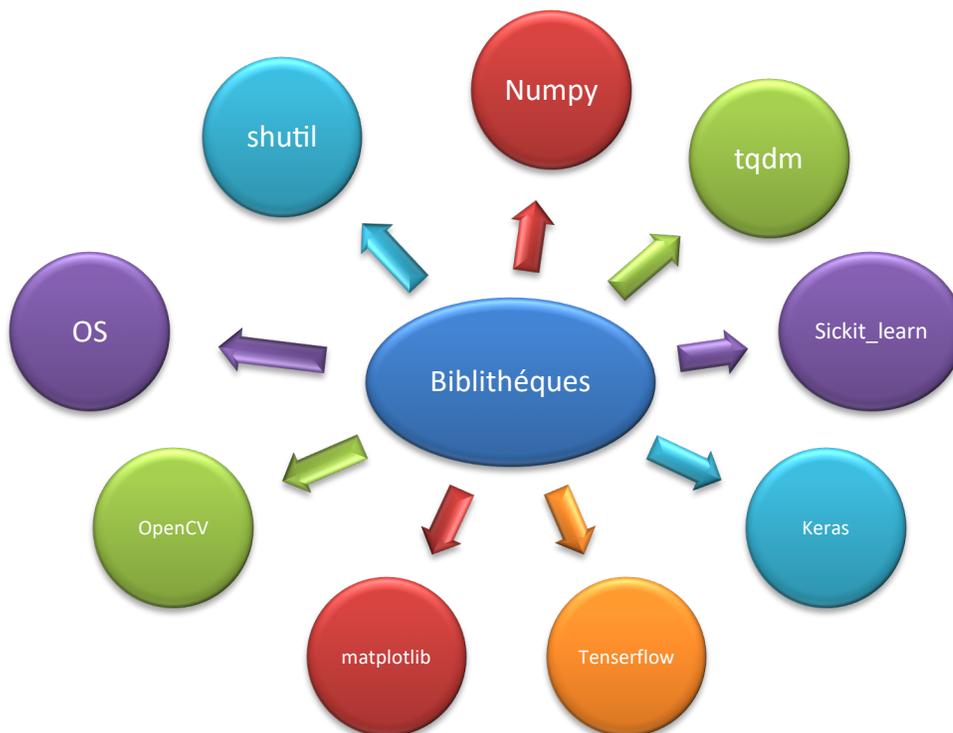


Figure III. 12 : Les bibliothèques utilisées

Dans cette partie on importe les bibliothèques qu'on a installé et aussi celles qui sont préinstallées :

```
import os
import shutil
import cv2
from keras import layers
import numpy as np
import itertools
import imutils
import matplotlib.pyplot as plt
import tensorflow as tf
from tqdm import tqdm
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split

from keras.applications.vgg16 import VGG16, preprocess_input
from keras.applications.vgg19 import VGG19, preprocess_input
from keras.applications import ResNet50

from keras.preprocessing.image import ImageDataGenerator
from keras.models import Model, Sequential
from keras.optimizers import Adam, RMSprop
from keras.callbacks import EarlyStopping
from keras.preprocessing import image
from keras.utils import plot_model
from keras.layers import Flatten, Dense, Dropout
```

Figure III. 13 :code d’importation des bibliothèques et les paramètres d’entrainement

III.7 .3 Créer les dossiers de la base de données :

Nous sommes créés c'est des dossiers dans cette méthode pour une utilisation dans le programme.

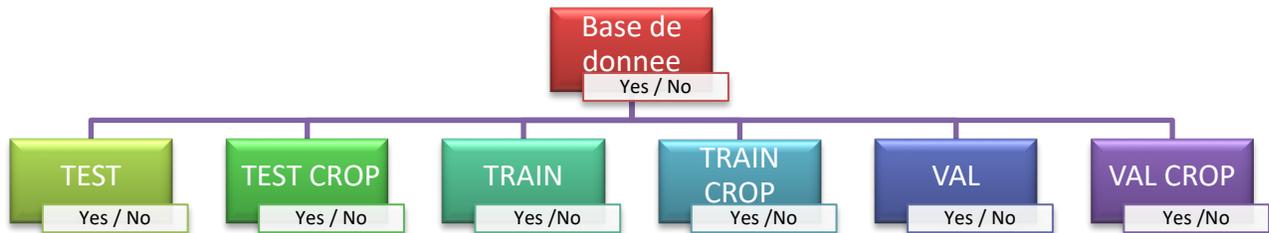


Figure III. 14 : Division de la base de données

III.7.4 Diviser la base de données et faire le pre-processing :

Ensuite, les photographies de la base de données sont copiées et réparties dans des dossiers en fonction de nos besoins d'apprentissage, 75% des images allant dans le dossier TRAIN et 25% allant dans les fichiers TEST et VAL.

```
# Split the data by train/val/test
for CLASS in os.listdir(IMG_PATH):
    if not CLASS.startswith('.'):
        IMG_NUM = len(os.listdir(os.path.join(IMG_PATH, CLASS)))
        for (n, FILE_NAME) in enumerate(os.listdir(os.path.join(IMG_PATH, CLASS))):
            img = os.path.join(IMG_PATH, CLASS, FILE_NAME)
            if n < 5:
                shutil.copy(img, os.path.join(TEST_DIR, CLASS.upper(), FILE_NAME))
            elif n < int(0.8 * IMG_NUM):
                shutil.copy(img, os.path.join(TRAIN_DIR, CLASS.upper(), FILE_NAME))
            else:
                shutil.copy(img, os.path.join(VAL_DIR, CLASS.upper(), FILE_NAME))
```

Figure III. 15 :code de division de la base de données

Les données (images) passent encore par un autre programme qui lit leurs noms, chemins et leur classe (YES ou NO) :

```
# Load data
def load_data(dir_path, img_size=(100, 100)):
    X = []
    y = []
    labels = {}
    i = 0
    for path in os.listdir(dir_path):
        if not path.startswith('.'):
            labels[path] = path
            for file in os.listdir(os.path.join(dir_path, path)):
                if not file.startswith('.'):
                    img = cv2.imread(os.path.join(dir_path, path, file))
                    img = cv2.resize(img, img_size) # Resize the image to desired dimensions
                    X.append(img)
                    y.append(i)
            i += 1
    X = np.array(X)
    y = np.array(y)
    print(f'{len(X)} images loaded from {dir_path} directory.')
    return X, y, labels
```

