

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



Université A/Mira de Béjaïa  
Faculté des Sciences Exactes  
Département d'Informatique

## MÉMOIRE DE MASTER RECHERCHE

En  
Informatique

Option  
*Intelligence Artificielle*

### Thème

**Etude comparative d'Algorithmes de Deep Learning  
pour le traitement d'images IRM pour la détection  
des tumeurs du cerveau**

Présenté par : Makhlouf Nour Melissa et Meziani Malika

Soutenu le 19 Juin 2023 devant le jury composé de :

Présidente	Dr BOUKERRAM Samira	Maître assist. A	U. Béjaïa.
Encadrant	Pr AMROUN Kamal	Professeur	U. Béjaïa.
Co-Encadrante stage	Pr OURRAD Dehbia	Professeur	U. Alger.
Examinatrice	Dr TAHAKOURT Zineb	Maître assist. A	U. Béjaïa.

## *\* Remerciements \**

Nous aimerions remercier nos parents pour leur patience et leurs encouragements tout au long de ce travail.

Nous tenons à remercier également notre honorable encadrant Professeur AMROUN KAMAL pour sa disponibilité, son partage du savoir, sa confiance, son aide et sa patience.

Nous voulons aussi remercier, Docteur OURRAD DEHBIA, pour sa disponibilité et son aide ainsi que son orientation dans le domaine médical.

Nous remercions également les membres du jury d'avoir accepté d'évaluer notre travail.

Enfin merci à Maluma et Shakira de nous avoir donné de l'énergie tout au long de ses mois de travail et de nous avoir diverti dans les temps de stress.

*M.lle Meziani Malika & M.lle Makhlouf Nour Melissa*

※ *Dédicaces* ※

A la mémoire des mes tres chères grand parents, Yayas et Jedis, qui nous ont quitté.

A mon petit frere qui est loin et qui n'a pas pu assisté à cette soutenance.

*Mlle. Meziani Malika*

A ma soeur ainée qui est loin et n'a pas pu assisté à cette soutenance.

*Mlle Makhlouf Nour Melissa*

# Table des matières

Table des matières	i
Table des figures	v
Liste des tableaux	viii
Notations Acronymes	ix
Introduction générale	1
<b>1 Tumeurs et Cancer du Cerveau</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.2 C'est quoi une tumeur? . . . . .	5
1.3 Quels sont les types de tumeurs? . . . . .	5
1.3.1 Les cancers "solides" . . . . .	5
1.3.2 Les cancers "liquides" ou sanguins . . . . .	5
1.3.3 Les cancers "métastatiques" ou disséminés . . . . .	6
1.3.4 Les cancers secondaires . . . . .	6
1.4 C'est quoi une tumeur du cerveau? . . . . .	6
1.4.1 C'est quoi le cancer du cerveau? . . . . .	7
1.5 Quels sont les types de tumeurs du cerveau? . . . . .	7
1.6 Degré d'agressivité d'une tumeur . . . . .	8
1.7 Quels symptômes? . . . . .	8
1.8 Quelles causes? . . . . .	9
1.8.1 L'environnement . . . . .	10
1.8.2 L'hérédité . . . . .	10
1.8.3 Les autres maladies . . . . .	10
1.9 Prévention . . . . .	11
1.10 Diagnostic . . . . .	11
1.10.1 L'examen clinique . . . . .	11
1.10.2 Les examens d'imagerie . . . . .	12
1.10.3 L'examen anatomopathologique (biopsie) . . . . .	12

---

1.11	Traitement d'une tumeur du cerveau . . . . .	13
1.12	Conclusion . . . . .	13
<b>2</b>	<b>L'imagerie par résonance magnétique</b>	<b>14</b>
2.1	Introduction . . . . .	14
2.2	Evolution de l'imagerie médical . . . . .	14
2.3	L'exploration radiologique . . . . .	15
2.4	Imagerie Par Résonance Magnétique . . . . .	17
2.4.1	Qu'est ce que c'est ? . . . . .	17
2.4.2	Objectifs de l'IRM . . . . .	17
2.4.3	En pratique . . . . .	17
2.5	Composition de l'appareil d'IRM . . . . .	18
2.6	Avantages de l'IRM . . . . .	19
2.7	Existe-t-il des risques ? . . . . .	19
2.8	Inconvénients de l'IRM . . . . .	19
2.9	L'importance d'une IRM cérébrale . . . . .	19
2.10	Conclusion . . . . .	20
<b>3</b>	<b>Intelligence Artificielle, Machine Learning et Deep Learning</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	Intelligence Artificielle . . . . .	22
3.2.1	Définition . . . . .	22
3.2.2	L'intelligence artificielle dans le milieu médical . . . . .	23
3.3	Machine Learning . . . . .	23
3.3.1	Types d'apprentissage automatique . . . . .	24
3.4	Deep learning . . . . .	25
3.5	Les réseaux de neurones . . . . .	26
3.6	Les différentes architectures du Deep Learning . . . . .	27
3.7	Les différents secteurs d'activité des DL . . . . .	29
3.8	Les réseaux de neurones convolutifs- Convolutional Neural Network CNN . . . . .	30
3.8.1	Définition . . . . .	30
3.8.2	But . . . . .	31
3.8.3	Fonctionnement d'un CNN . . . . .	31
3.8.4	Architecture d'un CNN . . . . .	32
3.8.5	Quelques réseaux CNN . . . . .	41
3.9	Transfer Learning . . . . .	42
3.10	Overfitting . . . . .	43
3.11	Conclusion . . . . .	43

---

<b>4</b>	<b>Etat de l'art</b>	<b>44</b>
4.1	Introduction . . . . .	44
4.2	Synthèse bibliographique . . . . .	50
4.3	Conclusion . . . . .	50
<b>5</b>	<b>Approche utilisée et Conception</b>	<b>51</b>
5.1	Introduction . . . . .	51
5.2	Modèles utilisés pour le Transfer Learning . . . . .	52
5.2.1	Premier modèle : Inception V3 . . . . .	52
5.2.2	Deuxième modèle : GoogLeNet . . . . .	53
5.2.3	Troisième et Quatrième modèles : VGG16 et VGG19 . . . . .	53
5.2.4	Cinquième modèle : ResNet50 . . . . .	54
5.3	Comment choisir les modèles? . . . . .	55
<b>6</b>	<b>Implémentation et Résultats</b>	<b>56</b>
6.1	Introduction . . . . .	56
6.2	Algorithmes Implémentés . . . . .	56
6.3	Ensemble de données d'entraînement de validation et de test . . . . .	59
6.3.1	Description des datasets . . . . .	59
6.3.2	Redimensionner les Images . . . . .	59
6.3.3	Recadrage et Detection . . . . .	60
6.4	Implémentation . . . . .	61
6.4.1	Langage de Programmation utilisé . . . . .	61
6.4.2	Frameworks, IDE et Bibliothèques Python . . . . .	61
6.5	Importation des bibliothèques . . . . .	65
6.6	Metriques d'évaluation des performances . . . . .	66
6.7	Connexion au GoogleColab . . . . .	67
6.8	Préparation des Données . . . . .	67
6.9	Augmentation des données . . . . .	68
6.10	Transfer Learning . . . . .	68
6.11	Compilation du modèle . . . . .	69
6.12	Entraînement du modèle . . . . .	69
6.13	Test du modèles . . . . .	69
6.14	Résultats . . . . .	72
6.14.1	Dataset 1 . . . . .	72
6.14.2	Dataset 2 . . . . .	73
6.15	Discussion . . . . .	74

<b>Conclusion et perspectives</b>	<b>76</b>
-----------------------------------	-----------



# Table des figures

1.1	Symptome d'une tumeur cerebrale . . . . .	9
1.2	Examen de biopsie . . . . .	12
2.1	Évolution de l'imagerie Médicale (source :[45]) . . . . .	15
2.2	Radio . . . . .	15
2.3	Scanner . . . . .	16
2.4	IRM . . . . .	16
2.5	Appareil IRM . . . . .	18
3.1	Evolution de l'IA . . . . .	22
3.2	Intelligence artificielle . . . . .	22
3.3	Machine learning . . . . .	23
3.4	apprentissage supervisé et non supervisé . . . . .	25
3.5	Architecture Deep Learning . . . . .	25
3.6	Machine Learning VS Deep learning . . . . .	26
3.7	Tableau comparatif de ML et DL . . . . .	30
3.8	Fonctionnement d'un CNN . . . . .	32
3.9	Couche de convolution . . . . .	33
3.10	Average Pooling vs Max Pooling . . . . .	34
3.11	Couche Fully Connected . . . . .	35
3.12	Graphe de la fonction ReLU . . . . .	37
3.13	Graphe fonction sigmoide . . . . .	37
3.14	Graphe de la fonction softmax . . . . .	38
3.15	Graphe de la fonction tanh . . . . .	39
3.16	Graphe de la fontion ELU . . . . .	39
3.17	Graphe de la fonction SELU . . . . .	40
3.18	Fonctionnement de la couche Dropout . . . . .	41
3.19	Architecture LeNet . . . . .	42
3.20	Architecture AlexNet . . . . .	42
3.21	Overfitting . . . . .	43
4.1	Structure du modele ameliore . . . . .	44

4.2	Couches Ajoutées . . . . .	45
4.3	Valeurs principales utilisées lors du training des réseaux . . . . .	45
4.4	Validation Accuracy Model . . . . .	45
4.5	Taux de Perte . . . . .	46
4.6	Critere de performance . . . . .	46
4.7	Precision et valeurs des modeles . . . . .	46
4.8	Images utilisees . . . . .	47
4.9	Algorithmes utilises . . . . .	47
4.10	Classificateurs utilises . . . . .	48
4.11	Architecture du modele de CNN propose . . . . .	48
4.12	Metriques d'evaluation . . . . .	49
4.13	Graphes de Precision des algorithmes etudies . . . . .	49
4.14	Courbe ROC-AUC . . . . .	50
5.1	Architecture InceptionV3 . . . . .	52
5.2	Architecture GoogLeNet . . . . .	53
5.3	Architecture VGG16 et VGG19 . . . . .	54
5.4	Architecture ResNet . . . . .	55
6.1	VGG 16 . . . . .	57
6.2	VGG 19 . . . . .	57
6.3	resnet50 . . . . .	58
6.4	Inception V3 . . . . .	58
6.5	Resize . . . . .	59
6.6	Recadrage . . . . .	60
6.7	Logo Python (source :[13]) . . . . .	61
6.8	Google Colab (source :[14]) . . . . .	61
6.9	Keras (source :[15]) . . . . .	62
6.10	TensorFlow (source :[16]) . . . . .	62
6.11	SciPy (source :[17]) . . . . .	63
6.12	Pandas (source :[38]) . . . . .	63
6.13	Numpy (source :[18]) . . . . .	64
6.14	Scikit Learn (source :[19]) . . . . .	64
6.15	Matplotlib (source :[20]) . . . . .	64
6.16	Importation des Bibliotheques . . . . .	65
6.17	Metriques d'evaluation des performances . . . . .	67
6.18	Connexion à google colab . . . . .	67
6.19	Preparation des donnees . . . . .	67
6.20	Augmentation . . . . .	68
6.21	Transfer Learning ResNet50 . . . . .	68

---

6.22	Transfer Learning VGG16 . . . . .	68
6.23	Transfer Learning VGG19 . . . . .	69
6.24	Compilation du modele . . . . .	69
6.25	Entrainement de Resnet50 . . . . .	69

# Liste des tableaux

4.1	Tableau de Synthèse . . . . .	50
6.1	Matrices VGG16 . . . . .	70
6.2	Matrices VGG19 . . . . .	70
6.3	Matrices ResNet 50 . . . . .	71
6.4	Matrices InceptionV3 . . . . .	71
6.5	Matrices GoogLeNet . . . . .	72
6.6	Tableau des tests du dataset1 . . . . .	72
6.7	Tableau des résultats du dataset1 . . . . .	73
6.8	Tableau des tests du dataset 2 . . . . .	73
6.9	Tableau des résultats du dataset1 . . . . .	73

# Notations et Acronymes

<b>A</b>	<i>AE</i>	Autoencodeurs.
<b>B</b>	<i>BM</i>	Boltzman Machine.
<b>C</b>	<i>CNN</i>	Conventional Neural Network.
<b>D</b>	<i>DBN</i>	Deep Belief Network.
	<i>DL</i>	Deep Learning.
	<i>DNN</i>	Deep Neural Network.
<b>E</b>	<i>ELU</i>	Exponential Linear Unit.
<b>F</b>	<i>FC</i>	Fully Connected.
<b>G</b>	<i>GAN</i>	Generative Adversarial Network.
	<i>GPU</i>	Graphical Processing Unit.
	<i>GRU</i>	Gated Recurrent Unit.
<b>I</b>	<i>IA</i>	Intelligence Artificielle.
	<i>IM</i>	Imagerie médicale.
	<i>IRM</i>	Imagerie par résonance magnétique.
<b>L</b>	<i>LSTM</i>	Long Short Term Memory.
<b>M</b>	<i>ML</i>	Machine Learning.
<b>N</b>	<i>NLP</i>	Natural Language Processing.
<b>R</b>	<i>RBM</i>	Restricted Boltzman Machine.
	<i>ReLU</i>	Rectified Linear Unit.
	<i>RNN</i>	Recurring Neural Network.
	<i>RVB</i>	Rouge Vert Bleu.
<b>S</b>	<i>SAE</i>	Stacked auto encoder.
	<i>SDAE</i>	Stacked Denoising Auto Encoder.
	<i>SELU</i>	Scaled Exponential Linear Unit.
	<i>SNC</i>	Système nerveux central.
<b>T</b>	<i>tanh</i>	Tangente Hyperbolique.
	<i>TEP</i>	Tomographie par Emission de Positons.
	<i>TDM</i>	Tomodensitometrie.
<b>V</b>	<i>VAE</i>	Variational auto encoder.
	<i>VGG</i>	Visual Geometry Group.

# Introduction générale

## Contexte

Durant les cinq dernières années, l'intelligence artificielle (IA) est devenue un sujet de discussion populaire suscitant la curiosité d'un bon nombre de personnes. Apparue dans les années 1940 lors d'une conférence de l'université de Dartmouth, avec pour objectif d'imiter l'intelligence humaine, son développement stagnera pendant des décennies par manque de ressources avant de reprendre son souffle dans les années 2000 grâce au développement d'Internet et des GPUs.

L'apprentissage automatique (en anglais Machine Learning, ML), est un domaine de l'intelligence artificielle qui apparaît suite à la volonté d'automatiser les tâches des ordinateurs grâce à des algorithmes, dû au développement des systèmes de tâches complexes tels que l'arrivée de l'internet, le traitement de données massives, l'avancée de l'imagerie médicale, etc.

Dans le domaine de la santé, elle est utilisée pour développer des traitements personnalisés, découvrir de nouveaux médicaments, ou encore pour analyser les imageries médicales telles que les rayons X et les IRM. Des assistants virtuels peuvent aussi venir en aide aux patients en leur rappelant de prendre leurs médicaments ou pilules ou de faire du sport pour rester en forme

Les méthodes Deep Learning (DL) sont une famille de techniques informatiques qui forment une sous branche du Machine Learning (ML). Celles ci ont été popularisées surtout par les applications d'imagerie.

L'objectif du DL est de modéliser le cerveau pour ne pas traiter l'information avec des règles strictes, mais plutôt en utilisant la pratique et la répétition, de la même manière qu'un enfant apprend à découvrir le monde

Le Deep Learning décrit un ensemble de modèles informatiques constitués de plusieurs couches de traitement de données, permettant l'apprentissage en représentant ces données en termes d'un certain niveau d'abstraction. Le deep learning se base sur les réseaux de neurones artificiels imitant les réseaux de neurones biologiques. Il donnera naissance à plusieurs techniques d'apprentissages, gérées par des réseaux de neurones artificiels profonds.

Les réseaux de neurones convolutifs (CNN) sont une approche spécifique du DL, dédiés au traitement d'images comme par exemple, l'imagerie médicale dans la détection de maladies.

L'imagerie médicale, à l'heure actuelle, se base sur plusieurs techniques différentes chacune favorisée à différents domaines.

Dans le domaine de la santé, l'imagerie par résonance magnétique est habituellement très sûre. C'est une technologie d'imagerie médicale non invasive utilisée dans l'étude de l'anatomie et du fonctionnement de l'organisme. L'IRM n'utilise pas de rayons X ionisants nocifs .

Elle est souvent utilisée pour la détection des maladies ainsi que le diagnostic et suivi de la prise en charge thérapeutique. Elle repose sur une technologie sophistiquée qui excite et détecte les modifications des protons présents dans l'eau qui compose les tissus vivants. Elle utilise un champ magnétique puissant et des ondes radio de très haute fréquence. Elle produit ainsi des images haute résolution de l'intérieur du corps qui aident à diagnostiquer une grande variété de problèmes.

L'IRM est le meilleur moyen d'identifier des tumeurs cérébrales. Elle a été le premier examen réaliser pour détecter une tumeur cérébrale et est donc très utilisée lors du dépistage. Elle permet une visualisation précise de la région cérébrale atteinte, du tissu cérébral ainsi que la taille de la lésion. La segmentation d'image isole la zone infectée ou lésée par la tumeur du reste de l'image en déterminant sa taille et son emplacement. Ainsi, la planification du traitement est facilitée.

L'interprétation visuelle des IRM cérébrales n'est pas toujours sûre. Pour cela, le besoin d'une interprétation automatique qui permet d'assister les médecins dans leur prise de décision s'est fait ressentir.

## Problématique

La croissance fulgurante de la population lors des dernières décennies, cause une surpopulation des hôpitaux ainsi que des centres médicaux spécialisés. Il est difficile de recevoir un diagnostic précis et assez tôt pour pouvoir déceler précocement les cancers. En effet, le traitement des patients dépend de la précision et de la rapidité du traitement des images médicales. Cependant le problème récurrent est qu'un diagnostic précis peut prendre un certain temps car c'est un processus généralement long et fastidieux où le médecin doit observer et étudier plusieurs images et les comparer afin d'arriver à une conclusion et en poser un diagnostic.

Pour remédier à ce problème de temps et de précision, plusieurs solutions ont été proposées dont l'application des techniques de deep learning et de vision par ordinateur au milieu médicale (l'e-Health) dont la plus utilisée est l'utilisation des modèles de réseaux de neurones convolutifs (CNN).

## Objectif

Dans notre travail nous allons étudier différents modèles existants, et les comparer afin de définir le meilleur modèle à appliquer dans le diagnostic d'une tumeur cérébrale.

Notre mémoire est structuré comme suit :

**Chapitre 1 : Tumeurs et cancers du cerveau**

Dans ce chapitre, nous avons exposé ce qu'est une tumeur, les différents types ainsi que ce qu'est une tumeur du cerveau. Nous avons parlé des symptômes, des causes et du diagnostic. Enfin nous avons conclu le chapitre par une conclusion.

**Chapitre 2 : L'imagerie par résonance magnétique**

Dans ce chapitre, nous avons exposé l'imagerie médicale et son évolution dans le temps puis nous avons parlé de l'imagerie par résonance magnétique (IRM), sa composition, ses avantages et inconvénients et enfin l'importance de l'IRM cérébrale pour le diagnostic d'une tumeur. Ce chapitre se termine par une conclusion

**Chapitre 3 : Intelligence Artificielle, Machine Learning et Deep Learning**

Dans ce chapitre, nous avons abordé ce qu'est l'intelligence artificielle avant de définir ce que sont l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning). Ensuite nous avons exposé ce que sont les réseaux de neurones ainsi que les différentes architectures et les différents secteurs d'activités du deep learning. Enfin nous avons parlé du transfer learning et de l'overfitting avant de terminer ce chapitre par une conclusion

**Chapitre 4 : Etat de l'art**

Dans ce chapitre nous avons mis en avant ce qui a été fait dans la littérature lorsqu'il s'agit d'appliquer le DL à des IRM cérébrales pour diagnostiquer un cancer du cerveau.

**Chapitre 5 : Implémentation et Résultat** Dans ce chapitre nous parlons de l'implémentation et présentons les résultats que nous avons trouvé dans notre étude.

**Conclusion** Nous concluons notre mémoire en récapitulant la structure de notre

# Tumeurs et Cancer du Cerveau

## 1.1 Introduction

Les tumeurs cérébrales, qu'elles soient cancéreuses ou non, peuvent se développer à partir du tissu nerveux dans le cerveau (primaire), ou se propager vers le cerveau depuis n'importe où dans le corps (métastase).

Les tumeurs cérébrales sont des cancers rares, selon les estimations, environ 4570 nouveaux cas ont été recensés en France en 2010, soit 1.3% de nouveaux cas de cancers. La même année, ils représentaient 2% de l'ensemble des décès par cancer, soit 2940 personnes. Bien qu'elles soient plus fréquentes chez les personnes âgées, les tumeurs cérébrales touchent les adultes à tout âge ainsi que les enfants. Chez ces derniers, c'est le cancer le plus fréquent après la leucémie (23% des cancers pédiatriques).[4]

Aux Etats-Unis, on estime que 700 000 personnes vivent avec une tumeur cérébrale primaire. Approximativement 72% de toutes ces tumeurs sont bénignes et 28% sont malignes. [3]

Dans la population âgée, l'incidence des tumeurs cérébrales, primitives et métastatiques, est en augmentation.

Les tumeurs du cerveau peuvent être mortelles et avoir un impact significatif sur la qualité de vie et tout changer pour un patient et ses proches. Elles ne discriminent pas et touchent, hommes, femmes et enfants de toute race et ethnique. Cependant, le taux de survie moyen de tout patient avec une tumeur cérébrale primaire est de 75.7%.[3]

Les tumeurs du cerveau peuvent être traitées par chirurgie, radiothérapie, chimiothérapie ou, le plus souvent, une combinaison de ces traitements. Le taux de survie d'un patient atteint d'une tumeur maligne est de 6.8% avec une médiane de survie de 8 mois, il est donc important d'avoir un diagnostic rapide et précis afin de pouvoir lancer le traitement médical.[3]

## 1.2 C'est quoi une tumeur ?

Une tumeur correspond à une anomalie de la croissance cellulaire qui peut être de nature bénigne ou maligne. Dans la plupart des organes de notre corps, une tumeur bénigne ne cause pas, ou peut, de complications.[8]

Cependant, si une excroissance ou masse dans le cerveau (tumeur cérébrale) grossit, elle peut provoquer des lésions considérables, car les structures qui renferment le cerveau (crâne) ne peuvent pas s'étendre pour faire de la place en cas d'augmentation de leur contenu.

## 1.3 Quels sont les types de tumeurs ?

Il existe pas moins de 20 types de cancers différents. On peut les répartir en 4 grandes catégories :

### 1.3.1 Les cancers "solides"

Les tumeurs solides peuvent se développer dans n'importe quel tissu : peau, muqueuse, os, organes, etc. Ce sont les plus fréquents puisque, à eux seuls, ils représentent 90% des cancers humains. On distingue 2 types de tumeurs :

- Les **carcinomes**, issus de cellules épithéliales (peau, muqueuses, glandes).  
Exemple : cancer du sein, des poumons, de la prostate, de l'intestin, etc.
- les **sarcomes**, moins fréquents, qui sont issus de cellules des tissus conjonctifs (dits tissus de "soutien").  
Exemple : Cancers de l'os, du cartilage, etc...

### 1.3.2 Les cancers "liquides" ou sanguins

On distingue 2 types de cancers dits liquides :

- les **leucémies** sont des cancers du sang et de la moelle osseuse. Elles sont liées à la multiplication anarchique de cellules précurseurs (cellules immatures qui donneront naissance à des cellules "adultes") des globules blancs dans la moelle osseuse. Ces cellules vont ensuite envahir le sang.
- Les **lymphomes** sont des cancers du système lymphatique : ganglions, mais aussi, rate, foie, etc... Ils affectent les lymphocytes, un type de globule blanc. Il existe deux grands types de lymphome, hodgkinien et non hodgkinien, dont les traitements et le pronostic sont différents.

### 1.3.3 Les cancers "métastatiques" ou disséminés

Un cancer métastatique ou disséminé (métastases) signifie que les cellules cancéreuses ont migré à travers le corps, par la voie des vaisseaux sanguins, ou lymphatiques et ont colonisé un ou plusieurs autre(s) tissu(s), loin de la tumeur d'origine. Différents organes peuvent être colonisés par ces métastases : os, foie, poumons, etc. Mais la nature d'un cancer reste toujours déterminée par son point de départ.

