REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université 20 Aout 1955 Skikda

Faculté de science

Département de l'informatique

Mémoire

En vue de l'obtention du diplôme de master

Filière : Informatique

Spécialité : Génie logiciel avancé et applications

Intégration de Modèles IA pour la Classification d'IRM Cérébrales

---

Présenté par
KADOUM Adem

Encadrant : PR. MAZOUZI Smaine

Année universitaire 2024-2025

Remerciements

Je voudrais exprimer ma profonde gratitude à ALLAH pour m'avoir donné la force et la détermination nécessaires pour mener à bien ce mémoire

Je tiens également à remercier mon encadrant, ***Pr. MAZOUZI SMAINE****, pour son soutien inestimable, ses conseils éclairés et sa disponibilité constante tout au long de ce projet. Son expertise a été cruciale pour orienter mes recherches et structurer mes idées.*

J'adresse aussi mes sincères remerciements aux membres du jury.

J'exprime enfin ma gratitude à mes enseignants ainsi qu'à mes amis pour leur soutien indéfectible tout au long de ce parcours académique.

Dédicace

Je dédie ce travail à mes parents, pour leur amour inconditionnel, leur soutien et leurs prières, sans lesquels rien de tout cela n'aurait été possible.
À mes enseignants, qui m'ont transmis leur savoir et accompagné tout au long de mon parcours universitaire.
À mes amis et collègues, pour leur encouragement et leur présence bienveillante.
Enfin, à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

Merci à vous.

ملخص

أصبح التصنيف الآلي للصور الطبية في الوقت الحاضر وسيلةً أساسيةً لمساعدة الأطباء وأخصائيي الأشعة في تفسير صور الأشعة والتوصل إلى تشخيصات موثوقة، وفي وقت قصير. في أطروحة الماستر هذه، اهتممتُ بالتصنيف المُشرف لصور الرنين المغناطيسي**(MRIs)** ، والذي يهدف إلى تشخيص وجود أو عدم وجود أورام دماغية. أجرينا تجارب على نموذجين للتعلم الآلي، يتميزان بأدائهما وكفاءتهما. وهما الشبكات العصبية التلافيفية **(CNN)** وأشجار القرار **.(DT)** كما أجرينا تجارب على دمج هذين النموذجين لتحسين نتائج التصنيف**.**

الكلمات المفتاحية: التصنيف المُشرف، **CNN**، **DT**، التصوير بالرنين المغناطيسي، تضمين النموذج**.**

Résumé

La classification automatique d'images médicales est devenue de nos jours un outlil incontournable d'aide aux médecins et aux radiologues pour l'interprétation des radiographies et pour établir des diagnostics fiables, et aussi en un temps court. Dans ce mémoire de master, je me suis intéressé à la classification supervisée des IRM (Images à résonnance magnétiques), dont le but de poser des diagnostics d'existence ou non de tumeurs cérébrales. Nous avons expérimenté deux modèles d'apprentissage automatique, connus pour leur performance et leur efficacité. Il s'agit des réseaux de neurones convolutifs (CNN :Convolutional Neural Network) et des arbres de décision (DT : Decision Tree). Nous avons également expérimenté l'intégration de ces deux modèles afin d'améliorer les résultats de classification.

Mots-clés :

Classification supervisée, CNN, DT, IRM, Intégration de modèles.

Abstract

Automatic classification of medical images has become an essential tool for physicians and radiologists in interpreting MRI (Magnetic Resonnance Images) and making reliable, and time-efficient diagnoses. In this master's thesis, I focused on the supervised classification of MRI, with the goal of diagnosing the presence or absence of brain tumors. We experimented two machine learning models known for their performance and efficiency: convolutional neural networks (CNNs) and decision trees (DTs). We also experimented the integration of these two models aiming to improve classification results.

Keywords:

Supervised classification, CNN, DT, MRI, Model integration.