Figure III. 16 : Lecture des chemins de la base de données

Définition de la taille des images ([Hauteur * Largeur]) [224 * 224] et chargement de l'ensemble de données.

```
67
68 IMG_SIZE = (224,224)
69 X_train, y_train, labels = load_data(TRAIN_DIR, IMG_SIZE)
70 X_test, y_test, _ = load_data(TEST_DIR, IMG_SIZE)
71 X_val, y_val, _ = load_data(VAL_DIR, IMG_SIZE)
```

```
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
100%|
188 images loaded from /Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/TRAIN/ directory.
100%|
10 images loaded from /Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/TEST/ directory.
100%|
51 images loaded from /Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/VAL/ directory.
```

Figure III. 17 : Chargement de la base de données

III.7.5 Visualisation de la base de données :

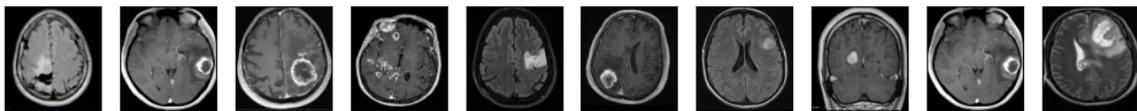
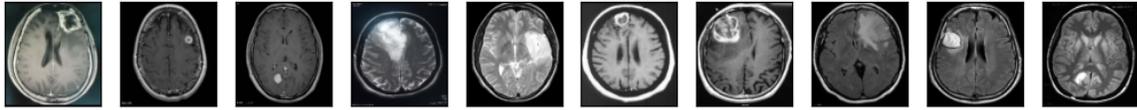
Maintenant, on va visualiser notre base de données chargée :

```
102 # show the imgaes original
103 def plot_samples(X, y, labels_dict, n=50):
104     """
105     Creates a gridplot for desired number of images (n) from the specified set
106     """
107     for index in range(len(labels_dict)):
108         imgs = X[np.argwhere(y == index)][:n]
109         j = 10
110         i = int(n/j)
111
112         plt.figure(figsize=(15,6))
113         c = 1
114         for img in imgs:
115             plt.subplot(i,j,c)
116             plt.imshow(img[0])
117
118             plt.xticks([])
119             plt.yticks([])
120             c += 1
121         plt.suptitle('Tumor: {}'.format(labels_dict[index]))
122         plt.show()
```

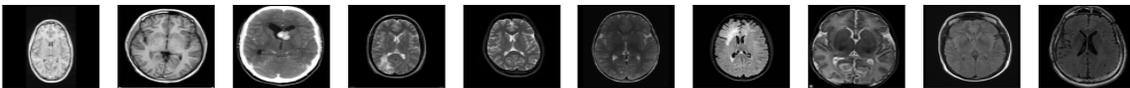
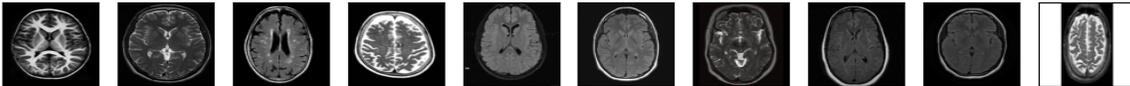
Figure III. 18 : code de préparation de la visualisation de la base de données

On va afficher quelques échantillons de notre base de données avec leurs classes respectives :

Tumor: YES



Tumor: NO



```
124 plot_samples(X_train, y_train, labels, 20)
```

Figure III. 19 : Visualisation de quelques échantillons de notre base de données avec leurs classes

III.7.6 Le prétraitement de la base de données et les réseaux de neurones :

Pour ce qui suit on va appliquer le Preprocessing à la base de données qu'on va l'introduire à nos réseaux de neurones :

```

137 #crop image
138 def crop_imgs(set_name, add_pixels_value=0):
139     """
140     Finds the extreme points on the image and crops the rectangular out of them
141     """
142     set_new = []
143     for img in set_name:
144         gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
145         gray = cv2.GaussianBlur(gray, (9, 9), 4)
146
147         # threshold the image, then perform a series of erosions +
148         # dilations to remove any small regions of noise
149         thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]
150         thresh = cv2.erode(thresh, None, iterations=2)
151         thresh = cv2.dilate(thresh, None, iterations=2)
152
153         # find contours in thresholded image, then grab the largest one
154         cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
155         cnts = imutils.grab_contours(cnts)
156         c = max(cnts, key=cv2.contourArea)
157
158         # find the extreme points
159         extLeft = tuple(c[c[:, :, 0].argmin()][0])
160         extRight = tuple(c[c[:, :, 0].argmax()][0])
161         extTop = tuple(c[c[:, :, 1].argmin()][0])
162         extBot = tuple(c[c[:, :, 1].argmax()][0])
163
164         ADD_PIXELS = add_pixels_value
165         new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS, extLeft[0]-ADD_PIXELS:extRight[0]+ADD_PIXELS].copy()
166         set_new.append(new_img)
167
168     return np.array(set_new)

```

Figure III. 20 : code de Preprocessing de la base de données

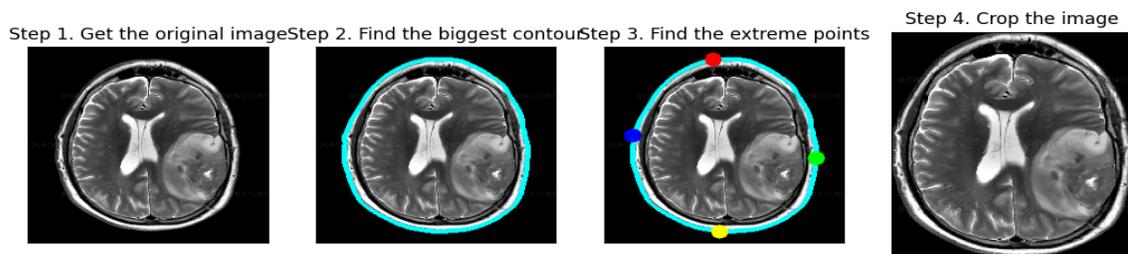


Figure III. 21 : Prétraitement de la base de données

Après que les réseaux appliquent le Prétraitement nécessaire à la détection on va maintenant déplacer notre base de données à d'autres dossiers qu'on va les définir avec leurs chemins :

```

248 # saving new images to the folder
249 save_new_images(X_train_crop, y_train, folder_name='/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/TRAIN_CROP/')
250 save_new_images(X_val_crop, y_val, folder_name='/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/VAL_CROP/')
251 save_new_images(X_test_crop, y_test, folder_name='/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/TEST_CROP/')
252