### 1.3.4 Les cancers secondaires

Il arrive que certains traitements anti-cancer, indispensables pour guérir la maladie, puissent eux-mêmes provoquer l'apparition ultérieure de nouvelles cellules cancéreuses. Ce sont les cancers secondaires, consécutifs à un traitement.

Le type de cancer, sa localisation, son degré d'agressivité, ses manifestations cliniques et l'état de santé du patient déterminent la façon dont le cancer est traité. [9]

## 1.4 C'est quoi une tumeur du cerveau ?

On parle indifféremment de tumeurs cérébrales, de tumeurs du cerveau, de tumeurs intracrâniennes ou du système nerveux central (SNC) pour désigner les tumeurs qui naissent et se développent à l'intérieur du crâne. [4]

Elles peuvent toucher toutes les parties du cerveau et leurs symptômes varient selon la région du cerveau où se développe la tumeur. Il existe une grande variété de tumeurs du cerveau selon le type de cellule concernée, la localisation de la tumeur, sa taille, la rapidité de son évolution, etc.

Il existe différents types de tumeurs cérébrales, les plus fréquents sont les gliomes, les méningiomes, et les médulloblastomes.

Les méningiomes sont les tumeurs primaires non malignes les plus courantes et les glioblastomes sont les tumeurs primaires malignes les plus communes.

Les tumeurs du cerveau ne doivent pas être confondues avec les métastases cérébrales provenant d'un cancer situé dans un autre organe. Ce ne sont pas non plus des lymphomes cérébraux, qu'on classifie plutôt comme hémopathie maligne que comme tumeur cérébrale.



### 1.4.1 C'est quoi le cancer du cerveau ?

Quand une tumeur maligne grossit et les cellules cancéreuses se propagent aux tissus voisins on parle alors de tumeur cancéreuse ou tout simplement de cancer.[1]

Contrairement aux autres cancers, les tumeurs cancéreuses du cerveau n'entraînent pas de métastases à l'extérieur du cerveau. Elles ne s'étendent pas à d'autres organes.

## 1.5 Quels sont les types de tumeurs du cerveau ?

Les principales fonctions du cerveau sont assurées par un réseau dense de milliards de cellules nerveuses connectées entre elles, les neurones. Ces neurones sont entourés et nourris par des cellules spécialisées, les cellules gliales, qui représentent 90 % de la masse du cerveau.

Il existe plusieurs types de cellules gliales (astrocytes, microglie, oligodendrocytes, etc.) qui possèdent diverses fonctions. Dans un peu plus de 50 % des cas, les tumeurs du cerveau sont des tumeurs des cellules gliales (des « gliomes »).

Le cerveau est entouré et protégé par des enveloppes, les méninges. Il arrive qu'une tumeur se développe à partir de ces enveloppes. On parle alors de « méningiome ».[10]

Les tumeurs du cerveau portent généralement le nom des cellules à partir desquelles elles se développent[4]

- Gliomes

Un gliome est une tumeur des cellules gliales qui peut survenir dans n'importe quelle région du cerveau. Les gliomes représentent la majorité des tumeurs cérébrales. Parmi les gliomes, les astrocytomes affectent les cellules gliales appelées astrocytes. Les astrocytes cancéreux à croissance rapide sont appelés "gliomes" et représentent 20 % des tumeurs cérébrales. Le gliome polygonal est une autre forme de gliome qui peut se développer lente-

ment (appelé "bas grade") ou se développer rapidement ("haut grade"). Les gliomes peuvent être bénins ou cancéreux.

- Neurinomes ou neurofibromes

Les neurofibromes sont des tumeurs, le plus souvent bénignes, qui se développent sur un nerf, en particulier le nerf acoustique.

- Méningiomes

Les méningiomes sont des tumeurs qui se développent à partir de cellules des méninges, les enveloppes qui entourent le cerveau.

- Adénome hypophysaire

L'adénome hypophysaire est une tumeur de l'hypophyse, une partie du cerveau qui contrôle de nombreuses hormones.[10]

Selon leur type (les médecins parlent de type histologique), les tumeurs ne se comportent pas de la même manière. Certaines se développent plus vite que d'autres ou ont un risque accru de récurrence par exemple.

## 1.6 Degré d'agressivité d'une tumeur

Les tumeurs du cerveau sont classées en différents grades selon leur agressivité. Plus la tumeur se développe, plus elle est jugée agressive. On parle de bas grade pour les tumeurs les moins agressives et de haut grade pour les tumeurs qui le sont davantage. Le grade peut aussi être exprimé par un chiffre romain allant de I à IV.

Le grade I correspond aux tumeurs non cancéreuses, ce sont les moins agressives. Le grade IV correspond aux tumeurs les plus agressives.

Connaître le grade d'une tumeur est déterminant dans le choix des traitements et dans le pronostic. Plus le grade est bas, plus le pronostic est favorable .

Quand une tumeur est découverte, plusieurs examens sont réalisés car il est indispensable de déterminer sa localisation, son type et son degré d'agressivité. Ce n'est qu'après ces examens que l'on sait de quel type de tumeur il s'agit et que le traitement approprié est défini.[48]

## 1.7 Quels symptômes ?

Les signes et symptômes apparaissent quand la tumeur est suffisamment grosse pour perturber le fonctionnement du cerveau. Ces symptômes varient selon la localisation de la tumeur.

Certains de ces symptômes incluent :

- maux de têtes persistant dus à l'augmentation de la pression du liquide qui baigne le cerveau.
- crises d'épilepsie liées à des modifications de sensibilité des cellules nerveuses du cerveau.

- nausées et vomissements.
- changement de la personnalité, de la pensée, de l'humeur et du comportement.
- difficulté à parler ou à comprendre les mots.
- mouvements anormaux.
- difficulté à marcher
- faiblesse d'un côté du corps
- troubles de la motricité fine.
- difficulté à avaler et à manger
- troubles de la vue, dont une vision légèrement floue, une vision double et une perte de vision.
- troubles de l'audition
- troubles de l'équilibre
- somnolence
- fatigue
- engourdissement d'une partie du corps
- confusion[2]

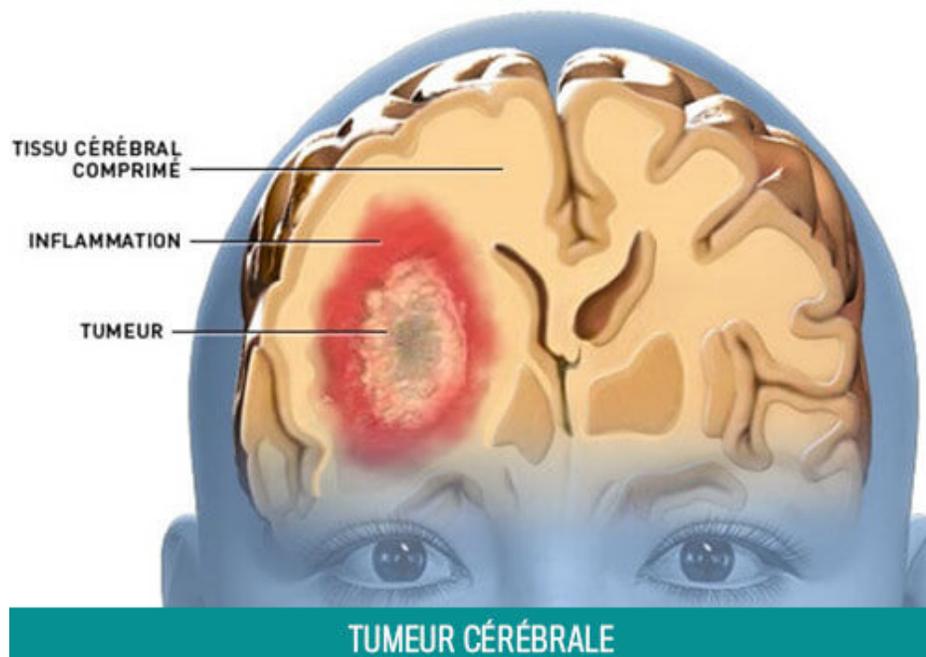


FIGURE 1.1 – Symptôme d'une tumeur cerebrale

## 1.8 Quelles causes ?

Les facteurs qui conditionnent l'apparition d'une tumeur du cerveau sont encore mal connus. Certains facteurs de risque sont suspectés mais leur influence reste peu claire.[7]

### 1.8.1 L'environnement

— **Les substances chimiques :**

Plusieurs substances chimiques sont reconnues comme cancérogènes, c'est-à-dire pouvant favoriser la survenue d'une tumeur cancéreuse du cerveau. Il s'agit de substances chimiques utilisées dans différentes activités professionnelles, telles que la fabrication de caoutchouc synthétique et de certains médicaments, la production de polyvinyle, de pétrole brut et de produits dérivés du pétrole, l'industrie nucléaire et l'armement, et enfin la manipulation de pesticides

— **Les substances radioactives :**

Une exposition prolongée et courante aux substances radioactives peut causer différents types de cancers dont ceux du cerveau.

— **Les champs électromagnétiques :**

Le risque lié à l'exposition régulière à des champs électromagnétiques est encore en cours d'évaluation. Les spécialistes n'ont pas clairement déterminé si cette exposition peut entraîner des tumeurs du cerveau. Quant à l'utilisation de téléphones portables, un principe de précaution s'impose. Les études ne sont pas définitivement tranchées, mais elles vont dans le sens d'un risque possible en particulier chez l'enfant et l'adolescent. Par précaution, il est donc conseillé de limiter l'utilisation du téléphone portable et de ne pas le garder sur soi. Pendant les communications, l'utilisation d'oreillettes est recommandée.[48]

### 1.8.2 L'hérédité

La grande majorité des tumeurs du cerveau n'ont aucun lien héréditaire. Ce n'est pas parce qu'une personne a une tumeur du cerveau que les autres membres de sa famille ont un risque accru de développer le même type de tumeur. Un risque héréditaire existe néanmoins pour deux types de tumeurs du cerveau : le neurinome et le neurofibrome. Ces tumeurs représentent moins de 5% de l'ensemble des tumeurs du cerveau. Ce sont des tumeurs bénignes, qui se développent sur le nerf auditif, reliant le cerveau et l'oreille. Ces tumeurs sont souvent liées à une maladie génétique appelée neurofibromatose. La neurofibromatose n'entraîne pas systématiquement une tumeur du cerveau.[48]

### 1.8.3 Les autres maladies

Les personnes immunodépressives par exemple, celles qui souffrent de VIH/SIDA, ont un risque plus élevé de développer un lymphome du cerveau (une tumeur des cellules chargées de défendre le cerveau contre les infections)[48]

## 1.9 Prévention

Les mesures de prévention sont peu spécifiques et concernent tous les cancers.

- ne pas fumer
- adopter une alimentation équilibrée
- limiter le plus possible sa consommation de boissons alcoolisées
- pratiquer régulièrement une activité physique.[7]

## 1.10 Diagnostic

Généralement c'est l'apparition de symptômes qui permet de suspecter la présence d'une tumeur du cerveau.

Le médecin traitant joue un rôle important dans le diagnostic, car c'est souvent lui qui est consulté en premier. En cas de suspicion de tumeur, l'intervention de plusieurs médecins spécialistes est ensuite indispensable : neurologue, neurochirurgien, neuro-oncologue, radiologue, anatomopathologiste (appelé aussi pathologiste)...

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

- Un examen clinique et neurologique complet
- Des examens d'imagerie
- Un examen anatomopathologique, qui consiste à analyser un échantillon de tumeur.[48]

### 1.10.1 L'examen clinique

L'examen clinique et neurologique est réalisé au cours d'une consultation médicale, soit par le médecin traitant soit par un spécialiste du cerveau.

Il consiste à évaluer l'état de santé général d'une part et le bon fonctionnement du système nerveux d'autre part.

Il commence par examiner l'ensemble du corps (tension artérielle, battements du coeur, etc.), puis évalue le fonctionnement du système nerveux grâce à différents tests.

- des exercices musculaires (marcher, serrer la main, sourire pour vérifier le bon fonctionnement des muscles du visage...).
- un test des réflexes.
- un test de sensibilité (sensation au toucher, réaction au chaud et au froid ou à une légère piqûre...).
- un test d'équilibre et de coordination des mouvements.
- un examen de la vue, des pupilles, des mouvements des yeux et du fond de l'oeil.
- un test d'audition.

- un test de langage (capacité à trouver le bon mot par exemple), de lecture, d'écriture ou de dessin.
- des exercices mentaux (calculs simples), tests de mémoire, de compréhension, etc.

L'examen clinique et neurologique permet de constater d'éventuelles anomalies. Pour déterminer si ces anomalies sont dues à une tumeur ou non, des examens d'imagerie sont indispensables.[48]

### 1.10.2 Les examens d'imagerie

Les examens d'imagerie consistent à réaliser des images précises du cerveau. Pour le diagnostic d'une tumeur du cerveau, on utilise un scanner, une IRM ou préfèrent une IRM précédée d'un scanner.[48]

### 1.10.3 L'examen anatomopathologique (biopsie)

La biopsie est une intervention chirurgicale consistant à prélever un très petit fragment d'une tumeur, une lésion, un kyste, afin de l'analyser en anatomopathologie.[48] Elle permet de confirmer le diagnostic avec certitude.

La biopsie par stéréotaxie est une technique permettant de réaliser un prélèvement avec une précision millimétrique. Elle consiste à utiliser des repères dans l'espace pour cibler une zone.

La biopsie après craniotomie ou "à ciel ouvert" est envisagée lorsque la tumeur est superficielle et facilement accessible et lorsqu'elle est située dans un zone du cerveau non fonctionnelle. L'échantillon prélevé est ensuite analysé par microscopie.[5]



FIGURE 1.2 – Examen de biopsie

## 1.11 Traitement d'une tumeur du cerveau

Il existe différentes possibilités de traitement, qui peuvent être proposés seuls ou associés les uns aux autres : chirurgie, radiothérapie, chimiothérapie. Selon les cas, ils visent à éliminer la totalité de la tumeur, à réduire le risque de récurrence, à contenir le développement de la tumeur, à soulager et prévenir les symptômes provoqués par la tumeur.

Le choix et l'ordre des traitements dépendent de nombreux facteurs :

- La gravité apparente de la tumeur
- Le type de tumeur
- Sa localisation et son étude
- L'âge et l'état de santé du patient[48]

## 1.12 Conclusion

Dans ce chapitre nous avons défini ce que sont les tumeurs cérébrales, leurs types et leurs symptômes. Nous avons aussi mentionné les causes et les différentes manières de diagnostiquer une tumeur dont le scanner et l'imagerie par résonance magnétique (IRM) qui permettent de définir l'étendue et la taille de la tumeur ainsi que son grade. Nous avons aussi cité les différents traitements.

# L'imagerie par résonance magnétique

## 2.1 Introduction

Les tumeurs du cerveau ne sont pas des maladies faciles à repérer car ayant des symptômes semblables à d'autres maux. Le patient passe donc plusieurs examens dont l'objectif est d'établir le diagnostic de cancer, déterminer son étendu et son taux d'agressivité.

Le diagnostic d'une tumeur du cerveau est un processus qui passe par plusieurs étapes; tout d'abord un examen clinique et neurologique complet réalisé par le médecin traitant ou un médecin spécialiste, suivi d'un examen anatomopathologique composé d'une biopsie. Enfin la partie indispensable est un scanner qui sera suivi d'une IRM du cerveau du patient qui permettra au médecin d'avoir un diagnostic plus précis.

Une IRM est un appareil qui conduit des champs magnétiques et des ondes radio pour créer des images détaillées des organes et des tissus. Le traitement des images IRM est extrêmement complexe et fait l'objet d'un examen constant par les chercheurs afin d'offrir aux pathologistes une meilleure expérience de diagnostic des patients. Pour ces raisons, l'auto-segmentation des tumeurs cérébrales devient importante.

## 2.2 Evolution de l'imagerie médical

L'imagerie médicale fait son apparition avec la découverte des rayons X en 1895 par Wilhelm Röntgen (physicien allemand). Les scientifiques peuvent alors observer et étudier l'intérieur du corps humain sur des sujets vivants. L'imagerie médicale se développe ensuite au cours de la première guerre mondiale, puis la découverte de la radioactivité fonde les bases des ultrasons. [28]

L'IM continue à se développer, et dans les années 70, l'imagerie 3D voit le jour avec l'invention du scanner, de la tomographie par émission de positons (TEP) et de l'imagerie par résonance magnétique (IRM)

Aujourd'hui l'imagerie médicale continue de se développer et de nouvelles technologies sont proposées telles que les images en 4D et le Real Time Imaging.[45]

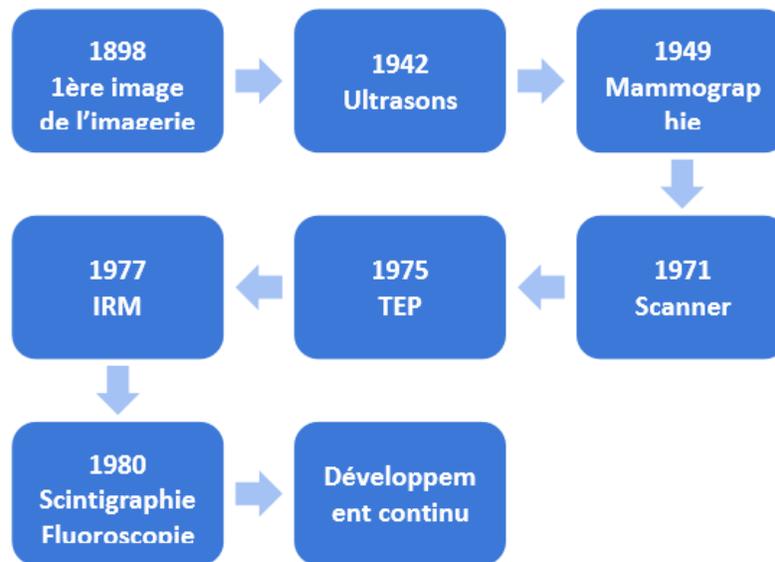


FIGURE 2.1 – Évolution de l'imagerie Médicale (source :[45])

### 2.3 L'exploration radiologique

Les examens radiologiques fournissent une image de l'ensemble du corps ou d'une partie de celui-ci. L'imagerie aide les médecins à diagnostiquer un trouble, à déterminer sa gravité. L'imagerie a une grande part dans la réévaluation et dans la surveillance de l'évolution des patients. La plupart des examens d'imagerie sont indolores, relativement sûrs et non invasifs (c'est-à-dire qu'ils ne nécessitent pas d'incision de la peau ni l'insertion d'un instrument dans le corps).[32]

On recense plusieurs types d'examen d'imagerie médicale :

- **La radiographie** Basée sur l'utilisation des rayons X, la radiographie permet principalement d'obtenir des clichés en deux dimensions des os et articulations, mais aussi des poumons et du sein (mammographie).

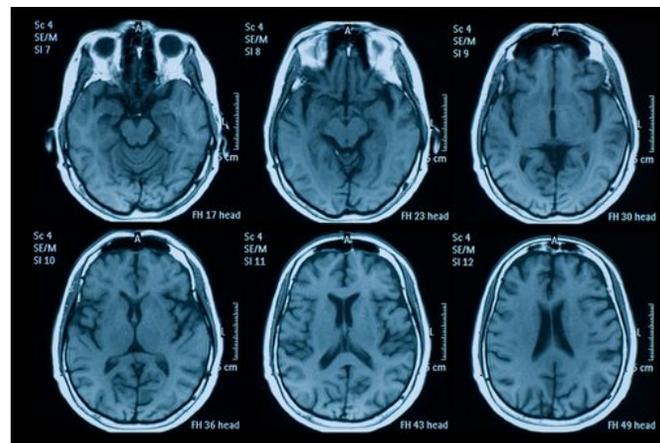


FIGURE 2.2 – Radio



## 2.4 Imagerie Par Résonance Magnétique

### 2.4.1 Qu'est ce que c'est ?

L'imagerie par Résonance Magnétique (IRM) est l'examen fondamental qui permet d'étudier la structure des tissus. L'appareil fonctionne au moyen d'un aimant et d'ondes radio. On doit enlever tous les objets métalliques et signaler les implants métalliques. Le radiologue utilisera les séquences pondérées en T2, FLAIR et après l'injection d'un produit de contraste, le gadolinium.[30]

### 2.4.2 Objectifs de l'IRM

- De localiser la tumeur (sus-tentorial, sous-tentorial, intra- ou extra- parenchymateux, uni ou multifocal)
- De préciser ses caractéristiques : lésion homogène ou hétérogène, prise de contraste ou non, effet de masse sur les structures de voisinage, existence d'un oedème péri-lésionnel, présence de calcifications, etc.
- D'apporter des arguments pour le diagnostic différentiel
- De diagnostiquer certaines complications comme une hémorragie intra-tumorale, une hydrocéphalie avec dilatation des ventricules cérébraux, un engagement...[30]

### 2.4.3 En pratique

La durée de l'examen est variable et en fonction du nombre de clichés. En règle générale 30 à 40 minutes sont nécessaires.

Un produit de contraste est utile pour voir précisément l'aspect de certaines tumeurs. L'examen est réalisé avec injection de gadolinium, 10 à 20 ml selon le poids de la personne. Le gadolinium n'est pas un produit à base d'iode. L'injection en intraveineuse n'est pas très douloureuse.

Les contre-indications de l'IRM relèvent du champ magnétique intense appliqué. L'IRM est contre-indiquée chez certaines personnes porteuses de stimulateurs cardiaques, de prothèses valvulaires cardiaques, les implants cochléaires ou de clips vasculaires intervertébraux et de prothèses articulaires. Il n'est pas recommandé chez les patients porteurs de pompes à insuline, de drain ventriculaire (valve), de prothèses auditives, de stérilet, et de lentille intra-oculaire (chirurgie de la cataracte).[30]

## 2.5 Composition de l'appareil d'IRM

L'appareil d'IRM est constitué d'un tunnel d'environ 1,60 mètre de long entouré d'un aimant très puissant. Pendant l'examen, vous êtes allongé sur un lit (ou une table d'examen) qui se déplace à l'intérieur de ce tunnel pour positionner la zone de votre corps à examiner au centre de l'aimant.

Cet aimant fait « vibrer » les atomes d'hydrogène, éléments qui composent les molécules d'eau de notre corps. Or notre organisme est constitué de plus de 65 % d'eau. Au centre de chaque atome d'hydrogène se trouve une particule encore plus petite appelée proton : c'est cette particule qui est à l'origine du phénomène.

Une antenne placée autour de la région à examiner va capter « les vibrations » des protons. Les protons « vibrent » à des vitesses variables selon le type de tissu dans lequel ils se trouvent. Cela permet de bien distinguer les différentes structures de l'organisme sur les images obtenues, dont les tumeurs par exemple.

L'injection d'un produit dit « de contraste » dans la circulation sanguine est souvent nécessaire pendant l'examen pour améliorer la visibilité de certains organes et la détection des anomalies.

Les signaux reçus des millions de protons dans le corps sont réunis pour créer des images en coupes détaillées de l'intérieur du corps. Des images en 2 ou 3 dimensions sont ensuite reconstituées par un ordinateur. Elles sont interprétées par un radiologue.[29]

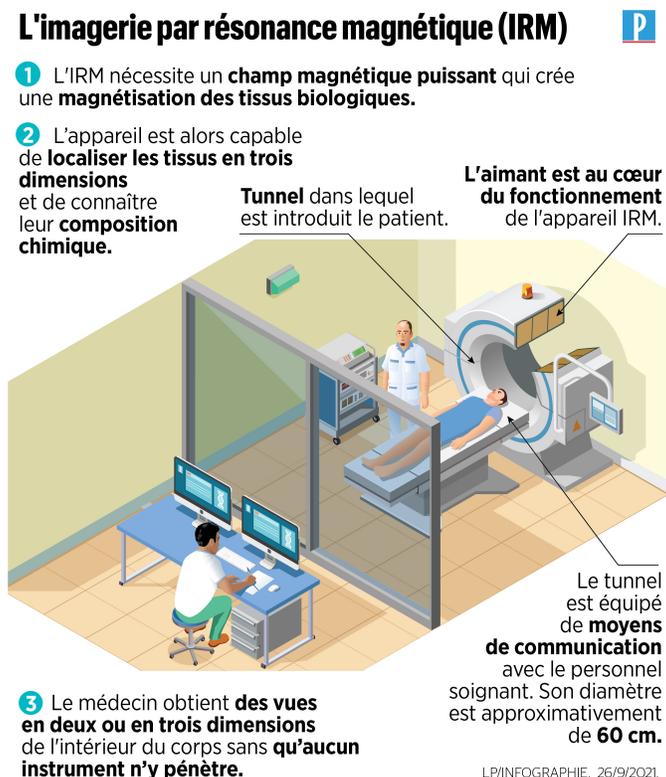


FIGURE 2.5 – Appareil IRM

## 2.6 Avantages de l'IRM

Contrairement au scanner, une IRM a l'avantage de ne pas exposer le patient aux rayons X.