Table des matières

Liste des Figures 3
Liste des tableaux 6
Introduction générale 7
Chapitre 01 : Imagerie Médicale 9
1-Introduction 10
2-Définition d'image 10
2-1- Image analogique 10
2-1- Image numérique 11
3- Les types des images numériques 14
3-1- Images Vectorielles 14
3-2- Images matricielles (bitmap) 15
4-L'imagerie Médicale 16
4-1-Techniques de l'imagerie médicale 16
5-Segmentation d'image 22
5-1- Définition de la segmentation 22
5-2-Fonctionnement de la segmentation d'image 23
5-3- Les techniques de segmentation d'image 24
5-4-Difficultés de la segmentation des images médicales 28
6-Conclusion 29
Chapitre 02 : Apprentissage automatique 30
1-Introduction 31
2-Intelligence Artificielle 31
3-Apprentissage automatique « machine-learning » 31
3-1-Les types d'apprentissage automatique 32
4-Quelques Algorithmes d'apprentissage automatique 34
4-1-Les machines à vecteur de support SVM 34
4-2- L'Arbre de décision 35
4-3- Forêt aléatoire 36
4-4-Le clustering K-moyennes mobiles 37
5-Apprentissage profond (Deep Learning) 37
6-Réseaux De Neurones (RNs) 38
6-1- Neurone biologique 38
6-2- Neurone artificiel 39
6-3- Fonctions d'activation 40
7-Types de réseaux de neurones 43

7-1- réseau de neurones récurrents(RNNs) 43
7-2- Réseaux de Neurones Convolutifs (CNN)....44
8-Conclusion....47
Chapitre 03 : Modèles d'apprentissage automatique en classification d'IRM avec intégration ..48
1-Introduction 49
2-Aperçu du système 49
2-1- Principe du fonctionnement 49
2-2-Organigramme 50
3-Modèles IA utilisées....50
3-1- Les réseaux de neurones convolutifs (CNN)....50
3-2-Les arbres de décision....60
- **Construction de l'arbre** 60
- **Avantages et inconvénients des arbres de décision** 61

4-La stratégie d'intégration 64
4-1-Intégration basée sur la précision (Vote pondéré) 64
4-2-Intégration basée sur l'historique (par classe) 64
Cette méthode d'intégration repose sur l'exploitation des performances historiques détaillées de chaque modèle, non pas globalement, mais par classe (par exemple : tumeur et normal).....64
Contrairement aux approches classiques qui utilisent une précision globale, cette méthode analyse les taux de précision obtenus par chaque modèle pour chaque classe spécifiquement. 64
Pour chaque classe, on identifie quel modèle est historiquement le plus fiable pour la reconnaître correctement. Ainsi, lorsqu'une nouvelle image est testée, et que les deux modèles donnent des prédictions différentes, la décision finale est guidée par le modèle ayant historiquement mieux performé sur la classe prédite. 64
5-Implémentation et tests....65
5-1-le langage python 65
5-2-Google colab 66
5-3-Codes de modeles....68
5-4-Resultats 79
5-5-Discussion....82
- **Performances du modèle CNN** 82
- **Performances du modèle Arbre de Décision (Decision Tree)** 83
- **Performances de l'intégration basée sur la précision (Precision-Based)**....83
- **Performances de l'intégration basée sur l'historique (History-Based)** 84