```

Figure III. 22 : Définir les nouveaux dossiers et chemins virtuels de la base de données

Maintenant, On va afficher nos nouvelles images détectées après avoir effectuer le Prétraitement et le traitement nécessaire :

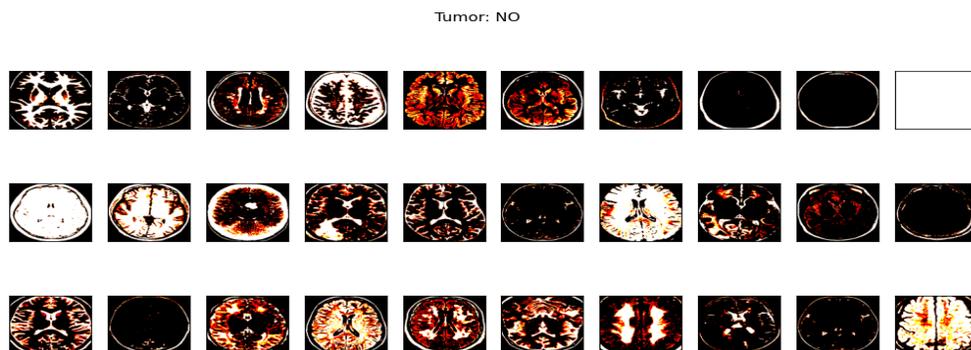
```

102 # show the imgaes original
103 def plot_samples(X, y, labels_dict, n=50):
104     """
105     Creates a gridplot for desired number of images (n) from the specified set
106     """
107     for index in range(len(labels_dict)):
108         imgs = X[np.argwhere(y == index)][:n]
109         j = 10
110         i = int(n/j)
111
112         plt.figure(figsize=(15,6))
113         c = 1
114         for img in imgs:
115             plt.subplot(i,j,c)
116             plt.imshow(img[0])
117
118             plt.xticks([])
119             plt.yticks([])
120             c += 1
121         plt.suptitle('Tumor: {}'.format(labels_dict[index]))
122         plt.show()

```

Figure III. 23 :code de préparation à visualiser nos nouvelles images

Le fichage des nouvelles images :



Tumor: YES

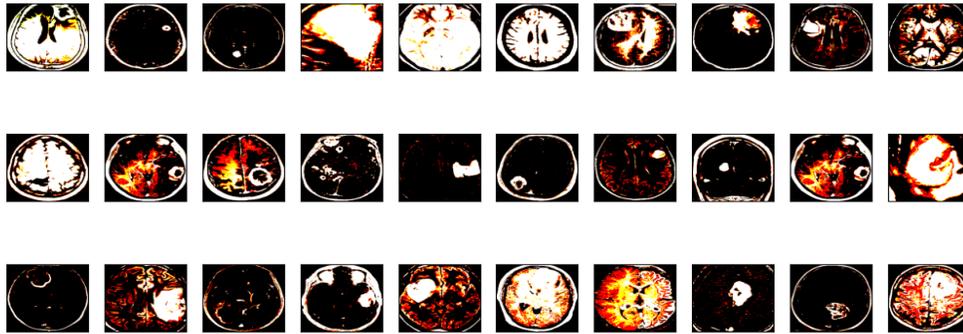


Figure III. 24 : Visualisation de nos nouvelles images classifiées

III.7.7 L'augmentation des données :

Comme illustré à la figure 25, nous utiliserons l'approche d'augmentation des données pour ajouter de nouvelles données à partir de photographies existantes à notre base de données après avoir soumis chaque image à une série de modifications aléatoires (y compris rotation aléatoire, mise à l'échelle, changement de luminosité, etc.) :

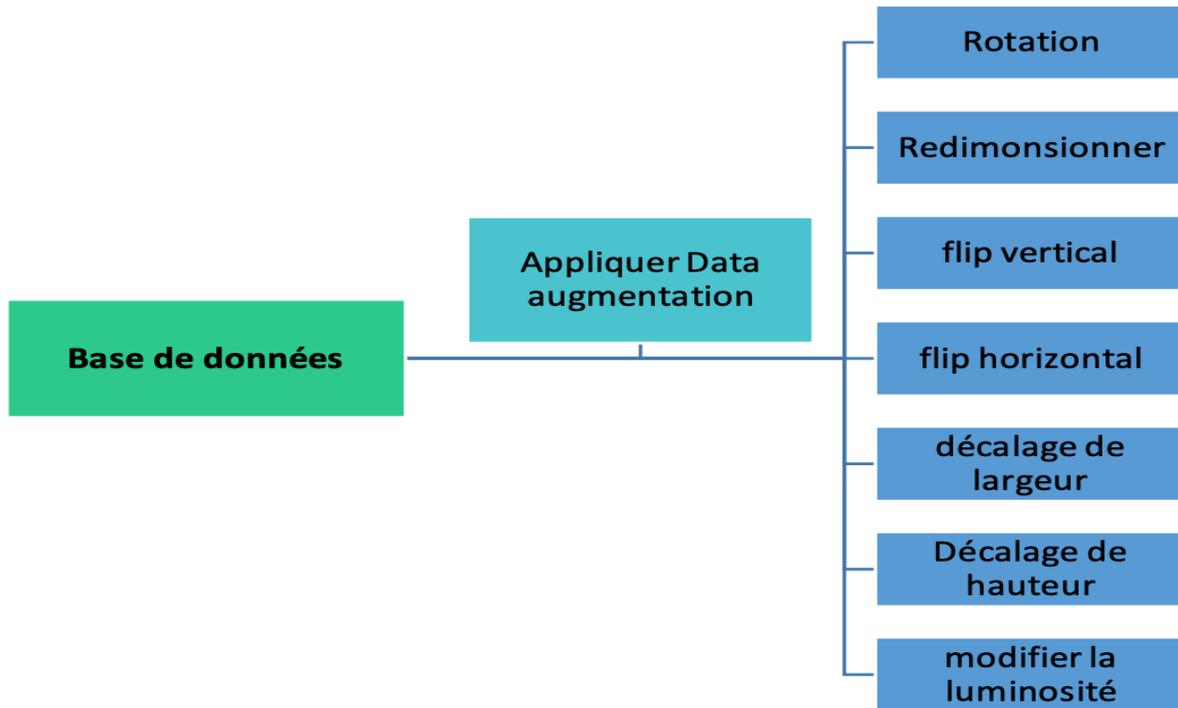


Figure III. 25 : Schémasynoptique du principe de la technique d'augmentation des données