L'IRM est extrêmement précise et a l'avantage d'apporter une bonne visualisation, avec une bonne résolution et un bon contraste, de la graisse, de l'eau, et aussi de l'oedème et de l'inflammation péri lésionnelle. Elle permet ainsi de mieux distinguer des tissus de composition différentes.[24]

## 2.7 Existe-t-il des risques ?

L'IRM est sans risque d'irradiation parce qu'elle n'utilise pas de rayons X. Afin de limiter l'exposition aux ondes électromagnétiques et aux effets indésirables possibles liés à l'emploi d'un produit de contraste, cet examen est prescrit par votre médecin dans le cadre d'indications précises, à certaines phases du diagnostic, du traitement ou du suivi.[29]

## 2.8 Inconvénients de l'IRM

L'IRM nécessite plus de temps que la TDM. Par ailleurs, l'IRM a généralement moins de chances d'être immédiatement disponible que la TDM. Par conséquent, la TDM peut être préférable en cas d'urgence, comme pour des blessures graves ou un accident vasculaire cérébral. L'IRM est aussi plus chère que la TDM.

Les autres inconvénients sont

- La claustrophobie et la difficulté à tenir une personne dans l'espace clos et étroit de l'appareil d'IRM.
- Les effets du champ magnétique sur les dispositifs métalliques implantés dans le corps
- Les réactions à l'agent de contraste[6]

## 2.9 L'importance d'une IRM cérébrale

L'IRM cérébrale permet de détecter des anomalies cérébrales et d'en déterminer la cause (vasculaire, infectieuse, dégénérative, inflammatoire ou tumorale).[31]

L'IRM permet d'identifier les tumeurs qui n'auraient pas été repérées sur les images du scanner. C'est un examen plus sensible et qui tend à remplacer progressivement le scanner en première intention. Les images obtenues ont également l'avantage de mieux décrire la tumeur, sa localisation exact, sa taille et son extension éventuelle à d'autres structures du système nerveux (méninges, moelle épinières...). Elle permet de recueillir des données qui faciliteront la décision thérapeutique ultérieure (chirurgie et/ou radiothérapie)[5]

## 2.10 Conclusion

Dans ce chapitre nous avons introduit les différents types d'imagerie médicale puis avons présenté l'imagerie par résonance magnétique, décrit ce que c'est, son déroulement, ses objectifs, ce qui la compose, ses avantages et inconvénients.

Enfin nous avons expliqué son importance lors du diagnostic.

# Intelligence Artificielle, Machine Learning et Deep Learning

## 3.1 Introduction

L'intelligence artificielle a été créée dans les années 1950 par un pionnier du domaine, John McCarty. Aujourd'hui l'IA est de plus en plus présente dans notre quotidien (des outils de recommandation comme Alexa ou Google, GPS, Email, etc.), mais aussi dans le domaine de la santé, plus spécifiquement dans les services d'imagerie médicale. Plusieurs enjeux sont à prendre en compte lors de l'utilisation de l'IA dans l'IM. Tout d'abord d'un point de vue médical, afin de pouvoir améliorer le diagnostic et les traitements associés, ainsi que leur précision. L'IA peut être un gain de temps pour le personnel soignant en déléguant certaines tâches à l'IA.

Pour cela nous utilisons des techniques de Deep learning qui permettent d'obtenir des résultats plus pointus lors du traitement des images. L'une de ces techniques les plus recommandées, le CNN, consiste à utiliser un réseau de neurones artificiels afin d'extraire les caractéristiques de l'image et d'arriver à une conclusion.

Dans ce chapitre nous allons définir ce qu'est l'IA et le machine learning, le deep learning ainsi que les différentes architectures existant dans le deep learning.

## 3.2 Intelligence Artificielle

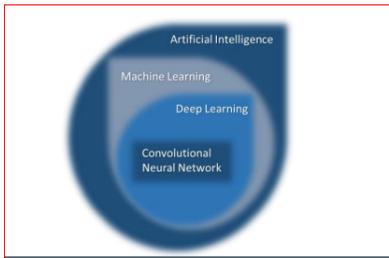


FIGURE 3.1 – Evolution de l'IA

### 3.2.1 Définition

L'intelligence artificielle représente tout outil utilisé par une machine afin de reproduire des comportements liés aux humains, tel que le raisonnement, la planification et la créativité.[25]

L'intelligence Artificielle (IA) est née avec l'objectif de faire produire des tâches humaines par des machines mimant l'activité du cerveau. Face aux déboirs des premières heures, deux courants se sont constitués.

Les tenants de l'intelligence artificielle dite *forte* visent à **concevoir une machine capable de raisonner comme l'humain**, avec le risque supposé de générer une machine supérieure à l'homme et dotée d'une conscience propre. Cette voie de recherche est encore explorée aujourd'hui, même si de nombreux chercheurs en IA estiment qu'atteindre un tel objectif est impossible.

D'un autre côté les tenants de l'IA dite *faible* mettent en oeuvre toutes les technologies disponibles pour **concevoir une machine capable d'aider les humains dans leurs tâches**. Ce champ de recherche mobilise de nombreuses disciplines, de l'informatique aux sciences cognitives en passant par les mathématiques, sans oublier les connaissances spécialisées des domaines auxquelles on souhaite l'appliquer tel que la médecine.[26]

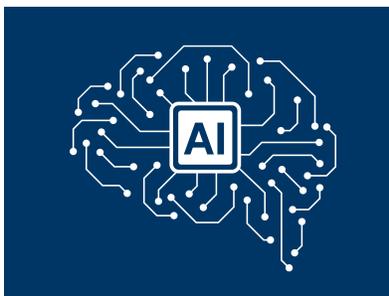


FIGURE 3.2 – Intelligence artificielle

### 3.2.2 L'intelligence artificielle dans le milieu médical

L'intelligence artificielle appliquée dans le milieu médical peut se présenter de différentes formes :

- La médecine prédictive qui consiste en la prédiction d'une maladie ou son évolution.
- Médecine de précision qui consiste en la recommandation de traitement personnalisé.
- Aide à la décision (diagnostic et thérapeutique)
- Robots compagnons, notamment pour les personnes âgées ou fragiles.
- Chirurgies assistées par ordinateur.
- Prévention en population générale (Anticipation d'une épidémie, pharmacovigilance).[26]

## 3.3 Machine Learning

Le Machine Learning est l'un des sous-ensembles de l'IA qui est lui-même composé du Deep Learning. Le Machine Learning se base sur l'expérience des machines qui peuvent l'utiliser pour améliorer leurs performances et l'exécution des tâches. L'apprentissage automatique permet l'ajout d'informations supplémentaires dans le modèle - comme le clustering (organisation des données de façon homogène) - et leur utilisation dans le cadre des entraînements. Il y a aussi évidemment une phase de test du modèle, ce dernier peut-être aussi utilisé pour des tâches de prédiction. Le Machine Learning permet à des machines d'effectuer des apprentissages automatiques même si elles n'ont pas été conçues et programmées pour cela au départ.[45]

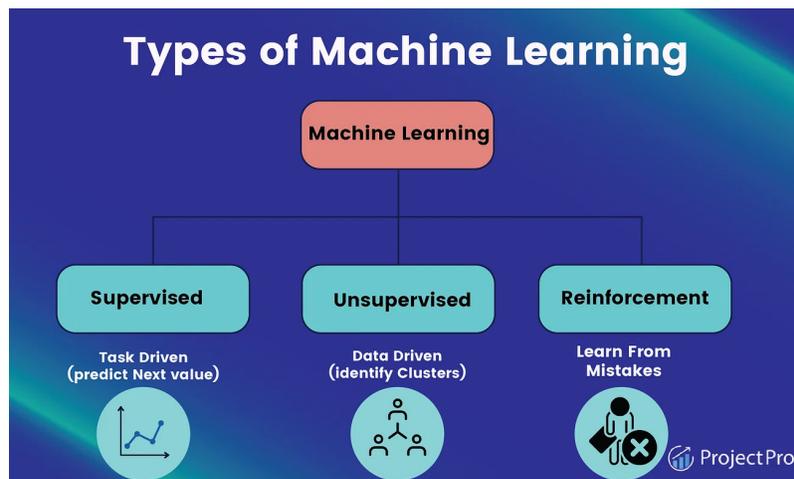


FIGURE 3.3 – Machine learning

### 3.3.1 Types d'apprentissage automatique

On distingue trois techniques de machine learning :

- **Apprentissage supervisé**

L'apprentissage supervisé est fait en utilisant une vérité, c'est-à-dire qu'on a une connaissance préalable de ce que les valeurs de sortie pour nos échantillons devraient être. Par conséquent, le but de ce type d'apprentissage est d'apprendre une fonction qui, compte tenu d'un échantillon de données et de résultats souhaités, se rapproche le mieux de la relation entre les entrées et les sorties observables dans les données.[40]

L'apprentissage supervisé est le type d'apprentissage automatique le plus couramment utilisé. Dans ce type d'apprentissage, on alimente l'algorithme en entrée avec des données déjà étiquetées (features) et on lui indique les résultats attendus (labels)

- *Classification*

Elle consiste à attribuer une classe à des objets. Par exemple, dans le milieu bancaire on peut identifier si une transaction est frauduleuse ou non frauduleuse de manière automatique. On parle de détection d'anomalie. On associe une réponse prédéfinie (oui ou non, jaune, rouge, vert,...) à un objet avant de demander à l'algorithme de réaliser cette classification. Quelques algorithmes de classification : SVM, les forêts aléatoires et les arbres de décision.

- *Régression*

On attribue pas une classe mais une valeur mathématique : un pourcentage ou une valeur absolue. Par exemple, une probabilité pour une machine de tomber en panne (15%, 20%, etc..) Quelques algorithmes de régression : La régression linéaire, la régression polynomiale et la régression logistique.

- **Apprentissage non supervisé**

La différence lorsqu'on parle du type d'apprentissage non supervisé, c'est que **les réponses que l'on cherche à prédire ne sont pas disponibles dans les jeux de données**. Ici, l'algorithme utilise un jeu de données non étiquetées. On demande alors à la machine de créer ses propres réponses. Elle propose ainsi des réponses à partir d'analyses et de groupement de données. Quelques exemples d'algorithmes non supervisés : K-Nearest Neighbors et K-Means et les règles d'association.

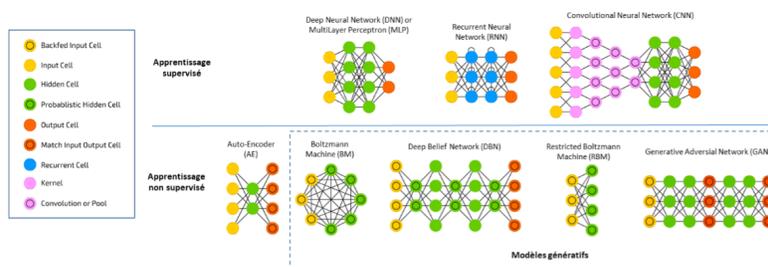


FIGURE 3.4 – apprentissage supervisé et non supervisé

- **Apprentissage par renforcement**

Ici, la machine peut se montrer encore plus innovante, car elle peut élaborer ses propres stratégies et s'adapter dans le temps et dans son environnement pour réaliser des tâches données. Elle élabore des réponses complexes. L'algorithme devient un agent autonome, dont l'objectif est de réaliser une action au sein d'un environnement. S'il y parvient selon les critères établis par le développeur, il est récompensé. À lui d'élaborer la suite d'actions qui lui permettra d'atteindre son objectif.[23]

### 3.4 Deep learning

Le Deep Learning est un type de machine learning qui repose sur l'**apprentissage de représentations de données**, par opposition à des algorithmes spécifiques à une tâche. L'apprentissage peut être supervisé, semi-supervisé ou non supervisé. Les architectures de Deep Learning telles que les *réseaux neuronaux profonds*, les *réseaux de croyance profonds* et les *réseaux neuronaux récurrents* ont été appliquées à des domaines tels que la vision par ordinateur, la reconnaissance vocale, le traitement du langage naturel et la reconnaissance audio.

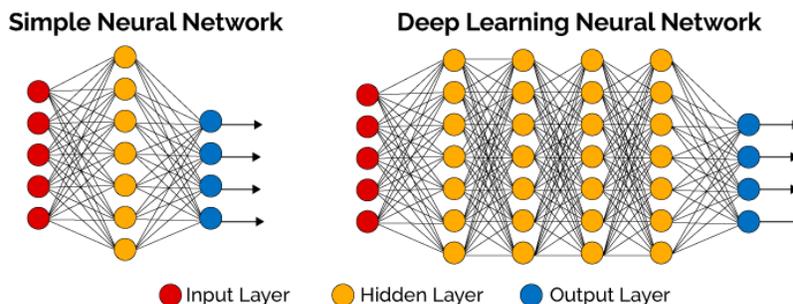


FIGURE 3.5 – Architecture Deep Learning

Dans l'apprentissage supervisé, un programme informatique est exposé à des exemples d'entrée-sortie connus sous le nom de données de formation, après quoi le programme généralise son expérience en créant un modèle interne qui lui permet de générer des sorties raisonnables pour de nouveaux cas. En d'autres termes, le programme informatique apprend à partir d'exemples de

données afin d'être en mesure de produire des résultats précis lorsqu'on lui présente de nouvelles données ; ce processus peut être répété sur de nombreuses itérations (*époques*) jusqu'à ce que les performances atteignent le niveau de précision souhaité. La capacité des modèles de réseaux neuronaux profonds à apprendre des **mappings non linéaires complexes** les rend particulièrement utiles dans des domaines où les données sont moins structurées qu'on ne le pensait auparavant : l'un de ces domaines est le **traitement du langage naturel (NLP)**. [33]

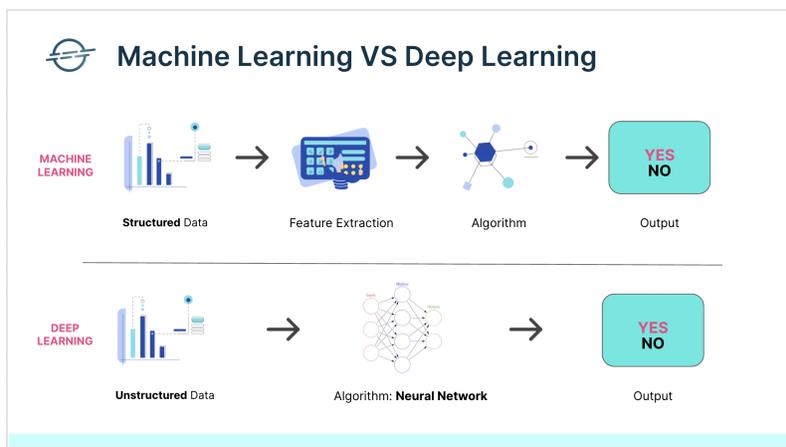


FIGURE 3.6 – Machine Learning VS Deep learning

### 3.5 Les réseaux de neurones

Ce sont des fonctions mathématiques à plusieurs paramètres, ajustables. L'analogie date des premiers automates proposés en 1943 par Warren McCulloch et Walter Pitts. Comme dans les neurones du cerveau où des connexions se créent, disparaissent ou se renforcent en fonction de différents stimuli et produisent une action, les réseaux de neurones artificiels (ou formels) ajustent des paramètres (baptisés poids synaptiques en référence au fonctionnement biologique du cerveau) en fonction de données, d'entrée afin de fournir la meilleure réponse possible.

En machine learning, un neurone fait une combinaison linéaire des entrées qu'il reçoit, à laquelle il ajoute une valeur appelée biais. Une fonction non linéaire, dite d'activation, (comme par exemple tangente hyperbolique) est alors appliquée à la valeur de sortie. Cette valeur est ensuite transmise à la couche de neurone suivante. Chaque neurone effectue ainsi un calcul très rudimentaire, et c'est la succession des couches de neurones qui permet d'obtenir des réseaux complexes. Durant cette phase dite «d'entraînement», le réseau va ajuster automatiquement les paramètres de chaque neurone, c'est-à-dire les valeurs des poids et du biais afin de minimiser l'erreur moyenne calculée sur l'ensemble des exemples entre la sortie attendue et celle observée. L'hypothèse est qu'après cette phase d'entraînement, le réseau sera capable de traiter de manière satisfaisante de nouveaux exemples, dont la sortie est inconnue, en fonction de ce qu'il a «appris».

Cette phase d'apprentissage peut se faire sur des caractères manuscrits, des objets dans une image, des sons...

Dans un réseau de neurones à deux couches, la première couche est constituée d'un ensemble de neurones connectés en parallèle et fournissant un ensemble de sorties, elles-mêmes combinées pour devenir les entrées d'un nouvel ensemble de neurones formant une seconde couche.[34]

## 3.6 Les différentes architectures du Deep Learning

Parmi les différentes architectures de réseaux de neurones nous pouvons citer les suivantes :

- **Réseaux neuronaux profonds (DNN) :**

Un DNN, parfois aussi appelé perceptron multicouche (Multilayer Perceptron - MLP), est composé de plusieurs couches cachées dans lequel tous les neurones d'une couche  $i$  sont connectés à tous les neurones de la couche  $i+1$ . Cette architecture simple a l'inconvénient de présenter un processus d'apprentissage potentiellement lent.

- **Réseaux neuronaux récurrents (RNN) :**

Un réseau de neurones récurrent (Recurrent Neural Network - RNN) est adapté au traitement d'informations ayant une dépendance temporelle. C'est le type de réseau utilisé pour le traitement de la parole ou encore de la vidéo. Contrairement à d'autres types de réseaux profonds, l'idée est ici de garder en mémoire les informations traitées précédemment, pour aider le réseau à prédire les données suivantes. La sortie  $y_t$  du réseau est donc fonction non seulement de l'entrée  $x_t$  à un temps  $t$  mais aussi des entrées  $x_{t-i}$  aux temps  $t-i$ . On compte parmi les RNN les modèles dits de Long Short-Term Memory (LSTM) et de Gated Recurrent Unit (GRU).

- **Long Short Term Memory LSTM :**

Ils sont capables de se souvenir des entrées sur une longue période de temps. C'est parce qu'ils contiennent des informations dans une mémoire, un peu comme la mémoire d'un ordinateur. Le LSTM peut lire, écrire et supprimer des informations de sa mémoire. Dans cette mémoire, on décide à chaque fois de stocker ou de supprimer des informations en fonction de l'importance qu'elle accorde à l'information. L'attribution de l'importance se fait par des poids, qui sont également appris par l'algorithme.

- **Gated Recurrent Units GRU :**

Ce modèle est similaire aux LSTM car il permet aussi de résoudre le problème de mémoire à court terme des modèles RNN. Pour résoudre ce problème, GRU intègre

les deux mécanismes de fonctionnement de porte appelés Update gate et Reset gate : les portails de réinitialisation et de mise à jour sont concrètement des vecteurs qui contrôlent les informations à conserver et à négliger.[36]

— **Les Transformers :**

Les transformers, ou les modèles séquence à séquence, sont des modèles de Deep Learning qui appartiennent à une classe spéciale d'architectures de réseaux de neurones récurrents. Ils sont devenus des modèles de choix pour les Data Scientists lorsqu'ils travaillent sur du texte (Natural Language processing - NLP), remplaçant les anciens RNNs tels que le LSTM.[37]

• **Auto-encodeur (AE) :**

Un auto-encodeur (Auto-Encoder-AE) est un réseau à une couche cachée de dimension inférieure à celle d'entrée. Cette couche cachée sert de couche d'encodage, dans le but d'identifier une structure dominante latente de dimension réduite par rapport au signal d'entrée. Les neurones de la couche d'entrée sont tous connectés à ceux de la couche cachée, qui sont tous connectés à la couche de sortie. Lorsque la couche d'encodage sert d'entrée à une autre AE, on parle d'empilement d'AE (Stacked Auto-Encoder, SAE), ce qui permet de générer plusieurs niveaux d'abstraction. Plusieurs variantes ont été proposées : Stacked Denoising (SDAE), Sparse (SAE) et Variational (VAE). Un des avantages de cette méthode est d'être utilisable dans le cadre d'apprentissages non supervisés, ne nécessitant pas de données labellisées.

• **Machine de Boltzmann (BM) :**

La machine de Boltzmann est une autre architecture d'apprentissage non supervisé. L'objectif est le même que dans un AE (extraire des représentations), mais basé sur un modèle statistique différent. Les connexions entre les neurones sont bidirectionnelles, une BM étant donc assimilable à un modèle dit génératif, c'est-à-dire qui peut générer de nouvelles données d'entrée pendant l'apprentissage. Dans une BM standard, tous les neurones sont connectés entre eux, tandis que seuls les neurones de couches distinctes le sont dans une machine de Boltzmann restreinte (Restricted Boltzmann Machine - RBM).

• **Réseaux de croyances profonds (Deep belief network - DBN) :**

Les Deep Belief Network (DBN) sont essentiellement des SAEs dans lesquelles les couches d'encodage sont remplacées par des RBMs. Seules les deux couches les plus profondes possèdent des connexions bidirectionnelles. L'entraînement est réalisé de façon non supervisée, sauf pour l'ajustement final des paramètres du réseau réalisé de manière supervisée via l'ajout d'une couche de classification en sortie de réseau. La machine de

Boltzmann profonde (Deep Boltzmann Machine - DBM) est l'équivalent du DBN, mais avec des connexions bidirectionnelles entre les neurones de chaque couche.

- **Architecture de generative adversial network (GAN) :**

L'architecture d'un Generative Adversial Networks (GAN) repose sur deux modèles : un modèle génératif G qui produit des données synthétiques et un modèle discriminant D qui estime la probabilité que ces données fassent partie des données d'entraînement. L'objectif du processus est de rendre capable le modèle G de tromper D.[50]

- **Réseaux neuronaux convolutifs (CNN) :**

Les CNN sont basés sur la structure du cortex visuel et utilisent des couches convolutives pour extraire les caractéristiques d'une image d'entrée. Les CNN sont utilisés pour des tâches de reconnaissance d'images et se sont avérés très efficaces pour détecter des objets dans des images.[33]

À partir d'une grande quantité de données d'apprentissage, ces modèles découvrent des structures itératives en ajustant automatiquement leurs paramètres internes à l'aide d'algorithmes de rétropropagation. Chaque couche du réseau transforme le signal de manière non linéaire pour augmenter la sélectivité et l'invariance de la représentation. Avec suffisamment de couches, le réseau peut créer une hiérarchie de représentations qui rendra le modèle à la fois sensible aux très petits détails et insensible aux grandes variations.

### 3.7 Les différents secteurs d'activité des DL

Le Deep Learning a été appliqué avec succès dans un certain nombre de domaines, notamment la *vision par ordinateur*, la *reconnaissance vocale*, le *traitement du langage naturel* et la *traduction automatique*. La technologie fournit également un moyen de représenter les connaissances et de faire des déductions à partir de données. Certains sont plus évidents que d'autres et certains sont encore émergents.

Dans le *domaine de la santé*, le deep learning a été utilisé pour détecter le cancer du sein plus tôt et avec plus de précision. Il est également utilisé pour identifier le cancer de la peau et d'autres formes de cancer sur des images médicales telles que les IRM et les scanners.[33]

	Machine Learning	Deep Learning
Organisation des données	Données structurées	Données non structurées
Base de données	Contrôlable	> 1 million de données
Entraînement	Entraînement par l'humain nécessaire	Système d'apprentissage autonome
Algorithme	Algorithme modifiable	Réseau neuronal d'algorithmes
Champ d'application	Actions simples de routine	Tâches complexes

FIGURE 3.7 – Tableau comparatif de ML et DL

## 3.8 Les réseaux de neurones convolutifs- Convolutional Neural Network CNN

### 3.8.1 Définition

L'un des algorithmes les plus performants du Deep Learning pour le traitement d'image. Les réseaux de neurones convolutifs (CNN) sont des réseaux de neurones multicouches spécialisés dans la classification des images et la reconnaissance d'objets ou de motifs (patterns) caractéristiques.