6-Conclusion....84
Conclusion générale....79

Liste des Figures

Figure 1.1 : Exemple image numérique……....11

Figure 1.2 : exemple d'ensemble de pixels……...12

Figure 1.3 **:** Image vectorielle……...15

Figure 1.4 : Image matricielle……...16

Figure 1.5 : Exemple d'image de rayons X……...17

Figure 1.6 : Exemple d'image de tomodensitométrie…….…….....18

Figure 1.7 : Exemple d'IRM…….19

Figure 1.8 : Principe de la formation d'IRM……....20

Figure 1.9 : Les 3 vues IRM selon les 3 principaux axes…….20

Figure 1.10 : Exemple d'échographie……...…21

Figure 1.11 : Exemple d'image Scintigraphies (Médecine Nucléaire)……...……..22

Figure 2.1 : L'apprentissage supervisé……..32

Figure 2.2 : Apprentissage non supervisé…….33

Figure 2.3 : Apprentissage semi-supervisé……...33

Figure 2.4 : Apprentissage par renforcement……34

Figure 2.5 : Exemple simple représentant la classification par SVM……...35

Figure 2.6 : Un arbre de décision pour un apprentissage non linéaire……..36

Figure 2.7 : Un exemple de la structure d'une forêt aléatoire prenant en compte plusieurs arbres de décision……..37

Figure 2.8 : Neurone biologique……...38

Figure 2.9 : Neurone artificielle……39

Figure 2.10 : Couches de réseaux de neurones…….40

Figure 2.11 : Fonctions d'activation couramment utilisées……...41

Figure 2.12 : GELU……..43

Figure 2.13 : Structure d'un réseau de neurones récurrents simple (RNN) et RNN déplié…..44

Figure 2.14 : Architecture typique d'un CNN……...45

Figure 2.15 : Explication de l'opération de convolution……...46

Figure 2.16 : Exemple de couche de regroupement……..47

Figure 3.1 : Organigramme du système……..……..50

Figure 3.2 : Une architecture typique d'un CNN……..51

Figure 3.3 : Exemple d'une opération de convolution……..53

Figure 3.4: Exemple d'opération de convolution avec padding, un tenseur d'entrée de taille 5×5, noyau de 3×3 et un stride de 1……..54

Figure 3.5 : Types d'opération de pooling……55

Figure 3.6 : schéma d'un réseau à convolution avec la couche Flattening……...…55

Figure 3.7 : exemple architecture d'alexnet……..57

Figure 3.8 **:** Logo Phyton……..65

Figure 3.9 : Logo Kears……65

Figure 3.10 : Logo Tensorflow……66

Figure 3.11 : Logo OpenCV…….67

Figure 3.12 : L`environnement Google Colaboratory……..69

Figure 3.13 : Les 2 classes dans le data set……...70

Figure 3.14 **:** code prétraitement des images……...71

Figure 3.15 : code normalisation des images……...71

Figure 3.16 : code division des images ……...71

Figure 3.17 : code Entrainement du modèle DT……..72

Figure 3.18 : code Évaluation du modèle DT……..72

Figure 3.19 : code préparation des données…….73

Figure 3.20 : code de la Division des images pour CNN……73

Figure 3.21 : code préparation des images pour CNN……...74

Figure 3.22 : code conception du modèle CNN……74

Figure 3.23 : code Compilation du modèle CNN……75

Figure 3.24 : code Entraînement du modèle AlexNet……..76

Figure 3.25 : code Évaluation des performances du modèle CNN……...77

Figure 3.26 : code Méthode d'intégration basée précision……..77

Figure 3.27 : code Évaluation du modèle intégré basé précision……..77

Figure 3.28 : code Méthode d'intégration basé historique……78

Figure 3.29 : code Évaluation du modèle intégré basé historique……...78

Liste des tableaux

Tableau 01 : Résultat du modèle DT……...79

Tableau 02 : Résultat du modèle CNN……80

Tableau 03 : Résultat de l'intégration basé précision (vote pondéré)…….81

Tableau 04 : Résultat de l'intégration basé historique (par classe)…….82

Introduction générale

Le diagnostic automatique basé sur l'imagerie médical a connu un fort progrès depuis plusieurs décennies. Ce progrès est dû à deux facteurs principaux : 1) l'amélioration des technologies de production d'images médicales, conduisant à des résolutions élevées permettant de ressortir des détails fins permettant aux médecins et au radiologues de poser des diagnostics fiables. 2) L'évolution importante des modèles d'intelligence artificielle, notamment d'apprentissage automatique, avec implémentation sur des architectures matérielles dédiées, telles que les GPU.