Le code dans la figure 24 montre comment appliquer l'augmentation de données en apportant des transformations :

```
273 # set the paramters we want to change randomly
274 demo_datagen = ImageDataGenerator(
275     rotation_range=20,
276     width_shift_range=0.2,
277     height_shift_range=0.2,
278     rescale=1./255,
279     shear_range=0.2,
280     brightness_range=[0.1, 1.5],
281     horizontal_flip=True,
282     vertical_flip=True
283 )
284
285 preview='/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/preview'
286
287 x = X_train_crop[0]
288 x = x.reshape((1,) + x.shape)
289
290 i = 0
291 for batch in demo_datagen.flow(x, batch_size=1, save_to_dir='preview', save_prefix='aug_img', save_format='jpg'):
292     i += 1
293     if i > 20:
294         break
295
296 plt.imshow(X_train_crop[0])
297 plt.xticks([])
298 plt.yticks([])
299 plt.title('Original Image')
300 plt.show()
301
```

Figure III. 26 : code d'implémentation de l'augmentation de données

Un exemple de resultat d'une image ayant subit l'augmentation des données :

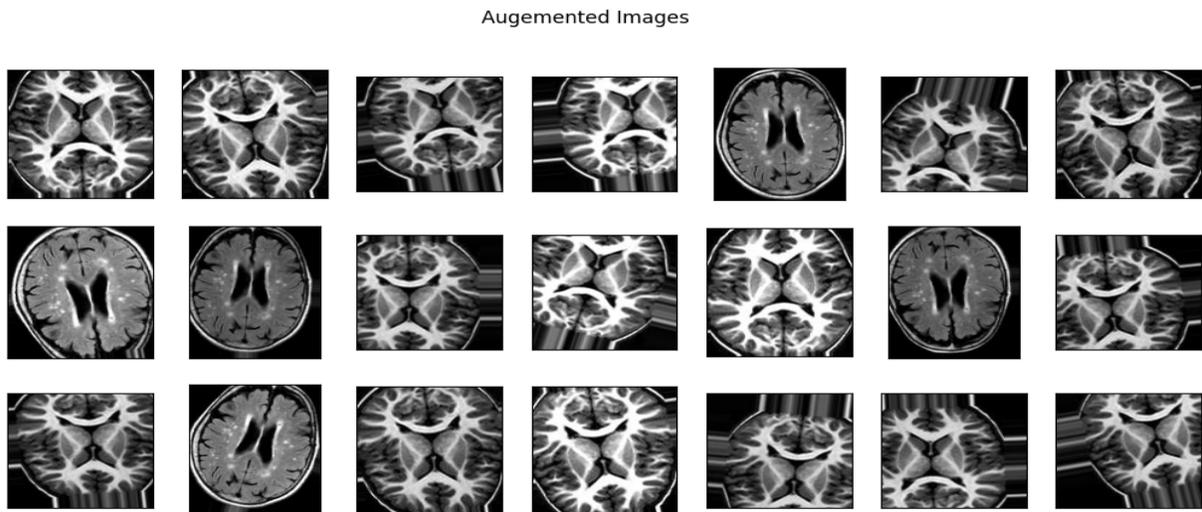


Figure III. 27 : Exemple d'une image ayant subit l'augmentation de données

III.7.8 Appliquer l'augmentation de données pour toute la base de données

Augmenter les données de formation est ce que fait «ImageDataGenerator». Afin d'améliorer les performances de nos réseaux et de leur fournir des données d'entraînement supplémentaires, ils seront automatiquement formés tout au long de la phase d'entraînement.

```
318 # Stage 3: Data Augmentation
319 # Create data generators with data augmentation
Run and Debug (⇧⌘D) val_datagen = ImageDataGenerator(
321     preprocessing_function=preprocess_input,
322     rotation_range=20,
323     width_shift_range=0.2,
324     height_shift_range=0.2,
325     shear_range=0.2,
326     zoom_range=0.2,
327     horizontal_flip=True
328 )
329
330 val_datagen = ImageDataGenerator(
331     preprocessing_function=preprocess_input
332 )
```

Figure III. 28 :code d'implémentation de l'augmentation de données pour toute la base

ImageDataGeneratorgénére les données en utilisant :

* train_datagen.flow_from_directory () * qui prend "le chemin du dossier de données (Data folderpath), taille ciblée (target_size), mode de couleurs (color_mode), taille du lot (batch_size), mode de classe (class_mode) " commeparamètres.

```
318 # Stage 3: Data Augmentation
319 # Create data generators with data augmentation
320 train_datagen = ImageDataGenerator(
321     preprocessing_function=preprocess_input,
322     rotation_range=20,
323     width_shift_range=0.2,
324     height_shift_range=0.2,
325     shear_range=0.2,
326     zoom_range=0.2,
327     horizontal_flip=True
328 )
329
330 val_datagen = ImageDataGenerator(
331     preprocessing_function=preprocess_input
332 )
```

Figure III. 29 :code de génération de la base de données augmentée.

III.7 .9 Appliquer le Transfer Learning :

Les poids des modèles sont chargés (à l'exception des couches finales de chaque modèle, qui sont laissées de côté puisque nous voulons que nos propres couches soient les couches finales ; cela est accompli en définissant `include_top` sur `False`) ; ces modèles seront ajoutés prochainement afin de former le réseau.

Il n'est pas nécessaire de former toutes les couches ; à la place, nous entraînerons simplement les poids finaux que nous ajouterons car les poids chargés des couches précédentes ont déjà été pré-entraînés et optimisés.

```
349 # Stage 4: Model Training
350 # Path to the VGG19 weights file
351 vgg16_weights_path = '/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5'
352
353 # Create the VGG16 model with custom weights path
354 base_model = VGG16(weights=vgg16_weights_path, include_top=False, input_shape=(224, 224, 3))
```

```
349 # Stage 4: Model Training
350 # Path to the VGG19 weights file
351 vgg19_weights_path = '/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5'
352
353 # Create the VGG19 model with custom weights path
354 base_model = VGG19(weights=vgg19_weights_path, include_top=False, input_shape=(224, 224, 3))

366 # Load ResNet50 base model with pre-trained ImageNet weights
367 base_model = ResNet50(
368     weights='/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5',
369     include_top=False,
370     input_shape=IMG_SIZE + (3,))
371 )
372
```

Figure III. 30 : Chargement des réseaux de neurones qui vont être utilisés pour le Transfer Learning