Un réseau neuronal convolutif profond (CNN) utilise plusieurs couches de traitement entre l'entrée et la sortie, créant une hiérarchie de caractéristiques à chaque couche. Les CNN sont utilisés en vision par ordinateur pour reconnaître des objets dans des images ou des vidéos en temps réel, mais ils sont également utiles pour d'autres tâches comme la reconnaissance vocale ou la traduction entre langues.

Le nombre de couches d'un CNN détermine sa complexité : plus il est complexe, meilleures sont les performances qu'il peut offrir. Mais il y a une limite supérieure pour chaque couche individuelle ; l'ajout d'un trop grand nombre de couches rendra les CNN moins efficaces pour collecter des informations à partir de leurs entrées et les traiter en sorties assez rapidement pour suivre les applications du monde réel telles que les véhicules autonomes qui nécessitent des temps de réponse de quelques millisecondes.[33]

Utilisé pour la première fois en 2012, lors de la compétition annuelle de VO ILSVRC, ce nouvel algorithme de deep learning est un réseau de neurones dont la méthodologie ressemble à celles des méthodes d'apprentissages traditionnelles : Recevoir des images, détecter les features (caractéristiques) et les classer. CNN est plus avancé que les autres méthodes car il extrait lui-même et décrit les features. Son architecture, alternant couches de pooling et couches de convolution ainsi qu'une couche d'entrée et une couche de sortie lui permettent d'extraire un large éventail de features allant des plus simples aux plus compliquées.

### 3.8.2 But

Le but d'un CNN est de trouver à chaque étape des motifs caractéristiques de l'image d'entrée, d'abord simples, puis de plus en plus élaborés au fur et à mesure que les couches se succèdent, les localiser et les classifier.

### 3.8.3 Fonctionnement d'un CNN

Les CNN comportent deux parties distinctes : la partie convolutionnelle et la partie entièrement connectée (fully-connected)

La première repose sur des couches de convolution alternant avec des couches pooling (agrégation) et des couches d'activation (ReLU).

L'image en entrée est fournie sous forme de matrice de pixels. Elle passe par une succession de filtres au niveau de chaque couche de convolution, créant de nouvelles images appelées cartes (maps) de convolution. La couche pooling réduit la taille de l'image par une fonction qui prend le max ou la moyenne de plusieurs pixels voisins. Les sorties de la partie convolutionnelle qui sont des matrices sont transformées en un vecteur unique appelé caractéristiques du CNN.

Ce vecteur en sortie de la partie convolution est ensuite branché en entrée de la deuxième partie qui est constitué d'une ou plusieurs couches fully-connected. Le rôle de cette partie est de combiner les caractéristiques du CNN pour classer l'image. La dernière couche de la partie fully-connected contient un nombre de neurones égale au nombre de catégories de classement (un neurone pour chaque catégorie (classe)). Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

Il est à noter que la première couche de convolution apprend des caractéristiques simples qui représentent des éléments de structure rudimentaires de l'image (contours, coins). Plus les couches de convolution sont "hautes" c'est-à-dire loin de l'entrée du réseau, plus les caractéristiques apprises sont complexes (les couches de convolution les plus hautes (profondes) apprennent donc des caractéristiques sophistiquées).

On peut aussi citer la couche de correction (ReLU), celle entièrement connectée, la couche de déconvolution, la couche de perte, la couche de normalisation et la couche de régularisation.

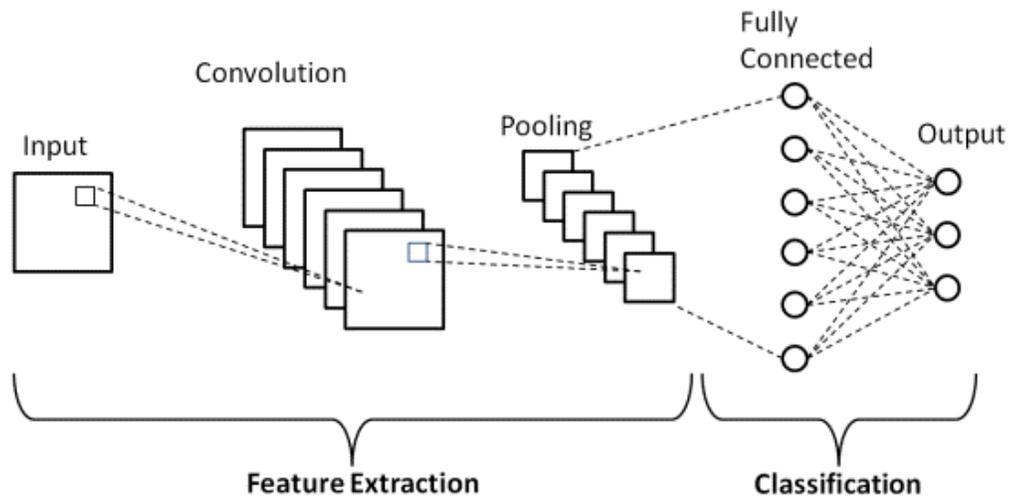


FIGURE 3.8 – Fonctionnement d'un CNN

### 3.8.4 Architecture d'un CNN

#### 3.8.4.1 Couche de convolution

La couche de convolution est la composante clé des CNN, et constitue toujours au moins leur première couche.

Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée. Pour cela, on réalise un filtrage par convolution : le principe est de faire "glisser" une fenêtre représentant la feature sur l'image, et de calculer le produit de convolution entre la feature et chaque portion de l'image balayée. Une feature est alors vue comme un filtre : les deux termes sont équivalents dans ce contexte.

La couche de convolution reçoit donc en entrée plusieurs images, et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux features que l'on souhaite retrouver dans les images.

On obtient pour chaque paire (image, filtre) une carte d'activation, ou feature map, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature.

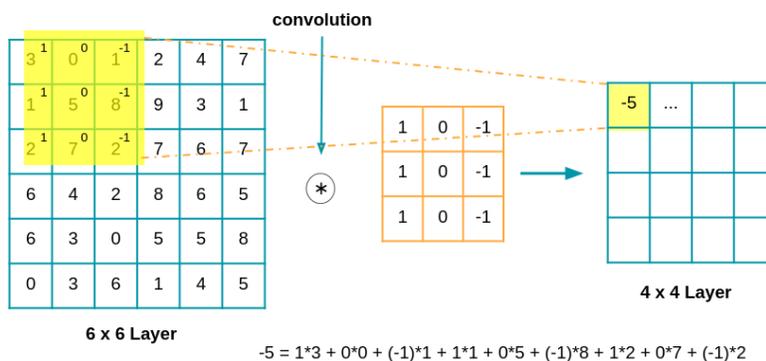


FIGURE 3.9 – Couche de convolution

### *Comment choisir les features ?*

Contrairement aux méthodes traditionnelles, les features ne sont pas prédéfinies selon un formalisme particulier (par exemple SIFT), mais apprises par le réseau lors de la phase d'entraînement. Les noyaux des filtres désignent les poids de la couche de convolution. Ils sont initialisés puis mis à jour par rétropropagation du gradient.

C'est là toute la force des réseaux de neurones convolutifs : ceux-ci sont capables de déterminer tout seul les éléments discriminants d'une image, en s'adaptant au problème posé. Par exemple, si la question est de distinguer les chats des chiens, les features automatiquement définies peuvent décrire la forme des oreilles ou des pattes.

#### 3.8.4.2 Couche de Pooling

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling.

L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2x2 pixels qui ne se chevauchent pas, ou des cellules de taille 3x3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc).

On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites.

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage

Les valeurs maximales sont repérées de manière moins précise dans les feature maps obtenues après pooling que dans celles reçues en entrée : c'est en fait un grand avantage ! En effet, lors-

qu'on veut reconnaître un chien par exemple, ses oreilles n'ont pas besoin d'être localisées le plus précisément possible : savoir qu'elles se situent à peu près à côté de la tête suffit !

Ainsi, la couche de pooling rend le réseau moins sensible à la position des features : le fait qu'une feature se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image.[11]

Il existe deux principaux types de couches de pooling :

- **Max Pooling** : Lorsque le filtre se déplace sur l'entrée, il sélectionne le pixel avec la valeur maximale à envoyer à la matrice de sortie.
- **Average Pooling** : Lorsque le filtre se déplace sur l'entrée, il calcule la valeur moyenne dans le champ réceptif à envoyer à la matrice de sortie.[42]

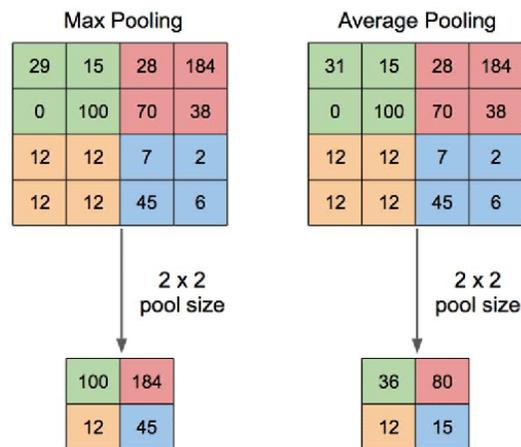


FIGURE 3.10 – Average Pooling vs Max Pooling

### 3.8.4.3 Couche de correction ReLU

ReLU (Rectified Linear Units) désigne la fonction réelle non linéaire définie par  $ReLU(x) = \max(0, x)$ .

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation

### 3.8.4.4 Couche Fully Connected

La couche fully-connected constitue toujours la dernière couche d'un réseau de neurones, convolutif ou non ; elle n'est donc pas caractéristique d'un CNN.

Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée.

La dernière couche fully-connected permet de classifier l'image en entrée du réseau : elle renvoie un vecteur de taille  $N$ , où  $N$  est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe.

Par exemple, si le problème consiste à distinguer les chats des chiens, le vecteur final sera de taille 2 : le premier élément (respectivement, le deuxième) donne la probabilité d'appartenir à la classe "chat" (respectivement "chien"). Ainsi, le vecteur  $[0.9 \ 0.1]$  signifie que l'image a 90% de chances de représenter un chat.

Chaque valeur du tableau en entrée "vote" en faveur d'une classe. Les votes n'ont pas tous la même importance : la couche leur accorde des poids qui dépendent de l'élément du tableau et de la classe.

Pour calculer les probabilités, la couche fully-connected multiplie donc chaque élément en entrée par un poids, fait la somme, puis applique une fonction d'activation (logistique si  $N=2$ , softmax si  $N>2$ ).

Ce traitement revient à multiplier le vecteur en entrée par la matrice contenant les poids. Le fait que chaque valeur en entrée soit connectée avec toutes les valeurs en sortie explique le terme fully-connected.

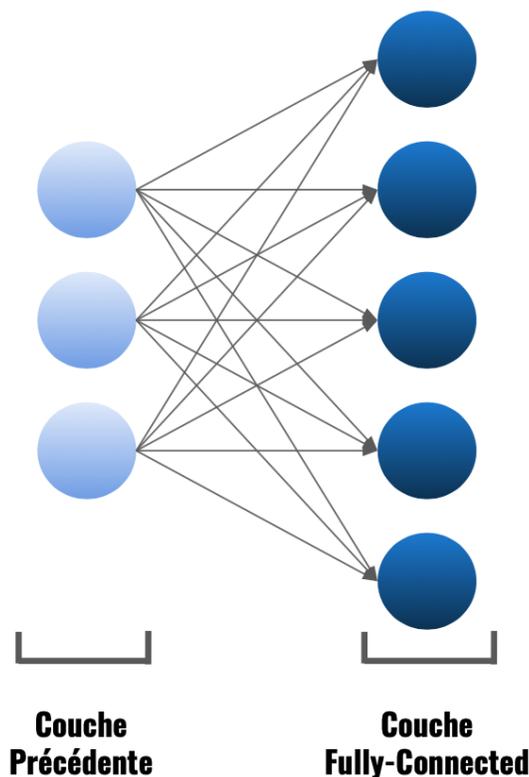


FIGURE 3.11 – Couche Fully Connected

*Comment connaît-on la valeur de ces poids ?*

Le réseau de neurones convolutif apprend les valeurs des poids de la même manière qu'il

apprend les filtres de la couche de convolution : lors de phase d'entraînement, par rétropropagation du gradient.

La couche fully-connected détermine le lien entre la position des features dans l'image et une classe. En effet, le tableau en entrée étant le résultat de la couche précédente, il correspond à une carte d'activation pour une feature donnée : les valeurs élevées indiquent la localisation (plus ou moins précise selon le pooling) de cette feature dans l'image. Si la localisation d'une feature à un certain endroit de l'image est caractéristique d'une certaine classe, alors on accorde un poids important à la valeur correspondante dans le tableau.[11]

### 3.8.4.5 Les fonctions d'activation

Les fonctions d'activation sont utilisées pour apprendre et approcher tout type de relation continue et complexe entre les variables du réseau. En termes simples, elles décident quelles informations du modèle doivent être déclenchées dans le sens direct et lesquelles ne doivent pas se déclencher à la fin du réseau. Elles ajoutent de la non-linéarité à ce dernier.

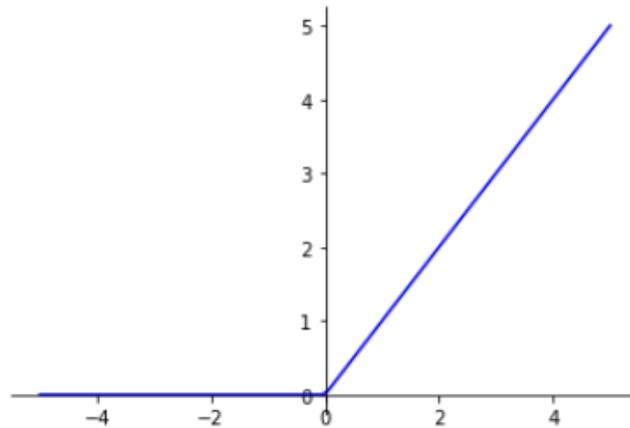
Il existe plusieurs fonctions d'activation couramment utilisées telles que ReLU, Softmax, tanH et Sigmoid. Chacune de ces fonctions a des particularités spécifiques.

Selon le problème à résoudre on utilise des fonctions d'activation différentes.[39]

- **ReLU :**

La fonction Rectified Linear Unit (RELU) est la fonction d'activation la plus simple et la plus utilisée, elle donne  $x$  si  $x > 0$ , 0 sinon. Autrement dit c'est le maximum entre  $x$  et 0. Cette fonction permet d'effectuer un filtre sur nos données. Elle laisse passer les valeurs positives ( $x > 0$ ) dans les couches suivantes du réseau de neurones. Elle est utilisée presque partout mais surtout pas dans le calcul final, elle est utilisée dans les couches intermédiaires.

$$\text{fonction\_ReLU}(x) = \max(x, 0)$$



Fonction ReLU – Rectified Linear Unit

FIGURE 3.12 – Graphe de la fonction ReLU

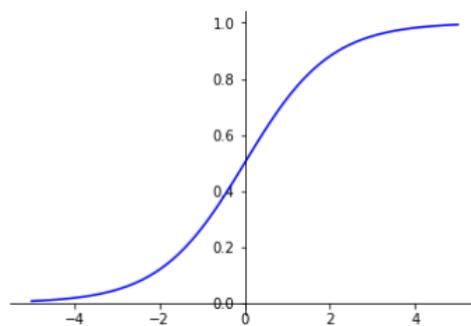
- **Sigmoïde :**

La fonction sigmoïde donne une valeur entre 0 et 1, une probabilité. Elle est donc très utilisée pour les classifications binaires, lorsqu'un modèle doit déterminer seulement deux labels.

Ainsi, pour la classification d'un critère (a), plus la valeur retournée par sigmoïde est proche de 1, plus le modèle considère que le critère (a) est positif. Au contraire, plus elle est proche de 0, plus elle est considérée comme négative.

La fonction sigmoïde est très simple à appliquer en python car il n'y'a pas de paramètres autres que la variable d'entrée.

$$\text{fonction\_Sigmoid}(x) = 1 / (1 + \exp(-x))$$



Fonction Sigmoïde

FIGURE 3.13 – Graphe fonction sigmoïde

- **Softmax :**

La fonction softmax permet elle de transformer un vecteur réel en vecteur de probabilité. On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les problèmes multiclasse.

Dans la fonction softmax, chaque vecteur est traité indépendamment. L'argument axis définit l'axe d'entrée sur lequel la fonction est appelée.

$$\text{fonction\_Softmax}(x) = \exp(x) / \text{tf.reduce\_sum}(\exp(x))$$

$$\text{fonction\_Softmax}(x) = \exp(x) / \text{sum}(\exp(x_i))$$

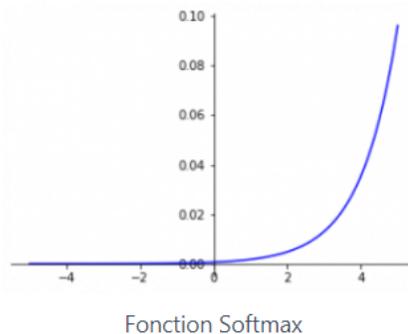


FIGURE 3.14 – Graphe de la fonction softmax

- **tanh :**

La fonction tanh est simplement la fonction de la tangente hyperbolique. Il s'agit en fait d'une version mathématiquement déclarée de la fonction sigmoïde :

Sigmoïde donne un résultat entre 0 et 1.

tanh donne un résultat entre -1 et 1.

L'avantage de tanh est que les entrées négatives seront bien répertoriées comme négatives. là où, avec sigmoïde, les entrées négatives peuvent être confondus avec les valeurs proches de nulles.

Cette fonction est, comme sigmoïde, mais fonctionne mieux dans la plupart des cas.

$$\text{fonction\_tanh}(x) = \sinh(x)/\cosh(x)$$

$$\text{fonction\_tanh}(x) = ((\exp(x) - \exp(-x))/(\exp(x) + \exp(-x)))$$

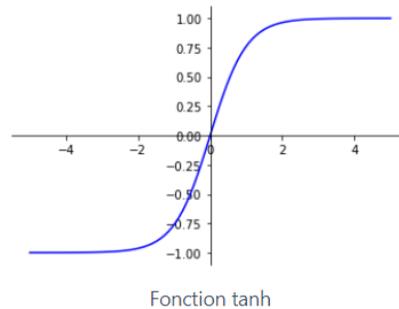


FIGURE 3.15 – Graphe de la fonction tanh

- **ELU** : La fonction Exponential Linear unit (ELU) est une amélioration de ReLU car elle permet d'avoir des valeurs lisses lorsque  $x < 0$ .

Lorsque  $x < 0$ , ELU a des valeurs négatives différentes de 0 (ce qui n'est pas le cas de ReLU). Ce qui permet de rapprocher la moyenne de la fonction de zero.

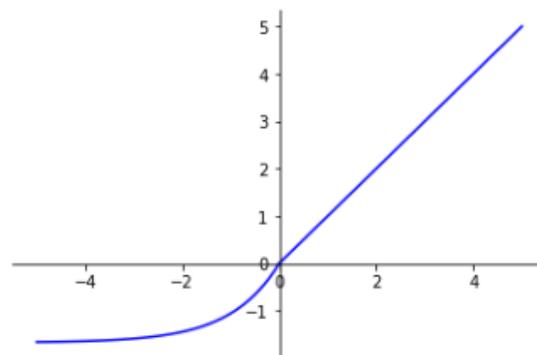
Une moyenne plus proche de 0 permet un apprentissage plus rapide car cela rapproche le gradient calculé du gradient naturel.

Effectivement, plus  $x$  diminue, plus ELU sature à une valeur négative. Cette saturation implique qu'ELU a une petite dérivée ce qui diminue la variation du résultat et donc l'information qui est propagée vers la couche suivante.

$$\text{ELU}(x) = x \text{ si } x > 0$$

$$\text{ELU}(x) = \alpha * (\exp(x) - 1) \text{ si } x < 0$$

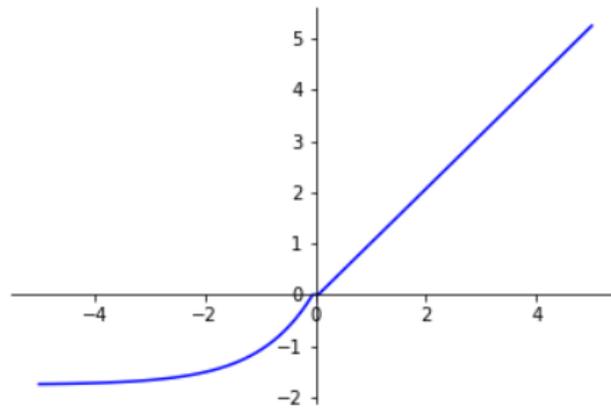
avec :  $\alpha > 0$



Fonction ELU – Exponential Linear Unit

FIGURE 3.16 – Graphe de la fonction ELU

- **SELU** : La fonction Scaled Exponential Linear Unit (SELU) est une optimisation de ELU. Le principe est le même qu'avec ELU, on multiplie seulement le résultat de ELU par un scalaire. On pourrait l'écrire comme cela :  $fonction_{SELU}(x) = scale * fonction_{ELU}(x)$ . scale et alpha sont prédéfinis, tel que scale est supérieure à 1. Cela permet à la pente de SELU sur  $x > 0$  d'être supérieure à 1 et évite certains problèmes lors du calcul de gradient.[39]



Fonction SELU – Scaled Exponential Linear Unit

FIGURE 3.17 – Graphe de la fonction SELU

Pour un modèle CNN de classification binaire, les fonctions sigmoïdes et softmax sont préférées et pour une classification multi-classes, nous utilisons généralement softmax. En termes simples, les fonctions d'activation dans un modèle CNN déterminent si un neurone doit être activé ou non. Il décide si l'apport au travail est important ou non à prédire à l'aide d'opérations mathématiques.

### 3.8.4.6 Les couches dropout

Le dropout est une technique permettant de réduire l'overfitting lors de l'entraînement du modèle. Le terme "dropout" fait référence à la suppression de neurones dans les couches d'un modèle de deep learning.

En effet, on désactive temporairement certains neurones dans le réseau, ainsi que toutes ses connexions entrantes et sortantes. Ceux-ci sont désactivés aléatoirement.

Le dropout est une méthode de régularisation, qui est un ensemble de méthodes qui permet à la fois d'optimiser l'apprentissage d'un modèle de deep learning et de contrer l'overfitting (manque de généralisation).[12]

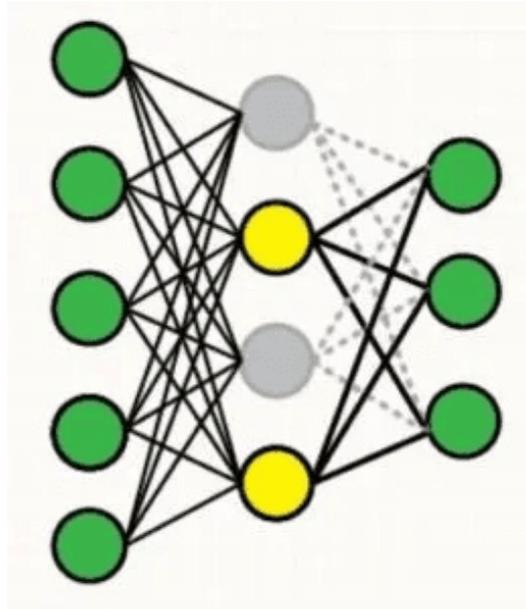


FIGURE 3.18 – Fonctionnement de la couche Dropout

#### 3.8.4.7 Les filtres

Un filtre est une matrice de pixels qui vient se poser sur une surface de l'image afin d'extraire des features que l'on souhaite retrouver dans les images.