Dans ce mémoire de master, je me suis penché sur l'étude de quelques modèles d'apprentissage automatique pour la classification d'IRM (Images à Résonnance Magnétique), dont le but est de déceler la présence ou non de tumeurs cérébrales dans les clichées d'IRM analysés. Je me suis intéressé également à l'intégration de ces modèles dans le but d'améliorer les résultats de classification de l'ensemble des modèles. Contraint par la limite du temps qui nous est accordé pour la préparation du mémoire de master, je me suis restreint à deux modèles, connus pour leur notoriété dans le domaine de l'apprentissage automatique. D'une part les arbres de décision (DT : Decision Trees), connues pour leur performance en classification supervisée, avec un moindre effort d'apprentissage, en temps et en taille des datasets d'entrainement. D'autre part, les réseaux de neurones convolutifs (CNN : Convolutional Neural Networks), assurant des performances avérées avec des données structurées en 2D et 3D, mais nécessitant des données d'entrainement et un temps d'apprentissage, les deux importants.

Ce mémoire est structuré comme suit :

Le chapitre 1 est consacré à l'imagerie médicale, dans lequel, je présente ce domaine, les différentes modalités en imagerie médicale, ainsi que les principaux traitements qui leur relèvent.

Au chapitre 2, je survole le domaine de l'apprentissage automatique, dont je présenterai les principaux modèles, selon les différentes approches de classification : supervisée et non supervisée.

Le chapitre 3 présente ma contribution qui consiste à l'utilisation de deux modèles d'apprentissage automatique supervisés pour la classification d'IRM cérébrales. Après la présentation des deux modèles, nous montrant les approches d'intégration auxquelles j'ai opté. Je présenterai également dans ce chapitre l'implémentation et l'expérimentation des modèles étudiés en utilisant le langage Python sous l'environnement de développement Google Colab. Je présenterai à la fin du chapitre les résultats obtenus ainsi qu'une discussion de ces derniers.

Je termine ce mémoire par une **conclusion générale**, où je reviendrai sur ce qui a été réalisé, tout en soulignant les potentielles perspectives à cette étude.

Chapitre 01 : Imagerie Médicale

1-Introduction

Dans le domaine de l'informatique appliquée à la santé, le traitement d'images médicales représente un enjeu fondamental pour l'analyse, le diagnostic et la prise en charge des pathologies. Les images numériques, issues de techniques avancées comme l'IRM, le scanner ou l'échographie, contiennent une grande richesse d'informations visuelles permettant d'examiner les structures internes du corps humain sans intervention invasive. Ce chapitre présente les fondements essentiels liés à la nature des images numériques, leurs caractéristiques, ainsi que les principales techniques d'imagerie médicale. Une attention particulière est ensuite portée à la segmentation d'images, une étape cruciale permettant d'extraire automatiquement des régions d'intérêt. À travers l'étude des méthodes classiques et modernes de segmentation, ce chapitre pose les bases indispensables pour comprendre les approches utilisées dans les systèmes intelligents de traitement d'images médicales.

2-Définition d'image

Une image est une représentation visuelle ou mentale d'un objet, d'un être vivant ou d'un concept. Elle peut être naturelle (comme une ombre ou un reflet) ou artificielle (comme une photographie, une peinture ou une sculpture) [1].

Ou :

L'image est une représentation d'une personne ou d'un objet par la peinture, le dessin, la photographie, d'un objet par la peinture, le dessin, la photographie, le film, etc.…. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain.

2-1- Image analogique

• Elle peut être décrite sous la forme d'une fonction I (x, y) analogique continue, définie dans un domaine borné, tel que

x et y sont les coordonnées spatiales d'un point de l'image

I est une fonction d'intensité lumineuse et de couleur.

Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation.

2-1- Image numérique

Plan discret dérivé d'une image analogique après numérisation (digitization).

Matrice dont la valeur de chaque élément représente une intensité discrète de la lumière [2].

Une image numérique est le résultat de la numérisation d'une image analogique en deux étapes [2].

Échantillonnage spatial : discrétisation des coordonnées de l'image réelle ;

Quantification des luminances : discrétisation des intensités de l'image réelle.

Une image numérique est composée d'un ensemble fini d'éléments appelés picture element, ou pixels (voxels en 3D)

La numérisation d'une image est la conversion de celleci de son état analogique en une image numérique représentée par f(x,y) une matrice bidimensionnelle de valeurs numériques f(x,y),

où :

• x,y : coordonnées cartésiennes d'un point de l'image.

• f(x, y) : niveau d'intensité.

• La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur.

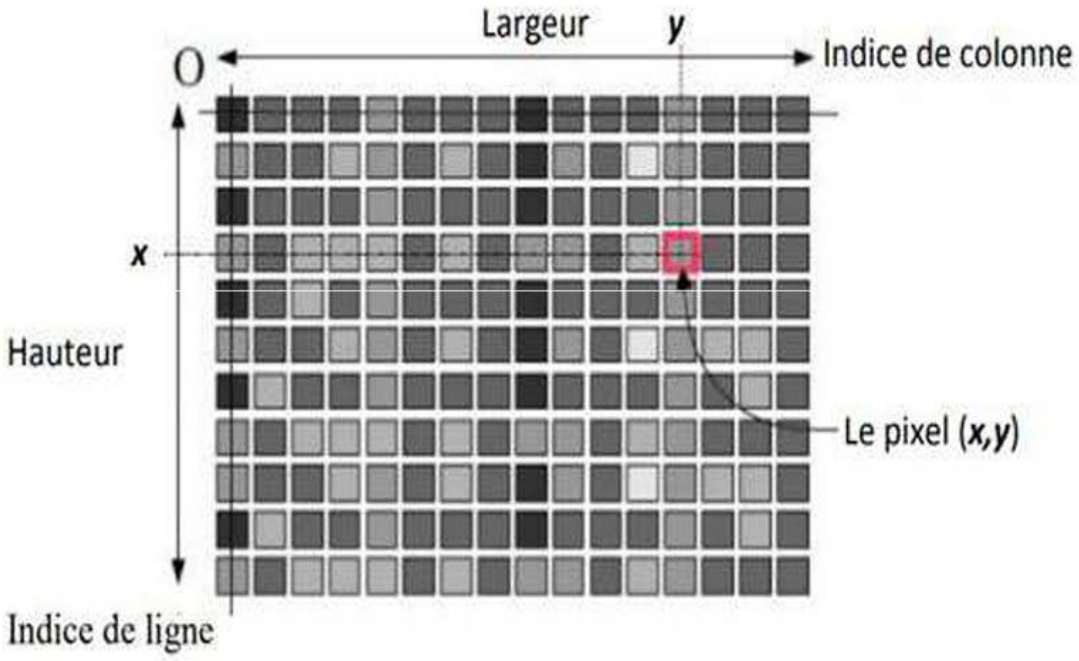

Figure 1.1 : Exemple image numérique

2-3-Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants [3] :

1- **Pixel** : Contraction de « Picture Eléments ": éléments d'image

• C'est le plus petit point de l'image, c'est une valeur numérique représentative des intensités lumineuses.

• Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels sur l'image [3].