Le prétraitement de VGG16, VGG19 et ResNet est utilisé, et voici comment nous affichons les traits (caractéristiques) des premières couches en utilisant des paramètres déjà formés que nous allons geler :

```
# Set the number of classes
NUM_CLASSES = 1

# Build the model
model = Sequential()
model.add(base_model)
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(NUM_CLASSES, activation='sigmoid'))

# Freeze the ResNet50 layers
model.layers[0].trainable = False

# Compile the model
model.compile(
    loss='binary_crossentropy',
    optimizer=RMSprop(lr=1e-4),
    metrics=['accuracy']
)

# Print model summary
model.summary()
```

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23587712
flatten (Flatten)	(None, 100352)	0
dropout (Dropout)	(None, 100352)	0
dense (Dense)	(None, 1)	100353
Total params: 23,688,065 Trainable params: 100,353 Non-trainable params: 23,587,712		

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 7, 7, 512)	20024384
flatten (Flatten)	(None, 25088)	0
dropout (Dropout)	(None, 25088)	0
dense (Dense)	(None, 1)	25089
Total params: 20,049,473 Trainable params: 25,089 Non-trainable params: 20,024,384		

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14714688
flatten (Flatten)	(None, 25088)	0
dropout (Dropout)	(None, 25088)	0
dense (Dense)	(None, 1)	25089
Total params: 14,739,777 Trainable params: 25,089 Non-trainable params: 14,714,688		

Figure III. 31 : Visualisation du résumé du modèle

III.8 L'entraînement des réseaux de classification :

Nous décidons comment configurer l'optimiseur Adam et la fonction d'activation sigmoïde avant de commencer la formation.

A l'aide du modèle, nous construisons le modèle. Compile()* accepte trois paramètres : un optimiseur (Optimizer), une perte (loss), et un ensemble de métriques (metrics).

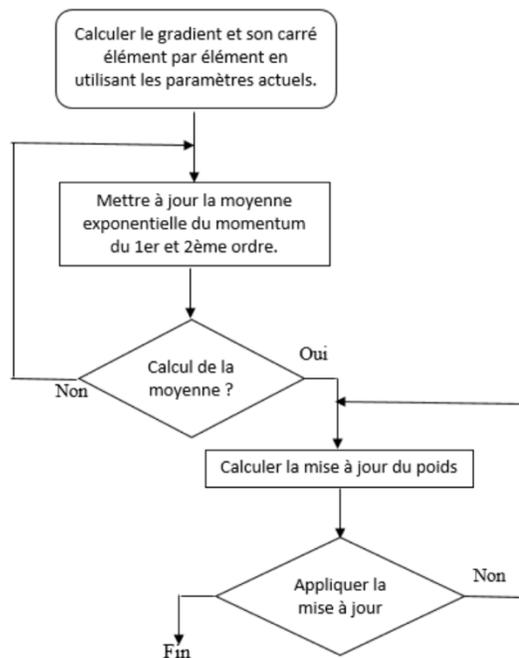


Figure III. 32 : Organigramme de l'optimiseur ADAM

Il y a une étape d'ajustement du biais dans l'optimiseur Adam. Le taux d'apprentissage, le taux de décroissance du moment du premier ordre et le taux de décroissance du moment du second ordre sont les trois hyperparamètres requis. Avec Keras, la configuration pour choisir ces paramètres est la suivante :

```
425 # Freeze base model layers
426 for layer in base_model.layers:
427     layer.trainable = False
428 # Create top model
429 top_model = base_model.output
430 top_model = Flatten()(top_model)
431 top_model = Dense(256, activation='relu')(top_model)
432 top_model = Dropout(0.5)(top_model)
433 top_model = Dense(1, activation='sigmoid')(top_model)
434
435 # Combine base and top models
436 model = Model(inputs=base_model.input, outputs=top_model)
437 # Compile the model
438 model.compile(loss='binary_crossentropy', optimizer=Adam(lr=0.001), metrics=['accuracy'])
439 # Set up early stopping
440 early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)
```

Figure III. 33 : code de configuration de la fonction d'activation et l'optimiseur ADAM

Ensuite, On exécute avec * model.fit_generator () *

On choisit les Epoques = 60, le pas dans les Epoques = 6 et on lance l'entraînement :

VGG16:

```
Epoch 51/60
6/6 [=====] - 50s 8s/step - loss: 0.1189 - accuracy: 0.9415 - val_loss: 0.4385 - val_accuracy: 0.8750
Epoch 52/60
6/6 [=====] - 50s 9s/step - loss: 0.1392 - accuracy: 0.9574 - val_loss: 0.3529 - val_accuracy: 0.8438
Epoch 53/60
6/6 [=====] - 50s 8s/step - loss: 0.2596 - accuracy: 0.9149 - val_loss: 0.3229 - val_accuracy: 0.8750
Epoch 54/60
6/6 [=====] - 50s 9s/step - loss: 0.1609 - accuracy: 0.9574 - val_loss: 0.3483 - val_accuracy: 0.8438
Epoch 55/60
6/6 [=====] - 50s 8s/step - loss: 0.1146 - accuracy: 0.9628 - val_loss: 0.3547 - val_accuracy: 0.8438
Test Loss: 1.061168909072876
Test Accuracy: 0.8999999761581421
1/1 [=====] - 2s 2s/step
Accuracy: 0.9
```

VGG19:

```
Epoch 56/60  
6/6 [=====] - 62s 11s/step - loss: 0.1424 - accuracy: 0.9468 - val_loss: 0.3704 - val_accuracy: 0.8438  
Epoch 57/60  
6/6 [=====] - 62s 10s/step - loss: 0.2305 - accuracy: 0.9255 - val_loss: 0.1650 - val_accuracy: 0.9062  
Epoch 58/60  
6/6 [=====] - 62s 11s/step - loss: 0.1078 - accuracy: 0.9415 - val_loss: 0.3333 - val_accuracy: 0.8125  
Epoch 59/60  
6/6 [=====] - 62s 11s/step - loss: 0.1573 - accuracy: 0.9681 - val_loss: 0.3189 - val_accuracy: 0.8125  
Epoch 60/60  
Accounts: 0.3059195876121521  
Test Accuracy: 0.8999999761581421  
1/1 [=====] - 3s 3s/step  
Accuracy: 0.9
```