### 3.8.5 Quelques réseaux CNN

Plusieurs architectures de CNN sont disponibles et jouent un rôle important dans le développement d'algorithmes dans différents domaines. Nous citons certains d'entre eux ci-dessous :

- *LeNet*

C'est une des premières architectures de CNN développée. Il est composé de sept couches de neurones : trois couches de convolution, une couche de pooling, une couche de normalisation, une couche de sortie fully-connected et une couche de classification. LeNet a été entraîné sur de nombreuses tâches de vision par ordinateur, y compris la classification d'images, la détection d'objets et la reconnaissance de chiffres manuscrits. Il a également été utilisé pour d'autres tâches d'apprentissage automatique, telles que la classification de texte et la classification de signaux audio. Il peut traiter des images en 32x32 pixels et sa fonction d'activation est ReLu

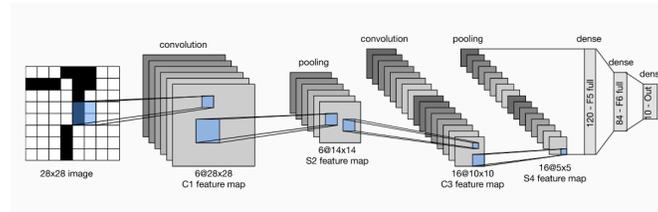


FIGURE 3.19 – Architecture LeNet

- *AlexNet*

AlexNet est le nom d'une architecture de réseau neuronal convolutif (CNN), conçue par Alex Krizhevsky en collaboration avec Ilya Sutskever et Geoffrey Hinton, qui était le Ph.D. de Krizhevsky. Utilisé pour la première fois en 2012 lors de la compétition ILSVRC, il est composé de 5 couches de convolution et 3 couches fully-connected. Il utilise des techniques d'apprentissage profond telles que l'utilisation de couches de pooling, de dropout et de normalisation récurrente.

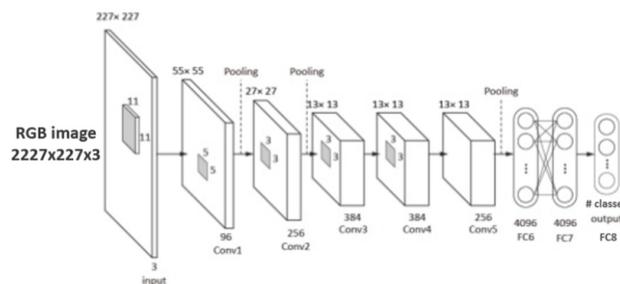


FIGURE 3.20 – Architecture AlexNet

Les réseaux de neurones convolutifs sont capables d'apprendre des modèles complexes dans les données et d'obtenir des performances de pointe sur une variété de tâches. Ils sont efficaces également pour capturer les caractéristiques spatiales et temporelles des vidéos. Mais ce sont des systèmes de calcul intensifs qui nécessitent des GPUs pour un entraînement efficace.

### 3.9 Transfer Learning

L'apprentissage par transfert est de plus en plus évoqué en Data Science faisant référence à un ensemble de méthodes qui permettent de transférer les connaissances acquises à partir de la résolution de problèmes donnés pour traiter un autre problème.

Le Transfer Learning a connu un grand succès avec l'essor du Deep Learning. En effet, bien souvent, les modèles utilisés dans ce domaine nécessitent des temps de calcul élevés et des ressources importantes. Or, en utilisant des modèles pré-entraînés comme point de départ, le Transfer Learning permet de développer rapidement des modèles performants et résoudre efficacement des problèmes complexes en Computer Vision ou Natural Language Processing (NLP). [35]

## 3.10 Overfitting

Le surajustement en Machine Learning se produit quand un modèle ajuste trop bien les données d'entraînement, et est donc dans l'incapacité de prédire avec précision des données de test invisibles. En d'autres termes, le modèle a simplement mémorisé du bruit et des patterns spécifiques dans les données d'entraînement, mais il n'est pas suffisamment flexible pour faire des prédictions sur des données réelles.[21]

L'overfitting arrive quand le modèle choisi est trop complexe par rapport à la taille du jeu de données.[22]

Pour surmonter ce problème, une couche de suppression (Dropout Layer) est utilisée, dans laquelle quelques neurones sont supprimés du réseau neuronal, pendant le processus de formation, ce qui réduit la taille du modèle.

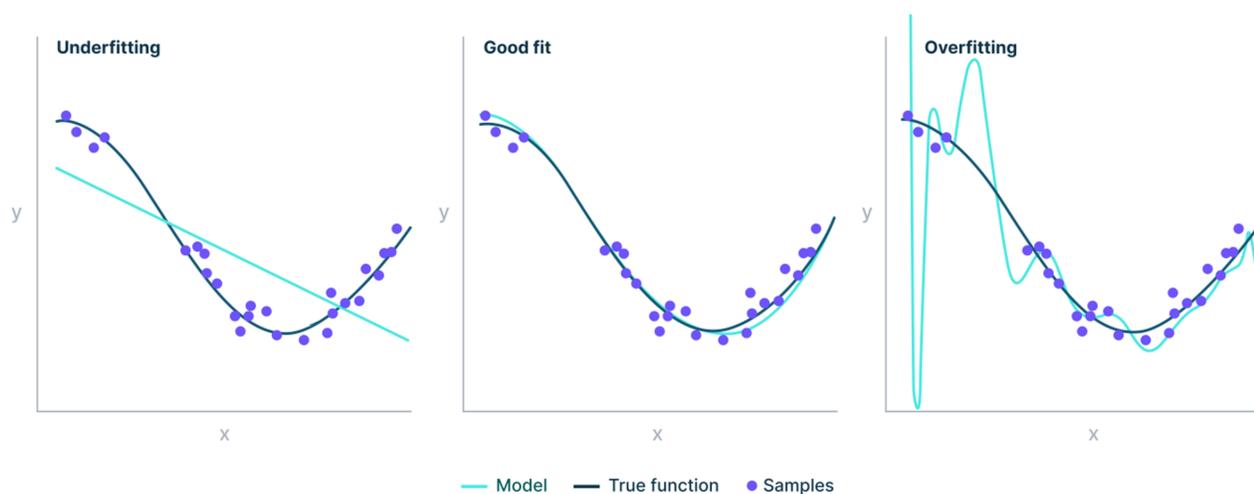


FIGURE 3.21 – Overfitting

## 3.11 Conclusion

Dans ce chapitre nous avons défini ce qu'est l'intelligence artificielle puis avons présenté l'intérêt de l'IA dans le domaine médical, ensuite nous avons vu ce qu'est le machine learning ainsi que les types d'apprentissage existant. Nous nous sommes familiarisés avec le deep learning, les réseaux de neurones et leur architecture et fonctionnement et certains modèles de réseaux. Enfin nous avons parlé du transfer learning et de l'overfitting et comment résoudre ce dernier.

# Etat de l'art

## 4.1 Introduction

Ce chapitre est consacré à l'étude de ce qui a été publié dans la littérature sur le sujet du traitement d'IRM du cerveau avec des techniques de deep learning.

Ce premier article relève de la détection de tumeurs du cerveau sur des images d'IRM en utilisant une architecture CNN hybride. Les auteurs, Cinar et al. [60], ont utilisé Alexnet, Resnet50, Densenet201, GoogleNet et Inception V3 ainsi que ResNet 50 Modifié.

Les auteurs se sont basés sur un algorithme ResNet 50 pré-entraîné, ont effacé les 5 dernières couches et ont ajouté 10 couches supplémentaires pour améliorer le modèle. Ces dernières couches sont respectivement composées de ReLu, Convolution, Batch Normalisation, Dropout, Fully Connected, Relu, Max Pooling, Fully Connected, SoftMax et des Couches de classification. Le nombre de couches est passé de 177 couches à 182 couches.

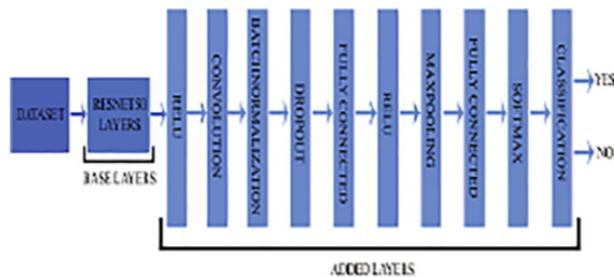


FIGURE 4.1 – Structure du modèle amélioré

Analysis Result			
	Name	Type	Activations
1	Imageinput	Image Input	$224 \times 224 \times 1$
2	conv_1	Convolution	$112 \times 112 \times 64$
172	add_16	Addition	$7 \times 7 \times 2048$
173	Relu	Relu	$7 \times 7 \times 2048$
174	conv_2	Convolution	$7 \times 7 \times 32$
175	Batchnorm	Batch Normalization	$7 \times 7 \times 32$
176	Dropout	Dropout	$7 \times 7 \times 32$
177	fc_1	Fully Connected	$1 \times 1 \times 2$
178	activation	Relu	$1 \times 1 \times 2$
179	Maxpool	Max Pooling	$1 \times 1 \times 2$
180	fc_2	Fully Connected	$1 \times 1 \times 2$
181	fc1000_soft	Softmax	$1 \times 1 \times 2$
182	Classoutput	Classification Output	-

FIGURE 4.2 – Couches Ajoutées

Solver Name	Sgdm
MiniBatchSize	16
MaxEpochs	5
InitialLearnRate	1.0000e-04
Shuffle	every-epoch
ValidationFrequency	3
Total Iteration	160

FIGURE 4.3 – Valeurs principales utilisées lors du training des réseaux

Dans cet article, on note que les cinq algorithmes cités plus haut ont donné de bons résultats avec le dataset utilisé, avec ResNet 50 donnant les meilleurs résultats. En comparaison avec l'algorithme hybride proposé (ResNet 50 Hybridé), les résultats de ce dernier dépassent ceux de ResNet 50 classique (Une accuracy de 97.01 pour l'hybride contre 92.54 pour ResNet50). Il a été conclu que le modèle proposé dans l'article peut être utilisé dans des systèmes assistés par ordinateur pour détecter une tumeur au cerveau.

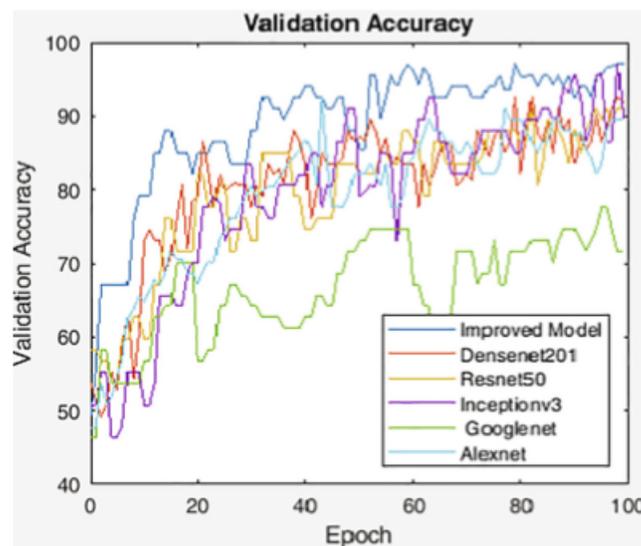


FIGURE 4.4 – Validation Accuracy Model

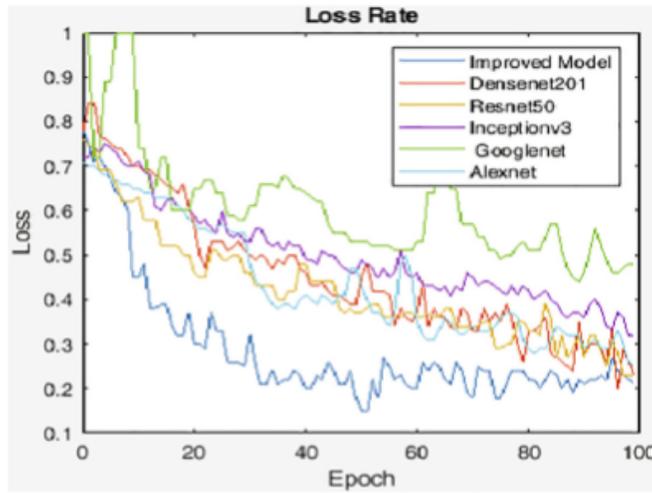


FIGURE 4.5 – Taux de Perte

	Accuracy	FPR	FDR	FNR
Improved Model	97.01	0	0	0.0526
Densenet201	91.04	0	0	0.1429
Resnet50	92.54	0.0357	0.0278	0.1026
Inceptionv3	88.06	0	0	0.1818
Googlenet	71.64	0.0714	0.0278	0.3396
Alexnet	89.55	0.0714	0.0556	0.1282

FIGURE 4.6 – Critere de performance

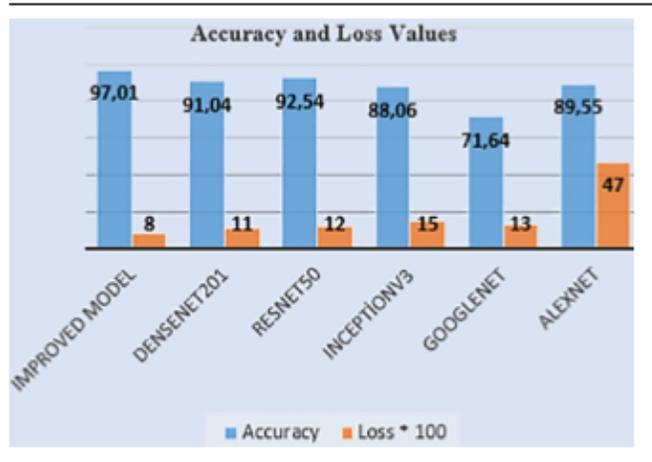


FIGURE 4.7 – Precision et valeurs des modeles

Les auteurs, Venkatesan et al.[55], proposent une comparaison d'algorithmes CNN sur deux datasets combinés, puis associent le résultat à des algorithmes de machine learning afin de trouver le meilleur modèle pour classifier des IRM du cerveau de façon exacte.

Les auteurs ont pris 5 algorithmes de CNN pré entraînés (VGG16, VGG19, ResNet101, ResNet50 et Alexnet) et les ont comparés sur les datasets BRATS, TCIA, un dataset clinique et la combinaison des deux premiers datasets (BRATS et TCIA) contenant des images IRM de type FLAIR, T2 et T1C.

Image Class	Modality	Number of Images for Training	Number of Images for Testing
Normal (BRATS+TCIA)	Mixed (Flair+T2)	1000	400
Abnormal (BRATS)	Flair	1500	600
	T2	1500	600
	T1C	1500	600
Abnormal (TCIA)	T2	1000	400
Abnormal (Clinical)	T2	200	200

FIGURE 4.8 – Images utilisees

Les auteurs ont ensuite pris le meilleur modèle résultant et ont remplacé le classificateur original SoftMax par d'autres classificateurs comme SVM-RBF, SVM-Linear, KNN et les arbres de décision.

Par la suite, ils ont comparé les résultats pour déduire la meilleure combinaison avec le modèle VGG 19 pré-entraîné selon différents critères de performance. L'objectif est d'implémenter dans le futur une technique de fusion pour aider à améliorer la précision de la détection de tumeurs..

Network	Modality	TP	FN	TN	FP	ACC	PRE	SEN	SPE	F1S	NPV
AlexNet	Flair	377	23	581	19	95.80	95.20	94.25	96.83	94.72	96.19
	T2	368	32	573	27	94.10	93.16	92.00	95.50	92.58	94.71
	T1C	361	39	568	32	92.90	91.86	90.25	94.67	91.05	93.57
VGG16	Flair	379	21	585	15	96.40	96.19	94.75	97.50	95.47	96.53
	T2	364	36	570	30	93.40	92.39	91.00	95.00	91.69	94.06
	T1C	368	32	564	36	93.20	91.09	92.00	04.00	91.54	94.63
VGG19	Flair	381	19	584	16	96.50	95.97	95.25	97.33	95.61	96.85
	T2	378	22	581	19	95.90	95.21	94.50	96.83	94.86	96.35
	T1C	370	30	574	26	94.40	93.43	92.50	95.67	92.96	95.03
ResNet50	Flair	371	29	567	33	93.80	91.83	92.75	94.50	92.29	95.13
	T2	358	42	565	35	92.30	91.09	89.50	94.17	90.29	93.08
	T1C	362	38	566	34	92.80	91.41	90.50	94.33	90.95	93.71
ResNet101	Flair	374	26	583	17	95.70	95.65	93.50	97.17	94.56	95.73
	T2	361	39	568	32	92.90	91.85	90.25	94.67	91.05	93.57
	T1C	366	34	565	35	93.10	91.27	91.50	94.17	91.39	94.32

FIGURE 4.9 – Algorithmes utilises

Il est déduit dans cet article, qu'avec les datasets BRATS, TCIA et le dataset clinique, VGG-19 est le meilleur algorithme de CNN comparés. Les résultats globaux obtenus avec VGG 19 et le classificateur SVM-RBF ont permis d'obtenir de meilleurs résultats sur les images des modalités Flair, T2 et T1C. La performance a été confirmée avec une validation croisée décuplée et des précisions de classification  $>99\%$ ,  $>98\%$  et  $>97\%$  pour ses dernières respectivement.

Classifier	Modality	TP	FN	TN	FP	ACC	PRE	SEN	SPE	FIS	NPV
DT	Flair	379	21	587	13	96.60	96.68	94.75	97.83	95.71	96.55
	T2	377	23	584	16	96.10	95.93	94.25	97.33	95.08	96.21
	T1C	368	32	571	29	93.90	92.69	92.00	95.17	92.35	94.69
KNN	Flair	369	31	581	19	95.00	95.10	92.25	96.83	93.65	94.93
	T2	371	29	580	20	95.10	94.88	92.75	96.67	93.80	95.24
	T1C	368	32	578	22	94.60	94.36	92.00	96.33	93.16	94.75
SVM-Linear	Flair	370	30	583	17	95.30	95.61	92.50	97.17	94.03	95.11
	T2	374	26	579	21	95.30	94.68	93.50	96.50	94.09	95.70
	T1C	372	28	571	29	94.30	92.77	93.00	95.17	92.88	95.33
SVM-RBF	Flair	378	22	589	11	96.70	97.17	94.50	98.17	95.82	96.40
	T2	374	26	582	18	95.60	95.41	93.50	97.00	94.44	95.72
	T1C	369	31	574	26	94.30	93.42	92.25	95.67	92.83	94.88

FIGURE 4.10 – Classificateurs utilisés

Les auteurs, Hassan et al. [49], effectuent une étude comparative entre leur modèle proposé de CNN et les modèles pré-entraînés de CNN suivants : VGG 16, ResNet 50 et Inception V3.

Les auteurs ont appliqué des techniques de traitement d'images et d'augmentation sur un dataset de 253 images d'IRM cérébrale dont 155 images de cancer malin, et 98 de tumeurs bénignes non cancéreuses.

Les auteurs ont créé un modèle CNN et comparé la précision de ce dernier avec celle des modèles pré-entraînés VGG 16, ResNet 50, Inception V3, en utilisant le transfer learning.

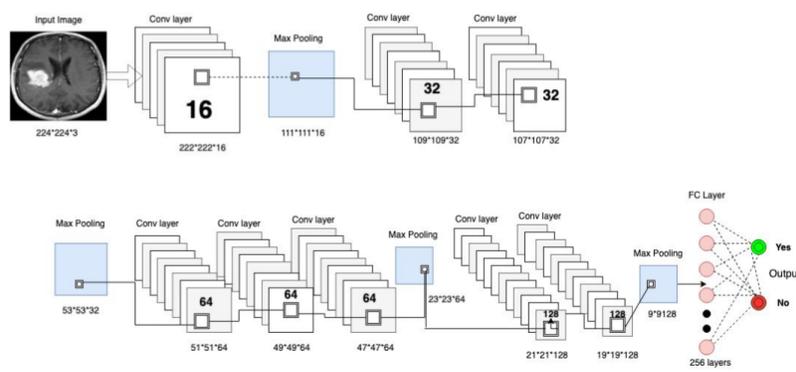


FIGURE 4.11 – Architecture du modèle de CNN proposé

Le modèle proposé contient 8 couches de convolution (**Figure 4.11**) : la première couche contient 16 filtres 3x3, suivit d'une couche Max Pooling avec un filtre 2x2, suivit de deux couches de convolution avec 32 filtres de taille 3x3, une couche de Max Pooling, trois couches de convolution de 64 filtres de taille 3x3, une couche Max Pooling, deux couches de convolution de 128 filtres de taille 3x3, et une dernière couche Max Pooling. Enfin une couche Fully connected dense avec un couche de sortie Softmax qui calcule le score de probabilité pour chaque classe, et classe les étiquettes de décision finale indiquant si l'image IRM d'entrée contient un cancer ou non.

Après comparaison, il est conclu que VGG 16 et le modèle de CNN proposé par les auteurs donnent de meilleurs résultats que Resnet 50 ou Inception V3.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC Score	Time(sec)
Proposed CNN	1.0	1.0	1.0	1.0	1.0	3085
VGG-16	0.96	0.93	1.0	0.97	0.96	6846
ResNet-50	0.89	0.87	0.93	0.90	0.89	9091
Inception-v3	0.75	0.77	0.71	0.74	0.75	5630

FIGURE 4.12 – Métriques d'évaluation

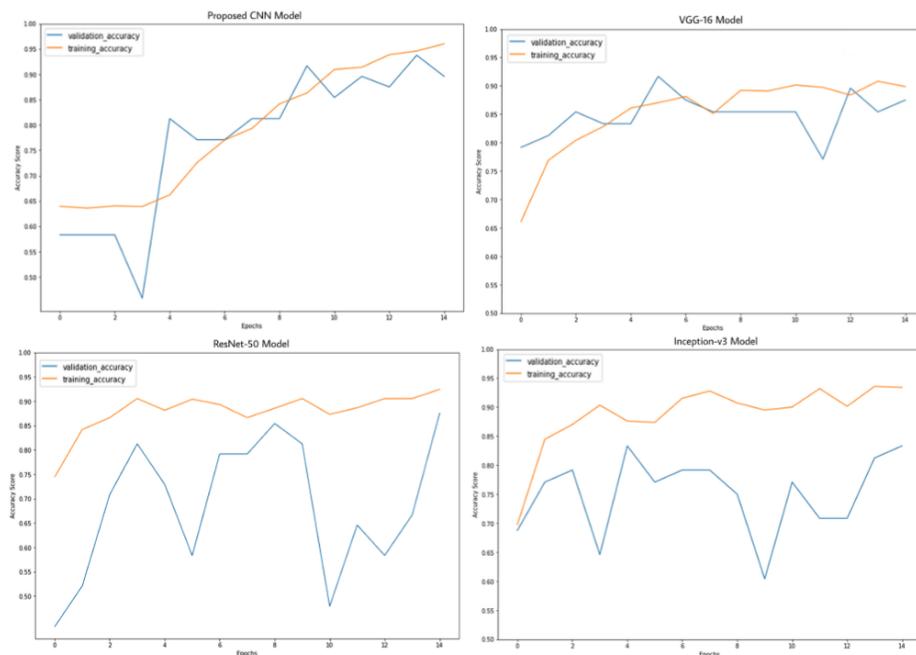


FIGURE 4.13 – Graphes de Precision des algorithmes etudies

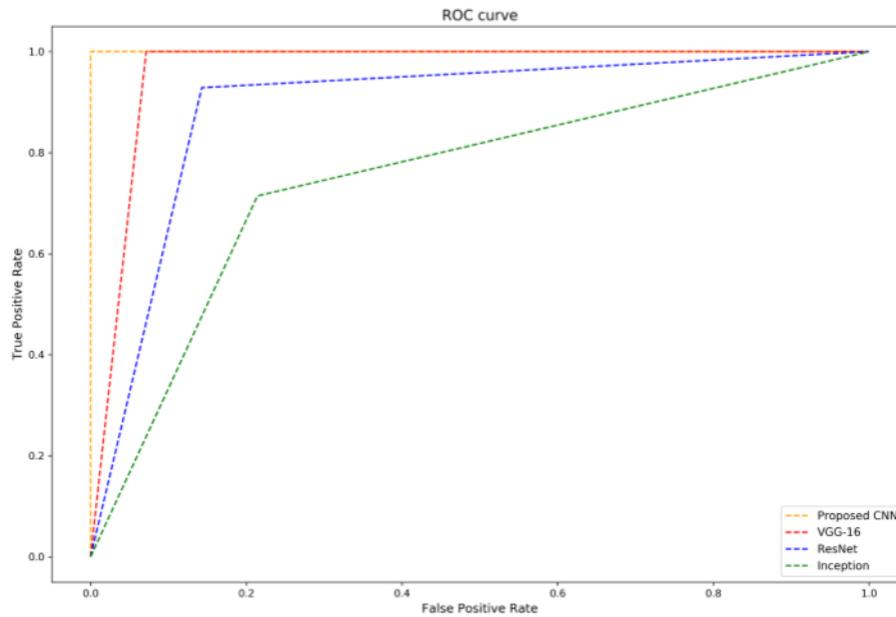


FIGURE 4.14 – Courbe ROC-AUC

## 4.2 Synthèse bibliographique

Auteur/Année	Méthode	Accuracy
Ahmet éinar and Muham- med Yildirim / 2020	Improved Model	97,01%
Rajinikanth, Venkatesan and Joseph Raj, Alex Noel and Thanaraj, Krishnan Palani and Naik, Ganesh R. / 2020	VGG 19 SVM-RBF	>99%
Hassan Ali Khan and Wu Jue and Muhammad Mush- taq and Muhammad Umer Mushtaq / 2020	Proposed CNN	100%

TABLEAU 4.1 – Tableau de Synthèse

## 4.3 Conclusion

Les études montrent qu'en général VGG a une meilleure accuracy que Resnet-50 et InceptionV3. Nous allons implémenter ces différents modèles et comparer les résultats obtenus afin de tirer celui qui résoudra le mieux notre problématique dans le chapitre suivant.