La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :

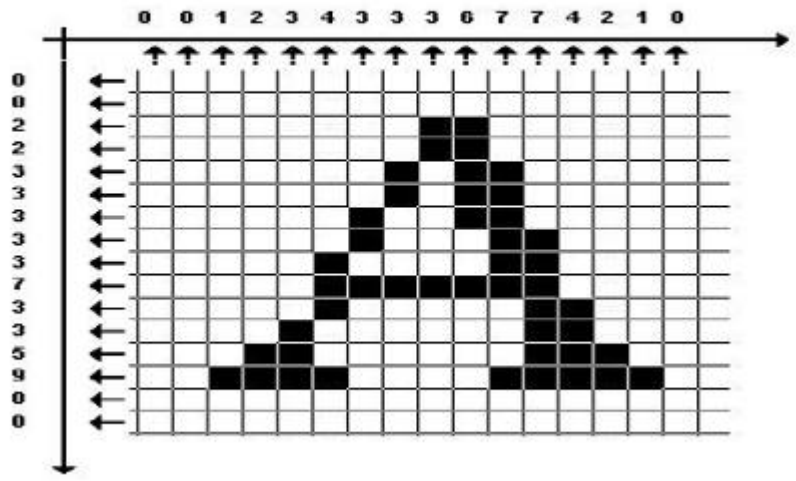

Figure 1.2 : exemple d'ensemble de pixels

2- Dimension

• C'est la taille de l'image.

• Elle se présente sous forme de matrice dont les

• Elle se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels).

• Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image

3- Résolution

• C'est la clarté ou la finesse de détails atteinte par un écran ou une imprimante dans la production d'images.

• Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

• On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou nombre total de pixels affichables horizontalement ou verticalement sur un moniteur;

• plus grand est ce nombre, meilleure est la résolution.

4- Bruit

• Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

5- Histogramme

• L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image.

• Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

• Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications.

• Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

6- Luminance

• C'est le degré de luminosité des points de l'image

• le mot luminance est substitué au mot brillance, qui • le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

• Une bonne luminance se caractérise par:

Des images lumineuses (brillantes);

Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses. L'absence de parasites.

7- Le contraste :

• C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image.

• Le contraste est défini en fonction des luminances de deux zones d'images.

• Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L1 - L2}{L1 + L2}$$

8- La texture

• Une texture est une région dans une image numérique

• Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont par exemple un motif basique qui se répète. La texture est composée de Texel, l'équivalent des pixels.

9- Contours et textures

• Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

• Les textures décrivent la structure de ceuxci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

3- Les types des images numériques

3-1- Images Vectorielles

Une image vectorielle est un ensemble de représentations d'entités géométriques : un cercle, un rectangle ou un segment [3]. Les entités géométriques sont représentées par des formules mathématiques appelées primitives graphiques : un rectangle est défini par deux points, un cercle

par un centre et un rayon, une courbe par plusieurs points et une équation. Chaque forme élémentaire constitue un objet et se voit assignée un certain nombre d'attributs : la couleur, la transparence, l'épaisseur du trait, le type de trait (pointillé, etc.). Le document numérisé prend la forme d'une suite de formules mathématiques décrivant les formes élémentaires constituant l'image. C'est le processeur qui est chargé de traduire ces formes en informations interprétables par la carte graphique [4].

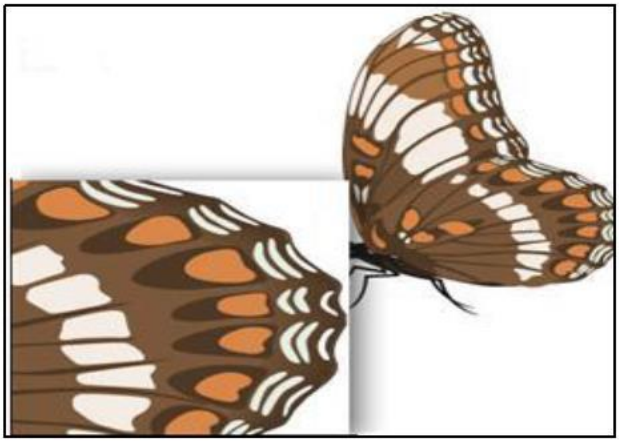

Figure 1.3 **:** Image vectorielle

3-2- Images matricielles (bitmap)