ResNet50:

```
Epoch 43/60  
6/6 [=====] - 22s 4s/step - loss: 0.0976 - accuracy: 0.9681 - val_loss: 0.3416 - val_accuracy: 0.9062  
Epoch 44/60  
6/6 [=====] - 22s 4s/step - loss: 0.2840 - accuracy: 0.9255 - val_loss: 0.3142 - val_accuracy: 0.9062  
Epoch 45/60  
6/6 [=====] - 22s 4s/step - loss: 0.2272 - accuracy: 0.9096 - val_loss: 0.3190 - val_accuracy: 0.9062  
Epoch 46/60  
6/6 [=====] - 22s 4s/step - loss: 0.1779 - accuracy: 0.9309 - val_loss: 0.2994 - val_accuracy: 0.9062  
Epoch 47/60  
6/6 [=====] - 22s 4s/step - loss: 0.1381 - accuracy: 0.9362 - val_loss: 0.2806 - val_accuracy: 0.9062  
Test Loss: 0.30237889289855957  
Test Accuracy: 0.8999999761581421  
1/1 [=====] - 2s 2s/step  
Accuracy: 0.9
```

Figure III. 34 : Visualisation de l'entraînement

III.8 .1 Résultats et discussion:

VGG16 :

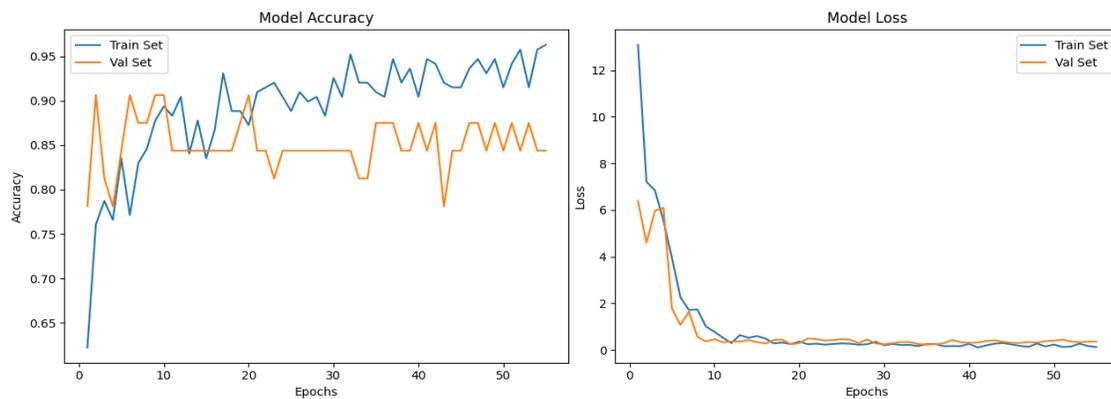


Figure III. 35 : Lacourbe de perte (loss) et la courbe de précision (accuracy) du réseau

VGG 19 :

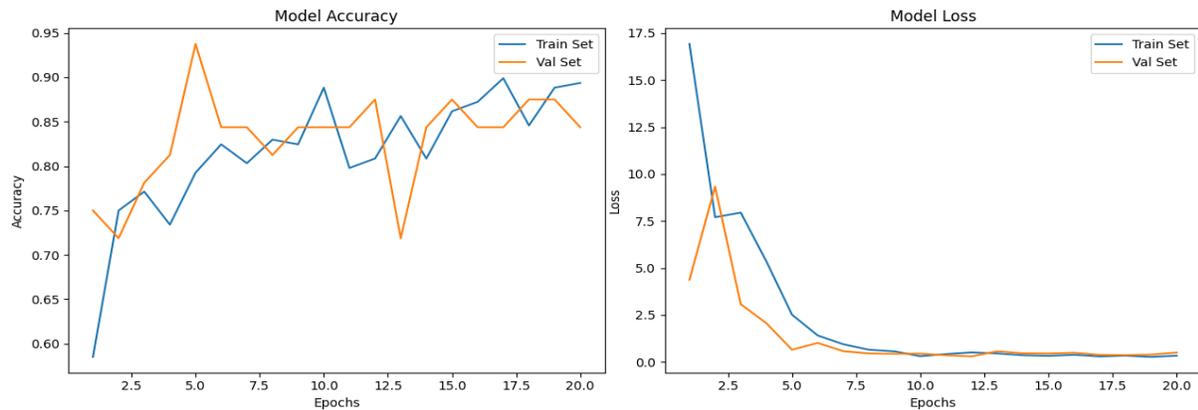


Figure III. 36 : Lacourbe de perte (loss) et la courbe de précision (accuracy) du réseau

ResNet 50 :

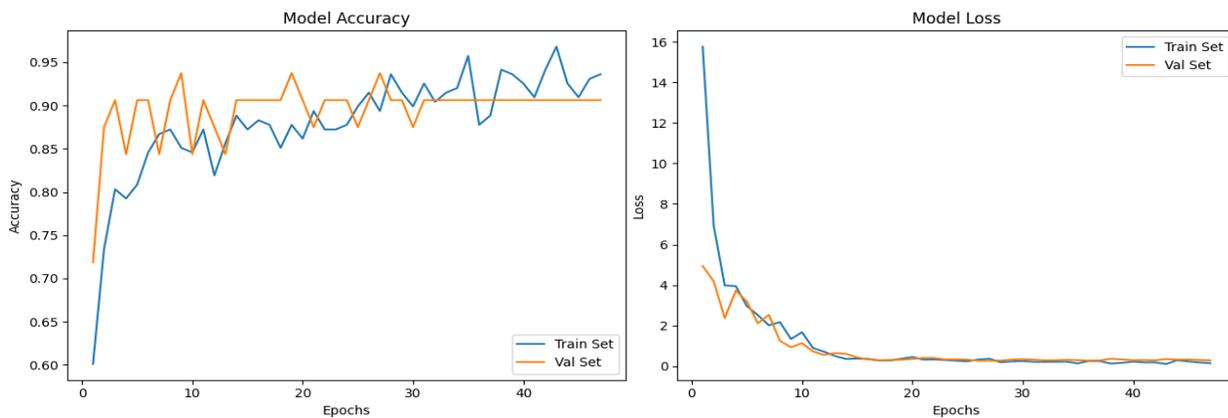


Figure III. 37 :Lacourbe de perte (loss) et la courbe de précision (accuracy) du réseau

D'après les résultats, il est clair que chaque réseau fonctionne différemment. Le ResNet est le plus performant ; son tracé de la perte d'entraînement (Train Loss) diminue jusqu'à un point stable, et celui de la perte de validation (ValidationLoss) diminue également jusqu'à un certain point, affichant une légère différence par rapport à la perte d'entraînement. « L'écart de généralisation » est le nom donné à cet écart.

Le modèle ResNet a bien fonctionné avant l'achèvement pour toutes les époques car il s'est arrêté à l'époque= 47.