## Approche utilisée et Conception

### 5.1 Introduction

Dans ce chapitre, nous proposons une approche basée sur le Transfer Learning pour résoudre notre problème de classification en deux classes : "Yes" pour l'existence d'une tumeur et "No" pour l'absence de tumeur.

L'idée est d'utiliser un modèle pré-entraîné qui a déjà appris des caractéristiques à partir de grandes quantités de données, plutôt que de former un nouveau modèle à partir de zéro. Cette approche présente plusieurs avantages, notamment en termes de gain de temps et de ressources. En utilisant le Transfer Learning, nous pouvons tirer parti des connaissances préalables du modèle pré-entraîné et les appliquer à notre tâche spécifique de classification. Cela permet d'améliorer les performances du modèle et d'accélérer le processus d'apprentissage, car une grande partie de l'apprentissage des caractéristiques générales a déjà été effectuée.

Dans notre approche de Transfer Learning, nous utilisons cinq modèles de référence : VGG16, VGG19, ResNet50, GoogLeNet et InceptionV3. Ces modèles ont été pré-entraînés sur de vastes ensembles de données, ce qui leur permet d'extraire des caractéristiques significatives à partir d'images. Nous allons fournir une brève présentation de chacun des modèles, en expliquant leur définition et leur architecture.

Par la suite nous expliquerons notre approche pour l'implémentation de ces modèles. Nous avons entraîné ces modèles sur deux datasets distincts. Nous avons modifié les epochs du modèles VGG19 afin d'avoir un temps d'entraînement équivalent à celui des autres modèles choisis.

Enfin nous analyserons les résultats d'entraînement grâce à des matrices de confusion ainsi que différentes statistiques.

## 5.2 Modèles utilisés pour le Transfer Learning

### 5.2.1 Premier modèle : Inception V3

Inception v3 est un réseau neuronal convolutif pour aider à l'analyse d'images et à la détection d'objets, et a fait ses débuts en tant que module pour GoogLeNet. Il s'agit de la troisième édition du réseau de neurones convolutionnels Inception de Google, initialement présenté lors du défi de reconnaissance ImageNet. La conception d'Inceptionv3 visait à permettre des réseaux plus profonds tout en évitant que le nombre de paramètres ne devienne trop important : il a "moins de 25 millions de paramètres", contre 60 millions pour AlexNet. Il est basé sur l'article original *Rethinking the Inception Architecture for Computer Vision* de Szegedy, et. al.[59]

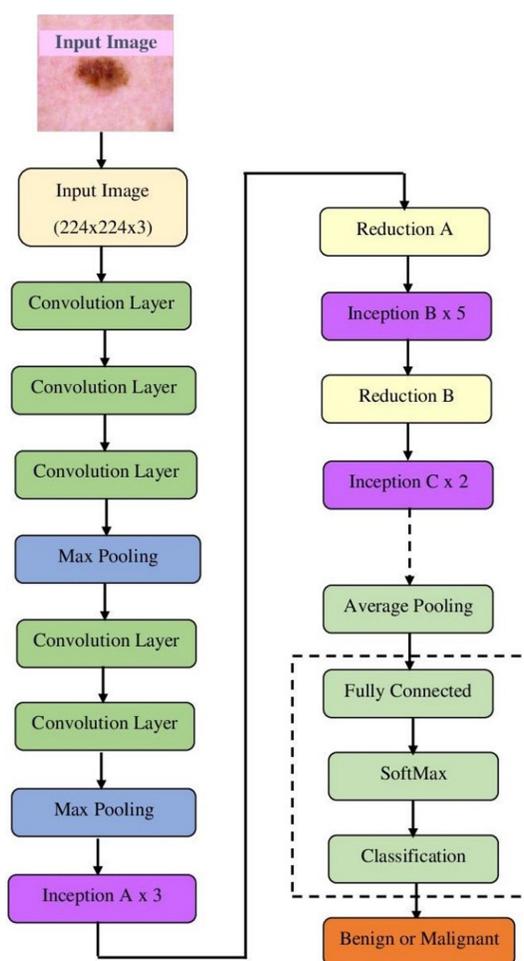


FIGURE 5.1 – Architecture InceptionV3

## 5.2.2 Deuxième modèle : GoogLeNet

Une famille de réseaux de neurones convolutionnels basé sur une architecture construite à partir de plusieurs modules Inception. Réalisé en 2014 pour ILSVRC. GoogLeNet se compose de 9 modules Inception arrangés en par ligne. Il a 22 couches de profondeurs (27 si on inclut les couches de pooling) et utilise Global Average Pooling à la fin du dernier module Inception. Cette architecture permet un niveau presque humain de reconnaissance d'images.[?]

# Inception-V1 (GoogLeNet)

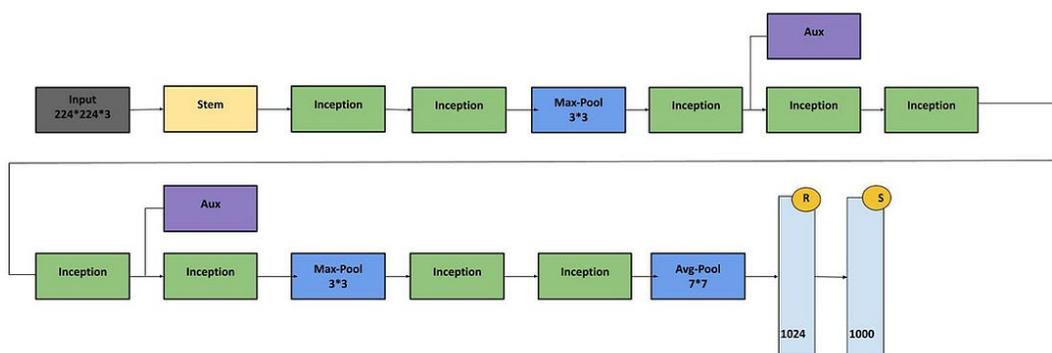


FIGURE 5.2 – Architecture GoogLeNet

## 5.2.3 Troisième et Quatrième modèles : VGG16 et VGG19

C'est une famille de modèles de réseaux de neurones convolutifs de grande profondeur (19 couches) construits par Visual Geometry Group (VGG) à l'Université d'Oxford. Ils ont été introduits dans le papier "Very Deep Convolutional Networks for Large-Scale Image Recognition" en 2014. Le modèle VGG est composée de 16 (VGG-16) ou 19 (VGG-19) couches de convolution et de pooling, suivies de 3 ou 4 couches de réseaux fully-connected à la fin. Les couches de pooling sont remplacées par des strided convolutions dans certaines versions.[42]

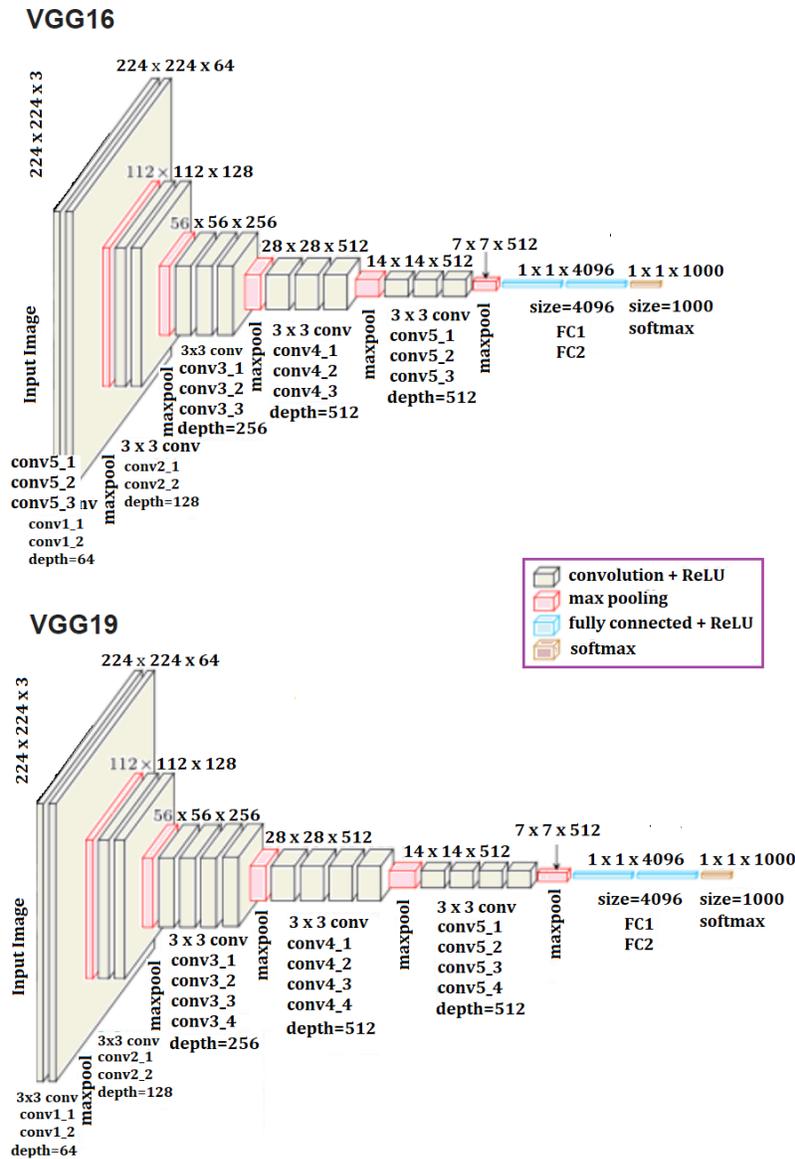


FIGURE 5.3 – Architecture VGG16 et VGG19

### 5.2.4 Cinquième modèle : ResNet50

Introduit dans le papier "Deep Residual Learning for Image Recognition" en 2015. C'est un réseau de neurones profond (50 couches) composé de plusieurs blocs de modules de raccourcissement de chemin. Un module de raccourcissement de chemin (shortcut) est composé de plusieurs couches de convolution et de pooling, suivies d'une addition nulle de l'entrée de module à sa sortie. La structure des raccourcissements de chemin permet de résoudre le problème de l'apprentissage profond (vanishing / exploding gradients).

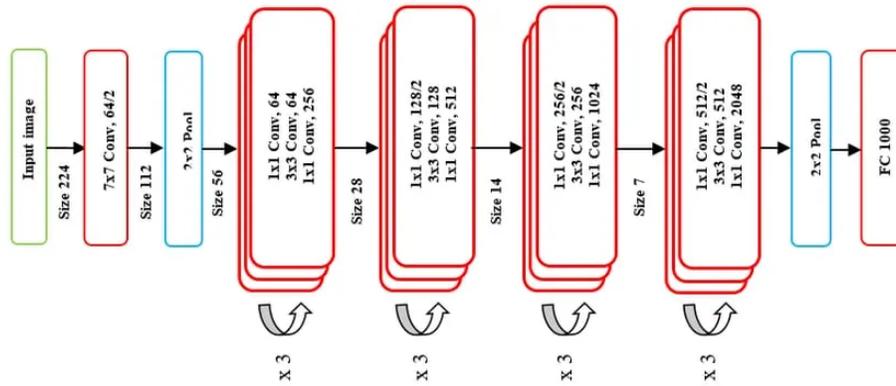


FIGURE 5.4 – Architecture ResNet

### 5.3 Comment choisir les modèles ?

Basée sur les articles étudiées et présentées dans le chapitre précédent, nous nous sommes basées sur la pourcentage de precision des résultats experimentaux pour choisir les meilleurs modèles à utiliser avec les deux datasets choisis.

# Implémentation et Résultats

## 6.1 Introduction

Dans ce dernier chapitre, nous aborderons l'implémentation de différentes architectures CNN pour la détection de tumeurs cérébrales, afin d'établir le meilleur modèle à utiliser dans notre cas. Ces modèles sont appliqués sur des images de résonance magnétique (IRM) du cerveau.

Nous abordons ici la problématique de la détection des tumeurs cérébrales qui consiste à attribuer à une image IRM d'entrée  $x$  une étiquette  $y$  à partir d'un ensemble de deux classes : oui, pour l'existence d'une tumeur et non pour son absence.

Cette problématique, d'apparence simple, donne lieu au développement de diverses applications dans les domaines de la vision par ordinateur et de l'informatique médicale (e-health).

Nous baserons notre choix du meilleur modèle, sur différents critères tel l'efficacité, le temps, la précision (exactitude), etc.

## 6.2 Algorithmes Implémentés

Dans notre travail nous avons comparés les modèles de CNN pré entraînés suivants : VGG 16[43], VGG 19[54], Resnet 50[53], Inception V3[52] et GoogLeNet[56].

Nous avons utilisé ces modèles car ils ont été utilisés dans plusieurs études dont les articles [49], [55], [60] et ont prouvé être fiable. Ce sont aussi parmi les modèles les plus utilisées pour la classification d'imagerie médical outre la détection de tumeurs cérébrales.

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 6.1 – VGG 16

Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 7, 7, 512)	20024384
flatten (Flatten)	(None, 25088)	0
batch_normalization (Batch Normalization)	(None, 25088)	100352
dense (Dense)	(None, 256)	6422784
batch_normalization_1 (Batch Normalization)	(None, 256)	1024
activation (Activation)	(None, 256)	0
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

=====  
 Total params: 26,548,801  
 Trainable params: 6,473,729  
 Non-trainable params: 20,075,072  
 =====

FIGURE 6.2 – VGG 19

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23587712
flatten_2 (Flatten)	(None, 100352)	0
batch_normalization_4 (Batch Normalization)	(None, 100352)	401408
dense_4 (Dense)	(None, 256)	25690368
batch_normalization_5 (Batch Normalization)	(None, 256)	1024
activation_2 (Activation)	(None, 256)	0
dropout_2 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 1)	257
=====		
Total params: 49,680,769		
Trainable params: 25,891,841		
Non-trainable params: 23,788,928		

FIGURE 6.3 – resnet50

Layer (type)	Output Shape	Param #
inception_v3 (Functional)	(None, 5, 5, 2048)	21802784
flatten (Flatten)	(None, 51200)	0
batch_normalization_94 (Batch Normalization)	(None, 51200)	204800
dense (Dense)	(None, 256)	13107456
batch_normalization_95 (Batch Normalization)	(None, 256)	1024
activation_94 (Activation)	(None, 256)	0
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257
=====		
Total params: 35,116,321		
Trainable params: 13,210,625		
Non-trainable params: 21,905,696		

FIGURE 6.4 – Inception V3

## 6.3 Ensemble de données d'entraînement de validation et de test

### 6.3.1 Description des datasets

Pour la comparaison de ces modèles CNN qui aident à détecter les tumeurs cérébrales, nous avons utilisé deux jeux de données (datasets) comprenant des images IRM du cerveau. Nous avons utilisé le dataset Brain tumor detection et le dataset br35H qui ont tous les deux été obtenus sur la plateforme Open-source Kaggle.

Le premier[41] a été créé en 2019 par Navoneel Chakrabarty. Il se compose de 253 images de type jpg, séparées en deux classes : Une classe 'yes' avec 155 images, qui indiquent la présence d'une tumeur, et une classe 'no' avec 98 images, qui indiquent l'absence de tumeurs.

Le deuxième[46] a été créé en 2020 par Ahmed Hamada, pour aider à la détection et à la classification de tumeurs cérébrales en utilisant des modèles CNN et Deep Learning. Il se compose de 3000 images de type jpg, séparées en deux classes : Une classe 'yes' avec 1500 images, qui indique la présence d'une tumeur, et une classe 'no' avec 1500 images, qui indique l'absence de tumeurs.

Les ensembles de données sont ensuite redistribués de façon aléatoire en ensemble d'apprentissage et ensemble de test, tel que 80% des images sont dans l'ensemble de test.

### 6.3.2 Redimensionner les Images

Redimensionner les images fait partie de la normalisation du contenu du dataset. En machine learning, la normalisation des données vise à uniformiser les plages de valeurs des différentes variables numériques d'un jeu de données. Cela permet de garantir une comparaison équitable entre les variables et évite tout biais causé par des différences de magnitude. La normalisation permet ainsi d'améliorer la convergence des algorithmes d'apprentissage, de prévenir les problèmes numériques et facilite l'interprétation des coefficients des variables. Tout cela se fait sans perte d'informations significatives

Les Images d'IRM ayant des tailles variées généralement supérieure à 500X500 pixel, nous avons redimensionné les images afin qu'elles aient toutes les dimensions 224x224.

```
import imutils
img = cv2.imread('../content/drive/MyDrive/dataset/Brain_Tumor_Detection/no/No12.jpg')
img = cv2.resize(
    img,
    dsize=IMG_SIZE,
    interpolation=cv2.INTER_CUBIC
)
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)
```

FIGURE 6.5 – Resize

### 6.3.3 Recadrage et Detection

Le recadrage des images en machine learning consiste à découper les images pour plusieurs raisons : éliminer les parties non pertinentes, réduire les variations d'échelle, améliorer la robustesse du modèle et réduire la complexité du modèle. Cela permet d'obtenir des images cohérentes, plus faciles à traiter, et d'améliorer les performances du modèle sans perte d'informations essentielles.

L'étape pour recadrer les images est faite comme suit :

```
import imutils
img = cv2.imread('../content/drive/MyDrive/dataset/Brain_Tumor_Detection/no/No12.jpg')
img = cv2.resize(
    img,
    dsize=IMG_SIZE,
    interpolation=cv2.INTER_CUBIC
)
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)

# threshold the image, then perform a series of erosions +
# dilations to remove any small regions of noise
thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
thresh = cv2.erode(thresh, None, iterations=2)
thresh = cv2.dilate(thresh, None, iterations=2)

# find contours in thresholded image, then grab the largest one
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
c = max(cnts, key=cv2.contourArea)

# find the extreme points
extLeft = tuple(c[c[:, :, 0].argmin()][0])
extRight = tuple(c[c[:, :, 0].argmax()][0])
extTop = tuple(c[c[:, :, 1].argmin()][0])
extBot = tuple(c[c[:, :, 1].argmax()][0])

# add contour on the image
img_cnt = cv2.drawContours(img.copy(), [c], -1, (0, 255, 255), 4)

# add extreme points
img_pnt = cv2.circle(img_cnt.copy(), extLeft, 8, (0, 0, 255), -1)
img_pnt = cv2.circle(img_pnt, extRight, 8, (0, 255, 0), -1)
img_pnt = cv2.circle(img_pnt, extTop, 8, (255, 0, 0), -1)
img_pnt = cv2.circle(img_pnt, extBot, 8, (255, 255, 0), -1)

# crop
ADD_PIXELS = 0
new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS, extLeft[0]-ADD_PIXELS:extRight[0]+ADD_PIXELS].copy()
```

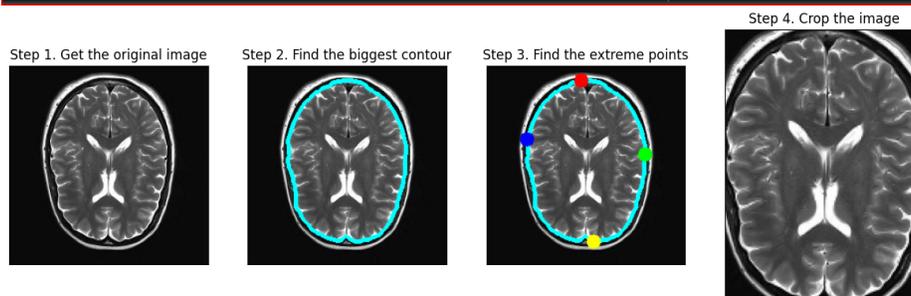


FIGURE 6.6 – Recadrage

## 6.4 Implémentation

### 6.4.1 Langage de Programmation utilisé

#### 6.4.1.1 Python 3.11.3

Pour ce projet, nous utilisons le langage de programmation Python. Python est un langage informatique généraliste, très utilisé pour l'analyse et le traitement de données, le machine learning et le deep learning. Il contient un nombre important de bibliothèques très performantes et très utiles dans la conception des réseaux de neurones.

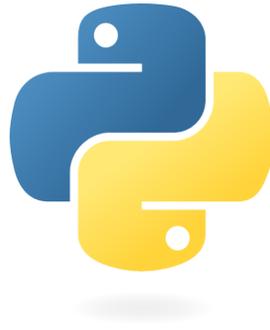


FIGURE 6.7 – Logo Python (source :[13])

### 6.4.2 Frameworks, IDE et Bibliothèques Python

#### 6.4.2.1 Google Colab

Google Colaboratory ("Colab" en abrégé) est un outil d'analyse de données et d'apprentissage automatique qui vous permet de combiner du code Python exécutable et du texte enrichi avec des graphiques, des images, HTML, LaTeX et plus encore dans un seul document stocké dans Google Drive. Il se connecte aux puissants environnements d'exécution de Google Cloud Platform et vous permet de partager facilement votre travail et de collaborer avec d'autres.



FIGURE 6.8 – Google Colab (source :[14])

### 6.4.2.2 Keras

Keras est une API de réseau de neurones écrite en langage Python. Il s'agit d'une bibliothèque Open Source, exécutée par-dessus des frameworks tels que Theano et TensorFlow

Conçue pour être modulaire, rapide et simple d'utilisation, Keras a été créée par l'ingénieur François Chollet de Google. Elle offre une façon simple et intuitive de créer des modèles de Deep Learning.



FIGURE 6.9 – Keras (source :[15])

### 6.4.2.3 TensorFlow

TensorFlow est une bibliothèque open-source très populaire pour le calcul numérique haute performance, développée par l'équipe Google Brain chez Google. Comme son nom l'indique, TensorFlow est un framework qui permet de définir et d'exécuter des calculs impliquant des tenseurs. Il peut entraîner et exécuter des réseaux neuronaux profonds qui peuvent être utilisés pour développer plusieurs applications d'intelligence artificielle. TensorFlow est largement utilisé dans le domaine de la recherche et de l'application de l'apprentissage profond.



FIGURE 6.10 – TensorFlow (source :[16])

### 6.4.2.4 Scipy

SciPy est une bibliothèque très populaire parmi les amateurs de Machine Learning car elle contient différents modules d'optimisation, d'algèbre linéaire, d'intégration et de statistiques. Il existe une différence entre la bibliothèque SciPy et la pile SciPy. SciPy est l'un des packages centraux qui composent la pile SciPy. SciPy est également très utile pour la manipulation d'images.

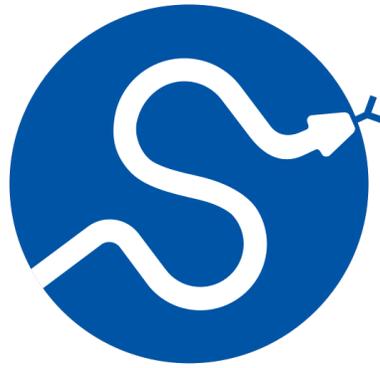


FIGURE 6.11 – SciPy (source :[17])

#### 6.4.2.5 Pandas

Pandas est une bibliothèque Python populaire pour l'analyse de données. Elle n'est pas directement liée à l'apprentissage automatique. Comme nous le savons, les ensembles de données doivent être préparés avant l'entraînement. Dans ce cas, Pandas est très pratique car elle a été spécialement développée pour l'extraction et la préparation des données. Elle fournit des structures de données de haut niveau et une grande variété d'outils pour l'analyse des données. Elle propose de nombreuses méthodes intégrées pour regrouper, combiner et filtrer les données.