Les réseaux VGG16 et VGG19 diffèrent légèrement du ResNet en ce sens qu'ils ont bien appris les données d'entraînement (Train Data set) et produit des résultats généralement favorables lorsqu'ils sont appliqués aux nouvelles données de validation (Validation Data set).

Ce phénomène est connu sous le nom de "surajustement", qui peut se produire pour diverses raisons, mais dans notre cas, on pense que c'est parce que l'ensemble de données utilisé pour réaliser notre projet est petit (environ 5000 images après l'augmentation des données) par rapport à d'autres ensembles de données. L'explication peut être que les modèles ont parfaitement appris l'ensemble de données d'apprentissage (y compris le bruit statistique ou les fluctuations aléatoires dans l'ensemble de données d'apprentissage), au point de ne pas pouvoir généraliser aux nouvelles données de validation.

Étant donné que le taux de précision des modèles que nous avons utilisés était de 90 %, ce qui est un chiffre assez décent, nous pouvons dire qu'ils ont produit des résultats acceptables.

III.8 .2 La phase de test

Enregistrer le modèle :

```
513 |  
514 # Save the model  
515 model.save('/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/preview/2023-06-17_ResNet50_model.h5')  
516
```

Figure III. 38 : code d'Enregistrer le modèle

III.8 .3 Entrez le chemin d'accès à l'image :

Demander à entrer l'image, faire le traitement nécessaire, puis la faire la détection avec le nouveaumodèle

```
523 # Load the trained CNN model
524 model = tf.keras.models.load_model('/Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/preview/2023-06-12_VGG_model.h5')
525
526 # Preprocess the input image
527 def preprocess_image(image_path):
528     image = Image.open(image_path)
529     image = image.convert('RGB') # Convert the image to RGB format
530     image = image.resize((224, 224))
531     image = np.array(image) / 255.0
532     image = np.expand_dims(image, axis=0)
533     return image
534
535 # Perform prediction
536 def predict_tumor(image_path):
537     image = preprocess_image(image_path)
538     prediction = model.predict(image)[0][0]
539     return prediction
540
541 # Classify the prediction
542 def classify_prediction(probability, threshold=0.5):
543     if probability > threshold:
544         return 'Tumor Present'
545     else:
546         return 'Tumor Absent'
547
548 # Get user input (path to the image)
549 image_path = input("Enter the path to the image: ")
550
551 # Perform prediction and classification
552 tumor_probability = predict_tumor(image_path)
553 prediction = classify_prediction(tumor_probability)
554
555 # Display the prediction result
556 print("Prediction: " + prediction)
557
```

Figure III. 39 : lecode de détection avec le nouveaumodel

III.8 .3.1 Le résultatdedétection avec le nouveaumodel :

```
Enter the path to the image: /Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive5/Testing/no_tumor/image(100).jpg
1/1 [=====] - 0s 457ms/step
Prediction: Tumor Absent
mac@MacBook-Pro-de-MAC ~ %
```

```
Enter the path to the image: /Users/mac/Desktop/HammouFE/mastre2/memioire/S2IRM/database/archive_3/testing_final /yes/Y147.JPG
1/1 [=====] - 0s 436ms/step
Prediction: Tumor Present
```

Figure III. 40 : Exemple d'image avec ou sans tumeur

III.9 Conclusion

Dans ce chapitre, nous avons présenté l'environnement de notre travail, nous avons ainsi présenté les modèles sur lesquels nous nous sommes basés pour explorer la base de données « Kaggle ». Nous avons pu achever la phase de détection de la tumeur et ceci grâce à l'amélioration des modèles de l'apprentissage profond. Nous avons présenté ainsi, les différents résultats que nous avons obtenus, ceci nous a permis de comparer les modèles et de déduire que le dernier est le mieux adapté à notre étude. Avec un ensemble de données plus important, nous pouvons obtenir des résultats plus précis. Enfin, l'utilisation efficace de l'apprentissage automatique et de l'apprentissage profond dans le domaine médical sera un atout majeur pour des études autour.

Conclusion générale

Conclusion

La segmentation des images médicales est un domaine résistible. Il sert à plusieurs fins, telles que l'aide au diagnostic, la surveillance de l'état des patients et les examens cliniques.

L'objectif de cette étude était d'un système pour classer les images d'IRM cérébrale et identifier les régions de tumeurs cérébrales primaires les plus courantes

Nous avons mis en évidence un certain nombre de concepts et de définitions d'apprentissage en profondeur que nous avons utilisés pour faire notre travail ... [49]

Nous avons créé une base de formation, l'avons annotée et mise en œuvre un réseau d'apprentissage en profondeur utilisant convolutional VGG16, VGG19 et ResNet 50, avec des résultats satisfaisants. La prédéfinition des paramètres du réseau est difficile. Pour cette raison, nous avons défini différents modèles avec différentes architectures pour obtenir les meilleurs résultats en termes de précision et d'erreur.

Perspectives :

Plusieurs perspectives peuvent être envisagées dans la prolongation de ce mémoire, nous pouvons citer :

- *Tester d'autres bases de données plus grandes.*
- *Utiliser des architectures plus profondes.*
- *Développer une architecture de segmentation des tumeurs à partir des images 3D. [48]*

Références bibliographiques

Références bibliographiques

- [1] : Les Cancers du cerveau collection comprendre et agir. Fondation ARC pour la recherche sur le cancer.
- [2] : Anatomie et physiologie du système nerveux en général, et du cerveau en particulier : Atlas
- [3] : <http://www.psychomedia.qc.ca/neurologie/2009-06-18/matiere-grise-et-matiere-blanche-du-cerveau-definitions>)
02-06-2022
- [4] : Le cerveau et ses fonctions (Vol. 19). Baillière.
- [5] : Mémoire en Master instrumentation médicale de BOUZIDI Wafaa (analyse et traitement du signal electroencephalogramme) 2020/2021 UNIVERSITE TLEMCEN
- [6] : (<https://www.passeportsante.net/fr/parties-corps/Fiche.aspx?doc=crane-savoir-sur-cette-partie-corps>)
- [7] : <https://www.e-cancer.fr/Patients-et-proches/Les-cancers/Tumeurs-du-cerveau/Le-cerveau/Les-differentes-parties>
- [8] : (<https://www.futura-sciences.com/sante/dossiers/medecine-voyage-cerveau/>)
- [9] : (<https://institutducerveau-icm.org/fr/actualite/comprendre-le-cerveau-et-son-fonctionnement/>)
- [10]: <https://www.msmanuals.com/fr/accueil/troubles-du-cerveau,-de-la-moelle-%C3%A9pini%C3%A8re-et-des-nerfs/maladies-des-nerfs-cr>, Par **Michael Rubin** ,
MDCM, New York Presbyterian Hospital-Cornell Medical Center
- [11]: (<https://www.fondation-arc.org/cancer/cancer-cerveau/symptomes-diagnostic-cancer>)

[12]:(http://fragile.ch/fileadmin/user_upload/downloads/01_leben_mit_HV/Tumeurs_cerebrales_et_SNC.pdf).

[13]: Thèse pour l'obtention du Grade de Docteur de l'université de Poitiers (Faculté des Sciences Fondamentales et Appliquées) (Diplôme National - Arrêté du 25 avril 2002). Université de Poitiers, 2003.

[14]: PFE pour obtenir le diplôme de master académique en Informatique 27/06/2022 Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie Biskra

[15]: Mémoire de Master en électronique des Systèmes Embarqués 2019-2020 (Traitement des images IRM pour la détection des tumeurs cérébrales par les algorithmes de Deep Learning CNN, Faster RCNN, Mask R-CNN et Transfer Learning sous environnement Cloud) Université SAAD DAHLAB de BLIDA

[16]: Mémoire de master en imagerie médicale 2021/2022 de BERDI Yasser (Détection Précoce Des Maladies Neurodégénératives) UNIVERSITE TLEMCEN

[17]: Mémoire en Master instrumentation médicale (analyse et traitement de signal électroencephalogramme) 2020/2021 UNIVERSITE TLEMCEN

[18]: <https://sante.journaldesfemmes.fr/fiches-anatomie-et-examens/2791337-os-occipital-occiput-definition-schema/>

[19]: <https://www.neuromedia.ca/quels-sont-les-12-nerfs-craniens>

[20]: S. Horowitz et T. Pavlidis: Picture segmentation by a directed split-and-merge procedure. Rapport technique, Department of Electrical Engineering, Princeton University, 1975.

[21]: J. Cocquerez and S. Philips, "Analyse d'Images : Filtrage et Segmentation", Masson, 1995.

[22] : SALIMA NEBTI, OPTIMISATION PAR ECOSYSTEMES ARTIFICIELS APPLICATION A LA SEGMENTATION D'IMAGES.

[23] : N Azouza, Segmentation par classification floue : application à l'imagerie par résonance magnétique, Thèse de master en Génie Électrique à UNIVERSITE AKLI MOAND OULHADJE-BOUIRA, 2018

[24] : G. Sarah, "Segmentation d'images couleurs par morphologie mathématique : application aux images microscopiques," Thèse de doctorat en science à l'université Toulouse 3 Paul Sabatier, 2010.

[25] : A. Nakib, "Conception de méta heuristiques d'optimisation pour la segmentation d'images : Application à des images biomédicales", Thèse de doctorat en science à l'université Paris 12-Valde De Marne, 2008.

[26] : " Fusion et fouille de données guidées par les connaissances : application à l'analyse d'image ", Thèse de doctorat en science à L'université Mohammed Khider –Biskra, 2013.

[27] : S. Ouadfel, " Contributions à la Segmentation d'images basées sur la résolution collective par colonies de fourmis artificielles", Thèse de doctorat en informatique à l'université Hadj Lakhdar de Batna, 2006.

[28] : A. Trémiau & P. Colantoni, "Regions adjacency graph applied to color image segmentation". IEEE Transactions on Image Processing, Vol, 9 N° 4, pp : 735–744, 2000.

[29] : C. Meurie, "Segmentation d'images couleur par classification pixellaire et hiérarchie de partitions ", Thèse de doctorat en informatique à l'université De Cean , 2005.

[30] : A. K. Jain, M.N. Murty, P.J. Flynn, " Data clustering: a review. ACM Computing Surveys", Vol. 31, N° 3, pp: 264-322, 1999.

[31] : A. Clément, "Algorithmes et outils informatiques pour l'analyse d'images couleurs: Application à l'étude de coupes histologiques de baies de raisin en microscopie optique", Thèse de doctorat en science à l'école doctorale d'angers, 2002.

[32]: J. C. Bezdek, "Pattern recognition with fuzzy objective functional algorithms", Kluwer Academic Editions, 1981.

[33] : Yanhao.H, Lihui.X, Chuanze.K, Minghui.W, Qin.M, Bin.Y, "SGL-SVM: A novel method for tumor classification via support vector machine with sparse group Lasso", Journal of Theoretical Biology Vol 486, 7 Feb 2020.

[34]: http://staff.univbatna2.dz/sites/default/files/merzougui_ghalia/files/support_de_cours_-deep_learning-chapitre3-cnn.pdf.

[35] : CH OTMANI, Détection de tumeur cérébrale avec l'apprentissage profond, Thèse de master en informatique à Université Mohamed Khider – BISKRA, 2021

[36] : https://slds-lmu.github.io/seminar_nlp_ss20/introduction-transfer-learning-for-nlp.htm

[37]: Khan A, Sohail A, Zahoor U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks. ArtifIntell Rev. 2020;53(8):5455–516.

[38]: Shrestha A, Mahmood A. Review of deep learning algorithms and architectures. IEEE Access. 2019;7:53040–65.

[39]: <https://freelance-france.com/comment-utiliser-visual-studio-code-afin-de-creer-une-application/>

[40]: Luis Perez, Jason Wang - The Effectiveness of Data Augmentation in Image Classification using Deep Learning – publié le 13 Décembre 2017

[41] Qiang Yang, SinnoJialin Pan, “A Survey on Transfer Learning”, IEEE Transactions on Knowledge & Data Engineering, vol. 22, no, pp. 1345–1359, October 2010, doi:10.1109/TKDE.2009.191

[42] Karen Simonyan, Andrew Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition, modifié le 10 Avril 2015

[43] Rohit. T. Step by step VGG16 implementation in Keras for beginners.
<https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c>.

[45] :<https://datagen.tech/guides/computer-vision/resnet-50/>

[46] :<https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>

[47] : Mémoire de Master en Sciences et sciences appliquées (Segmentation par classification floue : application à l'imagerie par résonance Magnétique) 2017/2018
UNIVERSITE AKLI MOAND OULHADJE-BOUIRA

[48] : B. Salima, B. Khadidja, ‘Segmentation des images IRM cérébrales par apprentissage profond’, Mémoire de Master en imagerie médicale de universite de tlemcen

[49] : KH. ELYES, Mémoire de Master en Électronique des Systèmes Embarqués 2019-2020 Université SAAD DAHLAB de BLIDA