FIGURE 6.12 – Pandas (source :[38])

#### 6.4.2.6 NumPy

NumPy est une bibliothèque Python très populaire pour le traitement de grandes matrices et tableaux multidimensionnels, grâce à une vaste collection de fonctions mathématiques de haut niveau. Elle est très utile pour les calculs scientifiques fondamentaux en apprentissage automatique. Elle est particulièrement utile pour l'algèbre linéaire, la transformation de Fourier et les capacités de génération de nombres aléatoires. Des bibliothèques de pointe telles que TensorFlow utilisent NumPy en interne pour la manipulation des tensors.



FIGURE 6.13 – Numpy (source :[18])

#### 6.4.2.7 Scikit-Learn

Scikit-learn est l'une des bibliothèques ML les plus populaires pour les algorithmes ML classiques. Elle est construite sur deux bibliothèques Python de base, à savoir NumPy et SciPy. Scikit-learn prend en charge la plupart des algorithmes d'apprentissage supervisé et non supervisé. Scikit-learn peut également être utilisé pour l'extraction de données et l'analyse de données, ce qui en fait un excellent outil pour ceux qui commencent avec le ML.



FIGURE 6.14 – Scikit Learn (source :[19])

#### 6.4.2.8 Matplotlib

Matplotlib est une bibliothèque Python très populaire pour la visualisation de données. Comme Pandas, elle n'est pas directement liée à l'apprentissage automatique. Elle est particulièrement utile lorsque le programmeur souhaite visualiser les motifs dans les données. Il s'agit d'une bibliothèque de tracé en 2D utilisée pour créer des graphiques et des tracés en 2D. Un module appelé pyplot facilite la création de graphiques, car il offre des fonctionnalités pour contrôler les styles de ligne, les propriétés de police, le formatage des axes, etc. Elle propose différents types de graphiques et de tracés pour la visualisation de données, tels que les histogrammes, les graphiques d'erreur, les graphiques à barres, etc.



FIGURE 6.15 – Matplotlib (source :[20])

## 6.5 Importation des bibliothèques

```
import numpy as np
import pandas as pd
import cv2
from PIL import Image
import scipy

import tensorflow as tf
from tensorflow.keras.applications import *
from tensorflow.keras.optimizers import *
from tensorflow.keras.losses import *
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
from tensorflow.keras.callbacks import *
from tensorflow.keras.preprocessing.image import *
from tensorflow.keras.utils import *
# import pydot

from sklearn.metrics import *
from sklearn.model_selection import *
import tensorflow.keras.backend as K

from tqdm import tqdm, tqdm_notebook
# from colorama import Fore
import json
import matplotlib.pyplot as plt
import seaborn as sns
from glob import glob
from skimage.io import *
%config Completer.use_jedi = False
import time
from sklearn.decomposition import PCA
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import lightgbm as lgb
import xgboost as xgb
from tqdm import tqdm

import os
import shutil
import itertools
import imutils
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from skimage import io
import PIL

import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
from plotly import tools

from keras.preprocessing.image import ImageDataGenerator
from keras.applications.vgg16 import VGG16, preprocess_input
from keras import layers
from keras.models import Model, Sequential
from keras.optimizers import Adam, RMSprop
from keras.callbacks import EarlyStopping
```

FIGURE 6.16 – Importation des Bibliothèques

## 6.6 Métriques d'évaluation des performances

Cette section présente les différentes métriques utilisées pour évaluer la performance de classification des modèles. Il existe plusieurs méthodes d'évaluation pour un classifieur, mais nous avons opté pour une métrique basée sur la matrice de confusion afin de valider les résultats. Les métriques de performance telles que l'exactitude, la précision, le recall, le score F1, le score Cohen Kappa, et le taux d'erreur sont utilisées pour évaluer la capacité prédictive du modèle.

- *Accuracy* : Ou exactitude est le ratio d'images classifiées correctement par le nombre total d'images. Elle est donnée par la formule suivante :

$$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$

- *Precision* : Demontre la capacité du modèle à classifier les images correctement (ie une image labélisée yes, classifiée en yes). Elle est donnée par la formule suivante :

$$\frac{(TP)}{(TP+FP)}$$

- *Recall* : Capacité du modèle à trouver tous les échantillons positifs. il est donné par la formule suivante :

$$\frac{TP}{(TP+FN)}$$

- *F1-Score* : Combine la précision et le recall en calculant la moyenne harmonique.

$$2 * \frac{(Precision*Recall)}{(Precision+Recall)}$$

- *Cohen Kappa Score* : Une statistique utilisée pour mesurer l'accord entre deux évaluateurs

Il est à noter que

- **TP** : Nombre d'images correctement classifiés comme patients atteints d'une tumeur.
- **FP** : Nombre d'images faussement classifiées comme patients atteints d'une tumeur.
- **FN** : Nombre d'images faussement classifiées comme patients non atteints d'une tumeur.
- **TN** : Nombre d'images correctement classifiés comme patients non atteints d'une tumeur.

```

from sklearn import metrics
print('Accuracy score is :', np.round(metrics.accuracy_score(y_test, predictions),4))
print('Precision score is :', np.round(metrics.precision_score(y_test, predictions, average='weighted'),4))
print('Recall score is :', np.round(metrics.recall_score(y_test, predictions, average='weighted'),4))
print('F1 Score is :', np.round(metrics.f1_score(y_test, predictions, average='weighted'),4))
print('ROC AUC Score is :', np.round(metrics.roc_auc_score(y_test, predict_prob, multi_class='ovo', average='weighted'),4))
print('Cohen Kappa Score:', np.round(metrics.cohen_kappa_score(y_test, predictions),4))

print('\t\tClassification Report:\n', metrics.classification_report(y_test, predictions))

```

FIGURE 6.17 – Métriques d'évaluation des performances

## 6.7 Connexion au GoogleColab

```

[ ] from google.colab import drive
    drive.mount('/content/drive')

```

FIGURE 6.18 – Connexion à google colab

## 6.8 Préparation des Données

La préparation des données en deep learning est essentielle pour nettoyer, normaliser, transformer, partitionner et équilibrer les données avant de les utiliser pour entraîner un modèle. Cela garantit la qualité des données, facilite l'apprentissage, améliore les performances du modèle et permet une évaluation précise.

Les images sont cassées en trois classes : une classe pour l'évaluation (test), une classe pour l'entraînement (train), et une classe pour la validation (val).

```

[ ] IMG_PATH = "/content/drive/MyDrive/dataset3"

# split the data by train/val/test
ignored = {"pred"}
# split the data by train/val/test
for CLASS in os.listdir(IMG_PATH):
    if CLASS not in ignored:
        if not CLASS.startswith('.'):
            IMG_NUM = len(os.listdir(IMG_PATH + "/" + CLASS))
            for (n, FILE_NAME) in enumerate(os.listdir(IMG_PATH + "/" + CLASS)):
                img = IMG_PATH + '/' + CLASS + '/' + FILE_NAME
                if n < 300:
                    shutil.copy(img, 'TEST/' + CLASS.upper() + '/' + FILE_NAME)
                elif n < 0.8*IMG_NUM:
                    shutil.copy(img, 'TRAIN/' + CLASS.upper() + '/' + FILE_NAME)
                else:
                    shutil.copy(img, 'VAL/' + CLASS.upper() + '/' + FILE_NAME)

```

FIGURE 6.19 – Préparation des données

## 6.9 Augmentation des données

L'augmentation des données en deep learning est une technique utilisée pour augmenter la taille et la diversité du dataset d'entraînement en générant de nouvelles données à partir des données existantes. Cette technique vise à améliorer la capacité du modèle à généraliser et à réduire le surapprentissage.

L'augmentation des données consiste à appliquer diverses transformations aléatoires et légères aux échantillons du dataset, telles que la rotation, le recadrage, le zoom, le retournement horizontal, le changement de luminosité, etc. Ces transformations préservent la sémantique des données, tout en introduisant des variations qui permettent au modèle de mieux capturer les différentes perspectives, les variations d'éclairage, les déformations, etc.

Nous avons fait une augmentation des données du dataset de la manière suivante :

```
[ ] datagen = ImageDataGenerator(  
    rotation_range=45,      #Random rotation between 0 and 45  
    width_shift_range=0.2,  #% shift  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode='nearest')
```

FIGURE 6.20 – Augmentation

## 6.10 Transfer Learning

Tout d'abord, nous avons récupéré les modèles utilisés grâce à la classe fournie par TensorFlow. Inclutop reçoit false est utilisé pour préciser qu'il ne faut garder que les premières couches de convolutions et exclure les dernières couches denses (FC). Les images en entrée doivent être au format RVB de taille 224x224 pixels.

```
[ ] base_Neural_Net= ResNet50(input_shape=(224,224,3), weights='imagenet', include_top=False)
```

FIGURE 6.21 – Transfer Learning ResNet50

```
[ ] # Load base model  
vgg16_weight_path = '../content/drive/MyDrive/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5'  
base_model = VGG16(  
    weights=vgg16_weight_path,  
    include_top=False,  
    input_shape=IMG_SIZE + (3,) )
```

FIGURE 6.22 – Transfer Learning VGG16

```
base_Neural_Net= VGG19(input_shape=(224,224,3), weights='imagenet', include_top=False)
```

FIGURE 6.23 – Transfer Learning VGG19

## 6.11 Compilation du modèle

```
model.compile(  
    loss='binary_crossentropy',  
    optimizer='adam',  
    metrics=['accuracy', 'AUC']  
)  
  
model.summary()
```

FIGURE 6.24 – Compilation du modele

## 6.12 Entraînement du modèle

```
[ ] EPOCHS = 30  
    es = EarlyStopping(  
        monitor='val_acc',  
        mode='max',  
        patience=6  
    )  
    history = model.fit_generator(  
        train_generator,  
        steps_per_epoch=50,  
        epochs=EPOCHS,  
        validation_data=validation_generator,  
        validation_steps=25,  
        callbacks=[es]  
    )
```

FIGURE 6.25 – Entraînement de Resnet50

## 6.13 Test du modèles

Ci dessous les tableaux englobent les matrices résultant de l'entraînement des modèles sur les deux datasets utilisés.

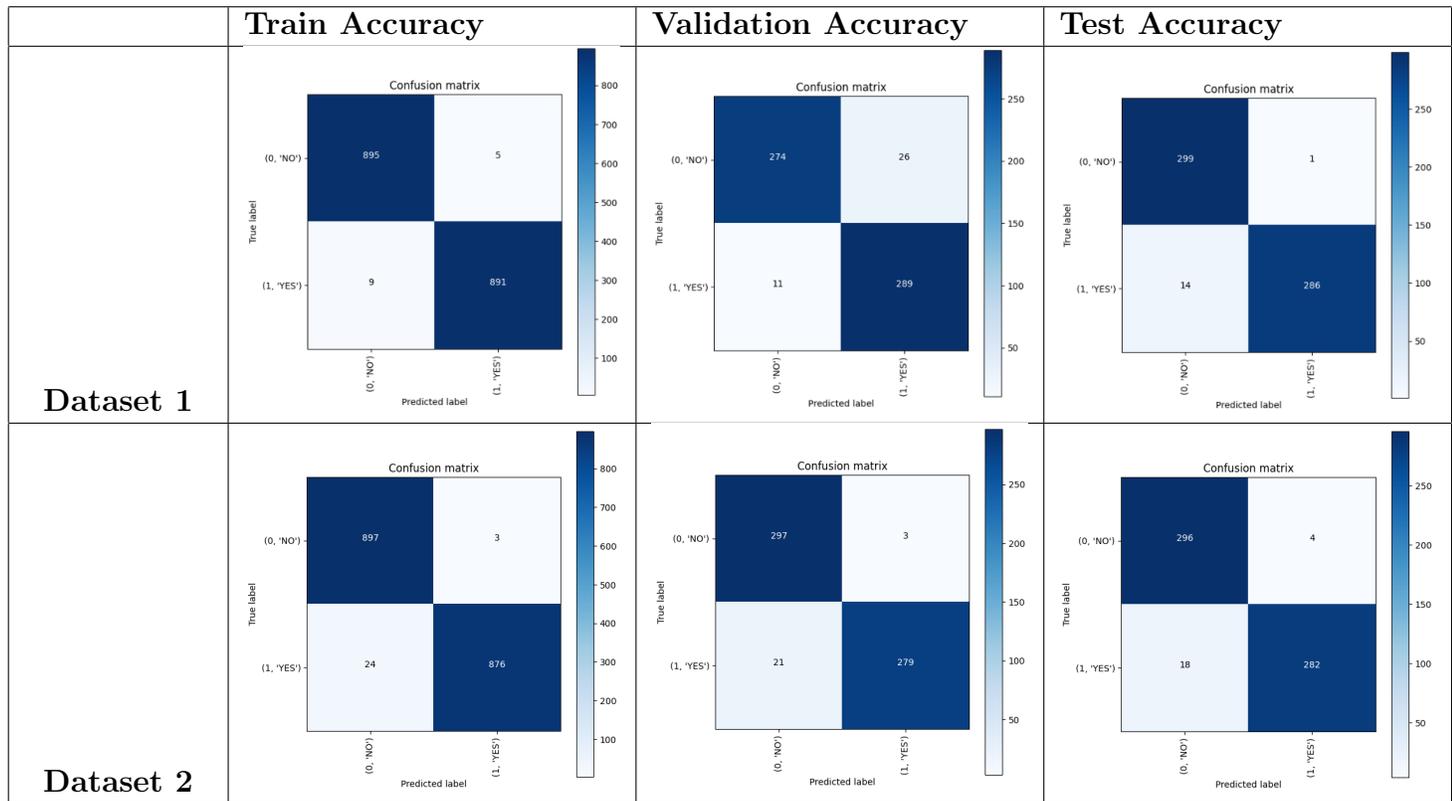


TABLEAU 6.1 – Matrices VGG16

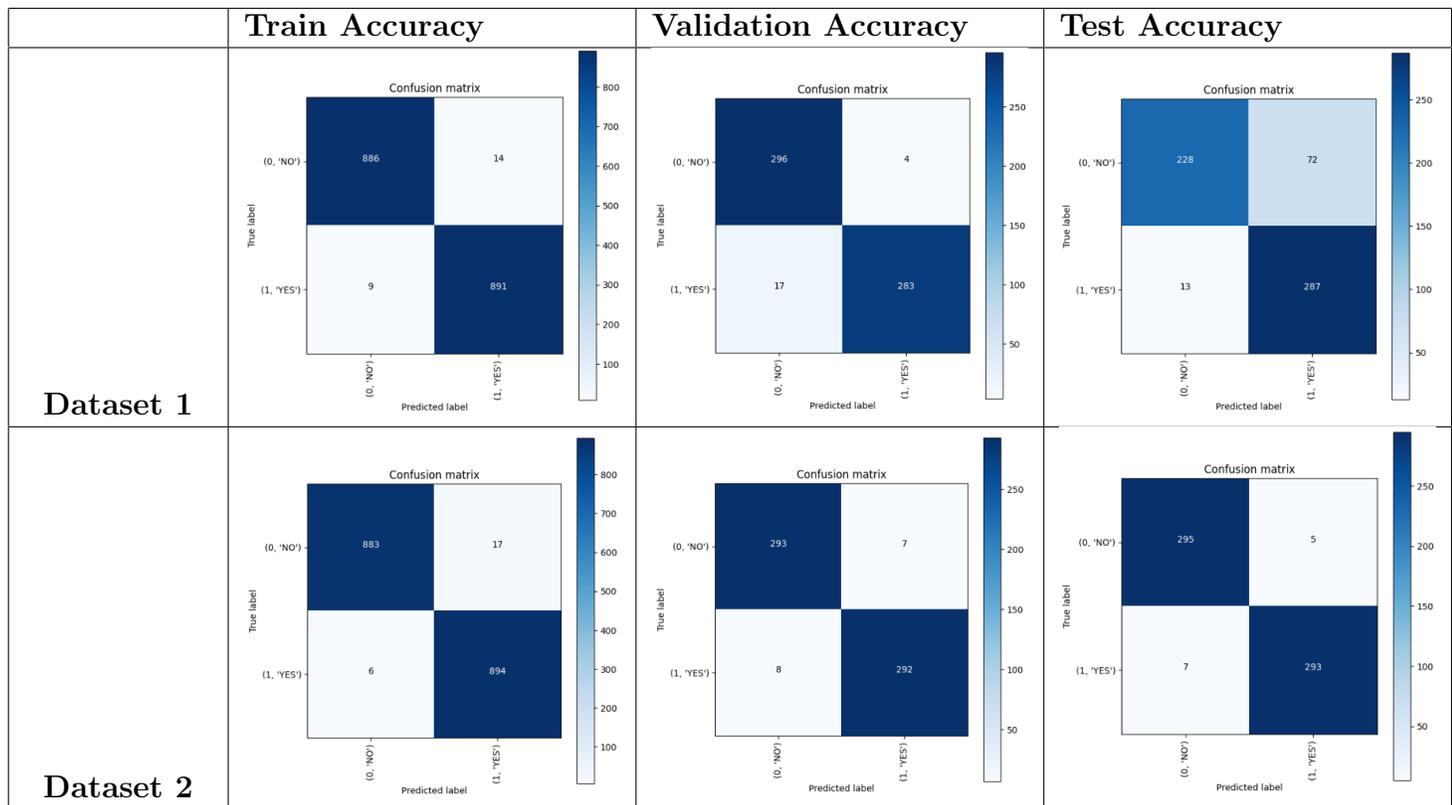


TABLEAU 6.2 – Matrices VGG19

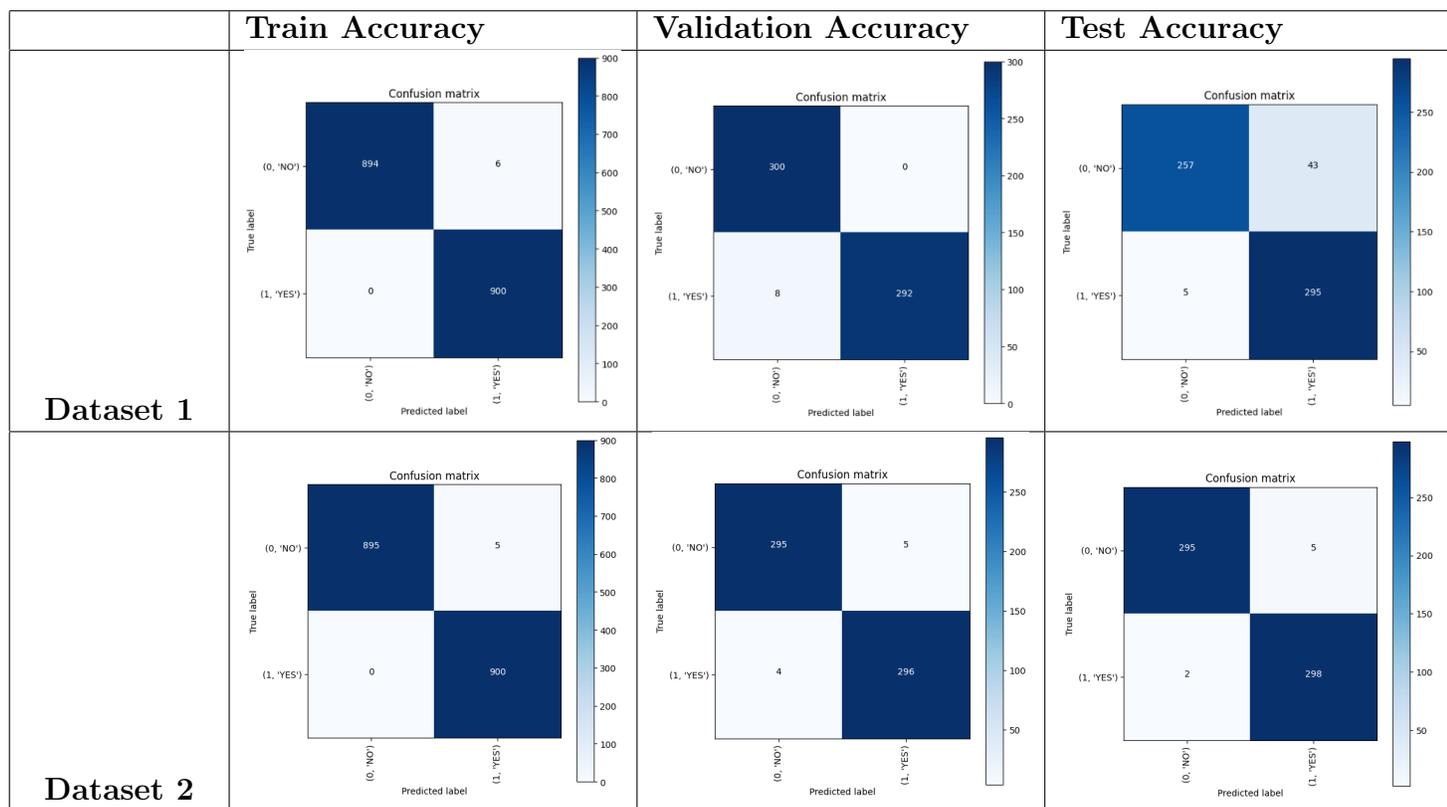


TABLEAU 6.3 – Matrices ResNet 50

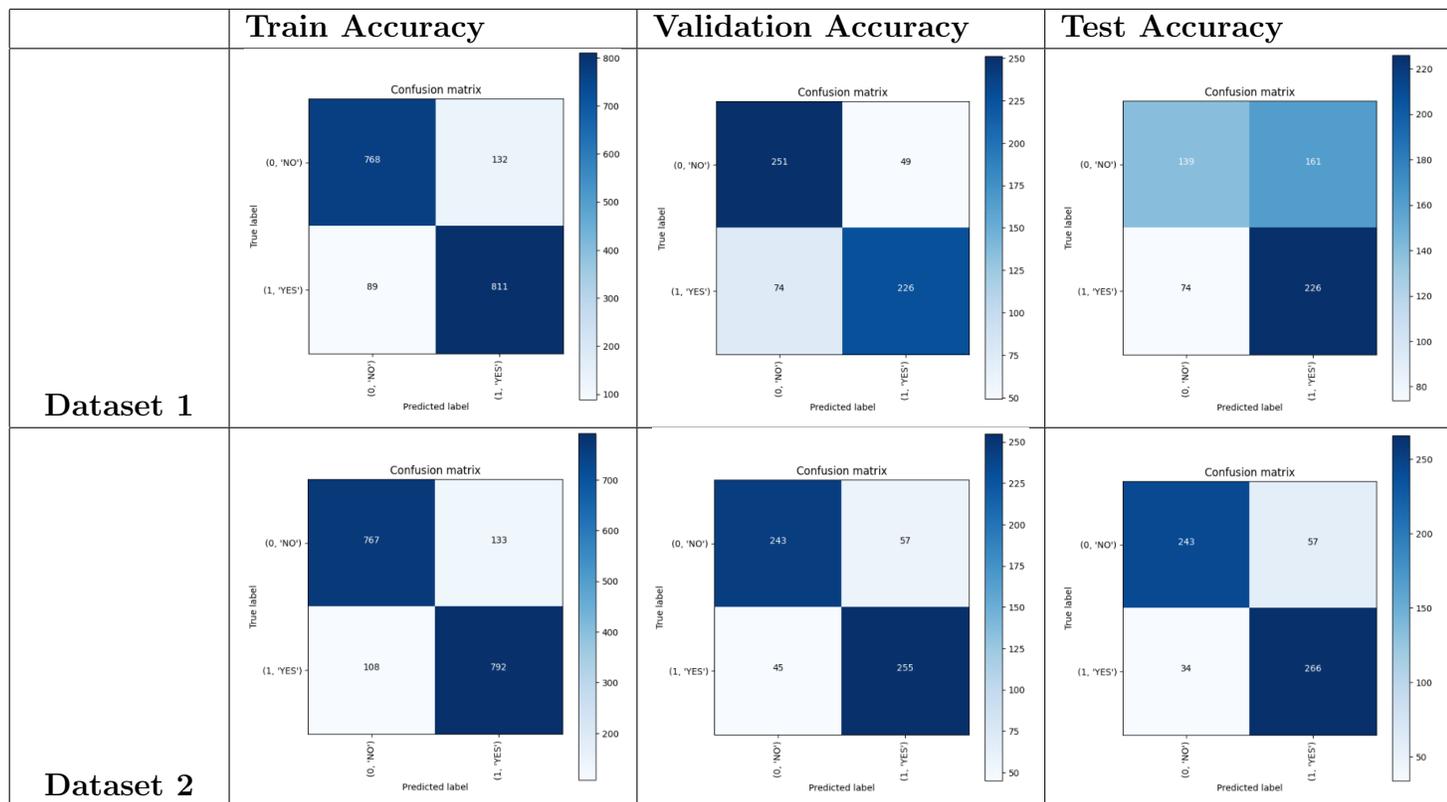


TABLEAU 6.4 – Matrices InceptionV3

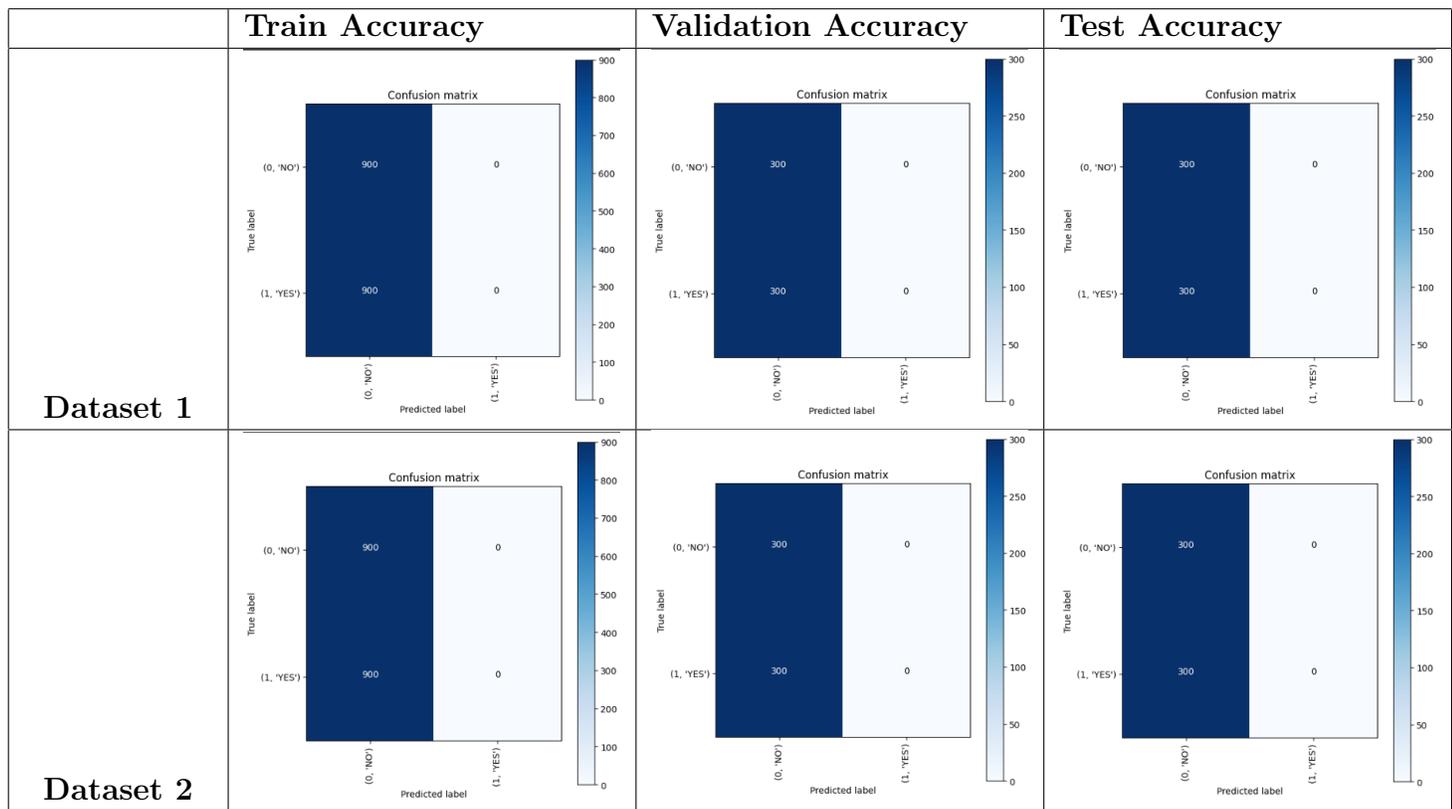


TABLEAU 6.5 – Matrices GoogLeNet

## 6.14 Résultats

### 6.14.1 Dataset 1

#### 6.14.1.1 Résultats des tests

	Train Accuracy	Validation Accuracy	Test Accuracy
VGG16	0,99	0,97	0,94
VGG19	0,99	0,96	0,86
GoogleNet	0,5006	0,5	0,5
InceptionV3	0,88	0,8	0,61
ResNet50	1,00	0,99	0,92

TABLEAU 6.6 – Tableau des tests du dataset1

## 6.14.1.2 Résultats du modèle

	Accuracy	Precision	Recall	F1	Cohen Kappa
<b>VGG16</b>	0,9383	0,9394	0,9383	0,9383	0,8767
<b>VGG19</b>	0,8583	0,8728	0,8583	0,857	0,7167
<b>GoogleNet</b>	0,5	0,25	0,5	0,3333	0
<b>InceptionV3</b>	0,6083	0,6183	0,6083	0,5999	0,2167
<b>ResNet50</b>	0,92	0,9268	0,92	0,9197	0,84

TABLEAU 6.7 – Tableau des résultats du dataset1

## 6.14.2 Dataset 2

## 6.14.2.1 Résultats des tests

	Train Accuracy	Validation Accuracy	Test Accuracy
<b>VGG16</b>	0,98	0,96	0,96
<b>VGG19</b>	0,99	0,97	0,98
<b>GoogleNet</b>	0,5044	0,5	0,5
<b>InceptionV3</b>	0,87	0,83	0,85
<b>ResNet50</b>	1	0,98	0,99

TABLEAU 6.8 – Tableau des tests du dataset 2

## 6.14.2.2 Résultats du modèle

	Accuracy	Precision	Recall	F1	Cohen Kappa
<b>VGG16</b>	0,9633	0,9643	0,9633	0,9633	0,9267
<b>VGG19</b>	0,98	0,98	0,98	0,98	0,96
<b>GoogleNet</b>	0,5	0,25	0,5	0,3333	0
<b>InceptionV3</b>	0,8483	0,8504	0,8483	0,8481	0,6967
<b>ResNet50</b>	0,9883	0,9884	0,9883	0,9883	0,9767

TABLEAU 6.9 – Tableau des résultats du dataset1

## 6.15 Discussion

Nous avons séparé les résultats en deux tableaux distincts pour chacun des deux dataset : Un tableau des tests et un tableau des résultats. Le tableau des tests englobe le train accuracy, la validation accuracy et le test accuracy des modèles. Le second englobe les métriques de l'accuracy, la precision, le recall, le F1 et le Cohen-Kappa.

Dans le dataset 1 le tableau des tests 6.6 nous dévoile l'ordre des modèles comme suit :

- ResNet 50 avec 1, 0.99 et 0.92 pour train accuracy, validation accuracy et test accuracy respectivement
- VGG16 avec 0.99, 0.97 et 0.94
- VGG19 avec 0.99, 0.96 et 0.86
- Inception V3 avec 0.88, 0.8 et 0.61
- GoogLeNet avec 0.5006, 0.5 et 0.5.

Le tableau 6.7 des résultats des modèles nous donne le classement suivant :

- VGG16 avec 0.9383, 0.9394, 0.9383, 0.9383 et 0.8767 pour accuracy, precision, recall, F1 et Cohen-Kappa respectivement.
- ResNet50 avec 0.92, 0.9268, 0.92, 0.9197 et 0.84.
- VGG19 avec 0.8583, 0.8728, 0.8583, 0.857 et 0.7167.
- Inception V3 avec 0.6083, 0.6183, 0.6083, 0.5999 et 0.2167.
- GoogLeNet avec 0.5, 0.25, 0.5, 0.3333 et 0.

Nous notons qu'avec le premier dataset, VGG16 donne de meilleurs résultats lors de la classification des IRM cérébrales en deux classe (yes/no), avec 0.9383, 0.9394, 0.9383, 0.9383 et 0.8767 pour accuracy, precision, recall, F1 et Cohen-Kappa respectivement, surpassant ResNet50 (accuracy de 0.92) et VGG19 (accuracy de 0.8583).

Dans le dataset 2 le tableau des tests 6.8 nous dévoile l'ordre des modèles comme suit :

- ResNet 50 avec 1, 0.98 et 0.99 pour train accuracy, validation accuracy et test accuracy
- VGG19 avec 0.99, 0.97 et 0.98.
- VGG16 avec 0.98, 0.96 et 0.96.
- Inception V3 avec 0.87, 0.83 et 0.85.
- GoogLeNet avec 0.5040, 0.5 et 0.5.

Le tableau 6.9 des résultats des modèles nous donne le classement suivant :

- ResNet50 avec 0.9883, 0.9884, 0.9883, 0.9883 et 0.9767 pour accuracy, precision, recall, F1 et Cohen-Kappa respectivement.
- VGG19 avec 0.98, 0.98, 0.98, 0.98 et 0.96
- VGG16 avec 0.9633, 0.9643, 0.9633, 0.9633 et 0.9267

- Inception V3 avec 0.8483, 0.8504, 0.8483, 0.8481 et 0.6967
- GoogLeNet avec 0.5, 0.25, 0.5, 0.3333 et 0.

Dans le cas du deuxième dataset, ResNet50 donne les meilleurs résultats avec 0.9883, 0.9884, 0.9883 et 0.9767 pour accuracy, precision, recall, F1 et Cohen-Kappa respectivement, surpassant VGG19 (accuracy de 0.99) et VGG16 (accuracy de 0.98).

On note que ces résultats sont en accord avec ceux des articles [60], [55] et [49].

GoogLeNet est le modèle avec les résultats les moins concluants, et cela avec les deux jeux de données, tel que les résultats obtenus dans notre étude sont encore plus faibles que ceux obtenus dans l'article[60] , ce qui en fait le modèle le moins fiable à l'utilisation dans le milieu médical.

En terme de temps, ResNet50 est le plus rapide avec 4 heures d'entraînement pour 30 époques de 50 pas, contrairement à VGG19 qui lui prend plus de 8 heures d'entraînement, faisant de lui le modèle le plus long à entraîner dans notre étude, et pour cela nous avons modifié les hyperparamètres du modèle en diminuant le nombre d'époques et de pas à 20 époques de 15 pas chacune pour atteindre 4 heures d'entraînement.

On observe aussi que la taille du jeu de donnée utilisé a un impact sur le degré d'exactitude (accuracy) et de précision (precision) des modèles utilisés.

Selon les données recollées dans notre travail ainsi que les articles étudiés, on conclut dans notre étude que les modèles VGG16, VGG19 et ResNet50 sont parmi les meilleures architectures à utiliser pour le développement d'une application de classification d'IRM médicale pour la détection de tumeurs cérébrales.

# Conclusion et perspectives

Dans notre travail nous présentons une étude comparative des techniques de deep learning qui permettent la détection de tumeurs cérébrales sur des images d'IRM, en utilisant des méthodes de classification qui reposent sur l'extraction des caractéristiques grâce aux couches des réseaux de neurones convolutifs (CNN).

Notre travail a été structuré comme suit :

- Une présentation des tumeurs et cancer du cerveau. Nous avons défini et expliqué ce qu'est une tumeur cérébrale, ses symptômes, ses causes ainsi que les traitements existants.
- Une présentation détaillée de l'imagerie par résonance magnétique qui est le principal outil de détection des tumeurs du cerveau.
- Nous avons défini ce que sont l'intelligence artificielle, l'apprentissage automatique, l'apprentissage profond et les réseaux de neurones ainsi que leurs caractéristiques et les différents modèles existant.
- Nous avons présenté certains articles fait dans le domaine.
- L'étude comparative des résultats obtenus lors de l'entraînement des modèles choisis.

Après l'étude et l'analyse de plusieurs travaux fais auparavant dans le domaine, nous avons sélectionné les modèles CNN les plus performants et répondant à notre problématique. Nous avons utilisé les cinq modèles préentraîné suivant : VGG16, VGG19, ResNet50, GoogLeNet et Inception V3 sur lesquels nous avons utilisé le Transfer Learning. Nous avons aussi appliqué des méthodes de Data Augmentation sur les deux datasets utilisés afin d'augmenter le nombre de données d'entraînement.

Dans notre travail nous avons comparé les résultats obtenus lors de l'entraînement de ces modèles sur chacun des deux dataset en fonction de caractéristiques précises, principalement l'accuracy, la précision, le recall, le F1 et Cohen-Kappa.

Nous avons déduit que les modèles de la famille VGG (16 et 19) ainsi que ResNet50 sont parmi les meilleures options à appliquer dans le milieu médical.

Plusieurs perspectives peuvent être envisageables suite à ce mémoire, parmi elles :

- Amélioration des modèles de CNN pour la classification.
- Créer une application se basant sur l'un des modèles.

- Test sur de plus grands jeu de données locaux.
- Tester les modèles sur des images d'IRM 3D

# Bibliographie

- [1] Comment le cancer se forme, se développe et se propage? <https://cancer.ca/fr/cancer-information/what-is-cancer/how-cancer-starts-grows-and-spreads>.
- [2] Symptômes de la tumeur au cerveau ou à la moelle épinière. <https://cancer.ca/fr/cancer-information/cancer-types/brain-and-spinal-cord/signs-and-symptoms>.
- [3] Brain tumor facts. <https://braintumor.org/brain-tumors/about-brain-tumors/brain-tumor-facts/>, (Consulté le 01 Juin 2023).
- [4] Cancer environnement. <https://www.cancer-environnement.fr/fiches/cancers/cancer-du-cerveau/>, (Consulté le 01 Juin 2023).
- [5] Cancers du cerveau : les symptômes et le diagnostic. <https://www.fondation-arc.org/cancer/cancer-cerveau/symptomes-diagnostic-cancer#diagnostic>, (Consulté le 01 Juin 2023).
- [6] Imagerie par résonance magnétique. <https://www.msmanuals.com/fr/accueil/sujets-particuliers/examens-d-imagerie-courants/imagerie-par-r%C3%A9sonance-magn%C3%A9tique-irm>, (Consulté le 01 Juin 2023).
- [7] Les causes des tumeurs du cerveau. <https://www.vidal.fr/maladies/cancers/tumeurs-cerveau/causes.html>, (Consulté le 01 Juin 2023).
- [8] Tumeurs du système nerveux central. <https://www.msmanuals.com/fr/accueil/troubles-du-cerveau,-de-la-moelle-%C3%A9pini%C3%A8re-et-des-nerfs/tumeurs-du-syst%C3%A8me-nerveux/pr%C3%A9sentation-des-tumeurs-du-syst%C3%A8me-nerveux>, (Consulté le 01 Juin 2023).
- [9] Types de cancer. <https://www.cancer.be/le-cancer/jeunes-et-cancer/les-cancers/les-types-de-cancers>, (Consulté le 01 Juin 2023).
- [10] Types de tumeurs. <https://www.vidal.fr/maladies/cancers/tumeurs-cerveau.html>, (Consulté le 01 Juin 2023).
- [11] Classez et segmentez des données visuelles. <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn>, (Consulté le 02 Juin 2023).

- [12] Le dropout c'est quoi? <https://inside-machinelearning.com/le-dropout-cest-quoi-deep-learning-explication-rapide/>, (Consulté le 02 Juin 2023).
- [13] <https://www.python.org/community/logos/>, (Consulté le 03 dJuin 2023).
- [14] <https://logowik.com/google-colaboratory-logo-vector-svg-pdf-ai-eps-cdr-free-download-1.html>, (Consulté le 03 dJuin 2023).
- [15] <https://keras.io/>, (Consulté le 03 dJuin 2023).
- [16] [https://www.tensorflow.org/extras/tensorflow\\_brand\\_guidelines.pdf](https://www.tensorflow.org/extras/tensorflow_brand_guidelines.pdf), (Consulté le 03 dJuin 2023).
- [17] <https://github.com/scipy/scipy.org/blob/main/static/images/logo.svg>, (Consulté le 03 dJuin 2023).
- [18] <https://github.com/numpy/numpy/tree/main/branding/logo>, (Consulté le 03 dJuin 2023).
- [19] <https://scikit-learn.org/stable/about.html>, (Consulté le 03 dJuin 2023).
- [20] <https://matplotlib.org/stable/gallery/misc/logos2.html>, (Consulté le 03 dJuin 2023).
- [21] Empêcher le surajustement et les données déséquilibrées avec le machine learning automatisé. <https://learn.microsoft.com/fr-fr/azure/machine-learning/concept-manage-ml-pitfalls?view=azureml-api-2#identify-overfitting>, (Consulté le 03 dJuin 2023).
- [22] Overfitting : qu'est-ce que c'est? comment l'éviter? <https://datascientest.com/overfitting-tout-savoir>, (Consulté le 03 dJuin 2023).
- [23] Apprentissage supervisé et non supervisé : quelles différences? <https://experiences.microsoft.fr/articles/intelligence-artificielle/apprentissage-supervise-et-non-supervise-quelles-differences/>, (Consulté le 03 Juin 2023).
- [24] Différence entre scanner et irm. <https://www.radiologieparisouest.com/irm/difference-scanner-irm/>, (Consulté le 03 Juin 2023).
- [25] Intelligence artificielle, de quoi parle-t-on? <https://www.cnil.fr/fr/intelligence-artificielle/intelligence-artificielle-de-quoi-parle-t-on>, (Consulté le 03 Juin 2023).
- [26] Intelligence artificielle et santé : Des algorithmes au service de la médecine. <https://www.inserm.fr/dossier/intelligence-artificielle-et-sante/>, (Consulté le 03 Juin 2023).
- [27] Les différents types d'imagerie. <https://www.lesechos.fr/2010/04/une-technique-issu-de-lastronomie-pour-mieux-explorer-la-retine-421341>, (Consulté le 03 Juin 2023).

- [28] L'imagerie médicale. <https://www.cea.fr/comprendre/Pages/sante-sciences-du-vivant/essentiel-sur-imagerie-medicale.aspx>, (Consulté le 03 Juin 2023).
- [29] L'imagerie par résonance magnétique (irm). <https://www.e-cancer.fr/Patients-et-proches/Se-faire-soigner/Traitements/Examens/IRM-et-cancer>, (Consulté le 03 Juin 2023).
- [30] L'irm. <https://www.arcagy.org/infocancer/localisations/autres-types-de-cancers/tumeurs-cerebrales/symptomes-et-diagnostic/le-bilan-initial.html/>, (Consulté le 03 Juin 2023).
- [31] L'irm cérébrale. <https://imagix.biron.com/fr/resonance-magnetique/irm-cerebrale/>, (Consulté le 03 Juin 2023).
- [32] Présentation des examens d'imagerie. <https://www.msmanuals.com/fr/accueil/sujets-particuliers/examens-d-imagerie-courants/pr%C3%A9sentation-des-examens-d-imagerie>, (Consulté le 03 Juin 2023).
- [33] Qu'est-ce que le deep learning? <https://www.intelligence-artificielle-school.com/deep-learning/>, (Consulté le 03 Juin 2023).
- [34] Qu'est-ce que le deep learning? <https://dataanalyticspost.com/Lexique/reseau-de-neurones/>, (Consulté le 03 Juin 2023).
- [35] Transfer learning : Qu'est-ce que c'est? <https://datascientest.com/Transfer-Learning>, (Consulté le 03 Juin 2023).
- [36] Cancer environnement. <https://blent.ai/blog/a/rnn-on-vous-explique-tout>, (Consulté le 04 Juin 2023).
- [37] Cancer environnement. <https://blent.ai/blog/a/transformers-deep-learning>, (Consulté le 04 Juin 2023).
- [38] Pandas. <https://pandas.pydata.org/>, (Consulté le 30 Mai 2023).
- [39] Fonction d'activation : Comment ça marche? une explication simple. <https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/>, (Consulté le 31 Mai 2023).
- [40] Nada Belaidi. L'apprentissage supervisé : définition et exemples. <https://blent.ai/blog/a/apprentissage-supervise-definition>, (Consulté le 03 Juin 2023).
- [41] Navoneel Chakrabarty. Brain mri images for brain tumor detection. <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>, 2019.
- [42] Comment ça marche? CNN. <https://blent.ai/blog/a/cnn-comment-ca-marche>, (Consulté le 31 Mai 2023).
- [43] def me(x). Brain tumor detectionv1.0 cnn vgg16. <https://www.kaggle.com/code/ruslankl/brain-tumor-detection-v1-0-cnn-vgg-16>, 2019.

- [44] A. Djouani. Une approche deep learning pour le diagnostic préalable de la tuberculose, June 2022.
- [45] A. LUANGRATH T. SEDE. L. ZUGAJ H. HAIDAR-AHMAD, M. A. HAMID SALIH. Intelligence artificielle et imagerie médicale, Janvier 2022.
- [46] Ahmed Hamada. Br35h : Brain tumor detection 2020. <https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection>, 2020.
- [47] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society.
- [48] INCa. *LES TUMEURS DU CERVEAU*. e-cancer.fr, 1st edition, 2010.
- [49] Hassan Ali Khan, Wu Jue, Muhammad Mushtaq, and Muhammad Umer Mushtaq. Brain tumor classification in mri image using convolutional neural network. *Mathematical Biosciences and Engineering*, 17(5) :6203–6216, 2020.
- [50] Philippe Meyer. *Vers la modélisation des traitements de radiothérapie : Radiothérapie externe par Tomothérapie Apprentissage profond en Radiotherapie*. PhD thesis, Université de Strasbourg.
- [51] C. Otmani. Détection de tumeur cérébrale avec l’apprentissage profond, June 2022.
- [52] Abhranta Panigrahi. Brain tumor detection inceptionv3. <https://www.kaggle.com/code/abhranta/brain-tumor-detection-inceptionv3>, 2021.
- [53] Abhranta Panigrahi. Brain tumor detection resnet50. <https://www.kaggle.com/code/abhranta/brain-tumor-detection-resnet50>, 2021.
- [54] Abhranta Panigrahi. Brain tumor detection vgg19. <https://www.kaggle.com/code/abhranta/brain-tumor-detection-vgg19/notebook>, 2021.
- [55] Venkatesan Rajinikanth, Alex Noel Joseph Raj, Krishnan Palani Thanaraj, and Ganesh R. Naik. A customized vgg19 network with concatenation of deep and handcrafted features for brain tumor detection. *Applied Sciences*, 10(10), 2020.
- [56] PATIPAN RATTANAWIN. Brain tumor mri classification : Googlenet. <https://www.kaggle.com/code/patipanrattanawin/brain-tumor-mri-classification-googlenet>, 2022.
- [57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.
- [58] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society.
- [59] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2016.

- 
- [60] Ahmet Çinar and Muhammed Yildirim. Detection of tumors on brain mri images using the hybrid convolutional neural network architecture. *Medical Hypotheses*, 139 :109684, 2020.

## RÉSUMÉ

Au cours des dernières années, de plus en plus de recherches ont vu le jour avec pour objectif l'emploi de l'Intelligence artificielle dans les milieux médicaux, et plus spécifiquement, une volonté des chercheurs à trouver le meilleur moyen de l'appliquer au diagnostic de maladies avec une haute précision et à temps record. Pour cela les chercheurs se sont tournés vers les méthodes de deep learning, plus précisément les réseaux de neurones convolutifs (CNN) pour atteindre leur objectif.

Dans notre travail, nous avons voulu dresser une étude comparative des cinq modèles de CNN suivants : ResNet50, VGG16, VGG19, InceptionV3 et GoogLeNet, reconnus pour leur efficacité dans la classification d'images IRM, plus précisément pour le diagnostic de tumeurs cérébrales

**Mots clés :** Tumeur cérébrale; Intelligence artificielle; Deep Learning; CNN; IRM; Détection; Classification; ResNet50; VGG16; VGG19; InceptionV3; GoogLeNet; Etude Comparative

## ABSTRACT

In recent years, more and more research has emerged with the objective of using Artificial Intelligence in medical settings, and more specifically, a desire by researchers to find the best way to apply it to diagnosis of diseases with high precision and in record time. For this, the researchers turned to deep learning methods, more specifically convolutional neural networks (CNN) to achieve their objective.

In our work, we wanted to draw up a comparative study of the following five CNN models : ResNet50, VGG16, VGG19, InceptionV3 and GoogLeNet, recognized for their efficiency in the classification of MRI images, more specifically for the diagnosis of brain tumors.

**Key words :** Brain Tumor; Artificial Intelligence; Deep Learning; CNN; MRI; Detection; Classification; ResNet50; VGG16; VGG19; InceptionV3; GoogLeNet; Comparative Study