Une image bitmap est composée d'une matrice (tableau) de points à plusieurs dimensions[5]. Chaque dimension représentant une dimension spatiale (hauteur, largeur, profondeur), temporelle (durée) ou autre (par exemple, un niveau de résolution). Dans le cas des images à deux dimensions (le plus courant), les points sont appelés pixels. D'un point de vue mathématique, on considère l'image comme une fonction :

$$I : R^2 \rightarrow R^n, \ (x,y) \rightarrow I\ (x,y)$$

Tels que : x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. n est le nombre de composantes de l'image ($n \geq 1$) dans ce cas, l'espace R2 est appelé plan image.

Figure 1.4 : Image matricielle

4-L'imagerie Médicale

L'imagerie médicale désigne l'ensemble des techniques permettant de produire des représentations visuelles de l'intérieur du corps humain, à des fins médicales ou scientifiques. Ces images sont utilisées pour observer, diagnostiquer, traiter et suivre l'évolution de diverses pathologies. Grâce à ces technologies, les professionnels de santé peuvent visualiser les organes, les tissus et les structures internes sans avoir recours à des procédures invasives.

Les principales modalités d'imagerie médicale comprennent la radiographie, le scanner (CT), l'imagerie par résonance magnétique (IRM), l'échographie, et la médecine nucléaire (TEP, SPECT). Chacune de ces techniques repose sur des principes physiques différents (rayons X, ultrasons, champs magnétiques, etc.) et fournit des types d'informations spécifiques, selon le contexte clinique.

4-1-Techniques de l'imagerie médicale

L'imagerie médicale regroupe plusieurs techniques reposant sur différents principes physiques, permettant de visualiser les structures internes du corps humain. Chaque technique présente des avantages, des limites et des indications cliniques spécifiques. Voici les principales modalités :

A- Radiographie (Rayons X) :

La radiographie est un examen d'imagerie médicale permettant de visualiser les organes, les os ou les tissus de l'organisme. Cet examen utilise les rayons X, qui traversent le corps humain. Les rayons X sont absorbés par les différents tissus selon leur densité. Souvent, la radiographie est réalisée pour observer les structures osseuses, les poumons ou d'autres organes, comme les seins dans le cas d'une tumeur. Les tissus très denses, comme les os,

apparaissent en blanc, tandis que les tissus moins denses, apparaissent en gris, voire en noir.

Parfois, la radiographie classique ne permet pas la visualisation de certains organes concernés. La radiographie de contraste est alors prescrite, permettant une mise en évidence des organes à l'aide de l'injection d'un produit de contraste en intraveineuse, qui va les colorer en blanc [6].

Figure 1.5 : Exemple d'image de rayons X

B- Tomodensitométrie (Scanner ou CT)

La tomodensitométrie, plus communément connue sous le terme de "scanner", est apparue pour la première fois dès 1972. Cette technique d'imagerie médicale, réalisée par un radiologue, est également appelée scanner (ou CT-Scan : en anglais, computed tomography). Elle combine l'utilisation des rayons X à un système informatisé. Cela permet d'obtenir des images en coupe fine du corps. Son principe ? Le patient est allongé sur une table qui se déplace à travers un anneau.

L'anneau contient un tube à rayons X et un ensemble de détecteurs :

le faisceau de rayons X tourne autour du patient ;

les détecteurs à rayons X recueillent les caractéristiques des faisceaux qui ont traversé le corps du patient ;

analysées par un ordinateur, ces informations vont permettre la création d'une image. C'est en effet un algorithme mathématique de reconstruction d'image qui permet d'obtenir la vue de l'organe.

Les organes peuvent être étudiés individuellement. La tomodensitométrie permet ainsi de reconstituer des images en 2D ou en 3D des différentes structures anatomiques. La taille minimale de détection des lésions, notamment, est grandement améliorée avec le scanner [7].

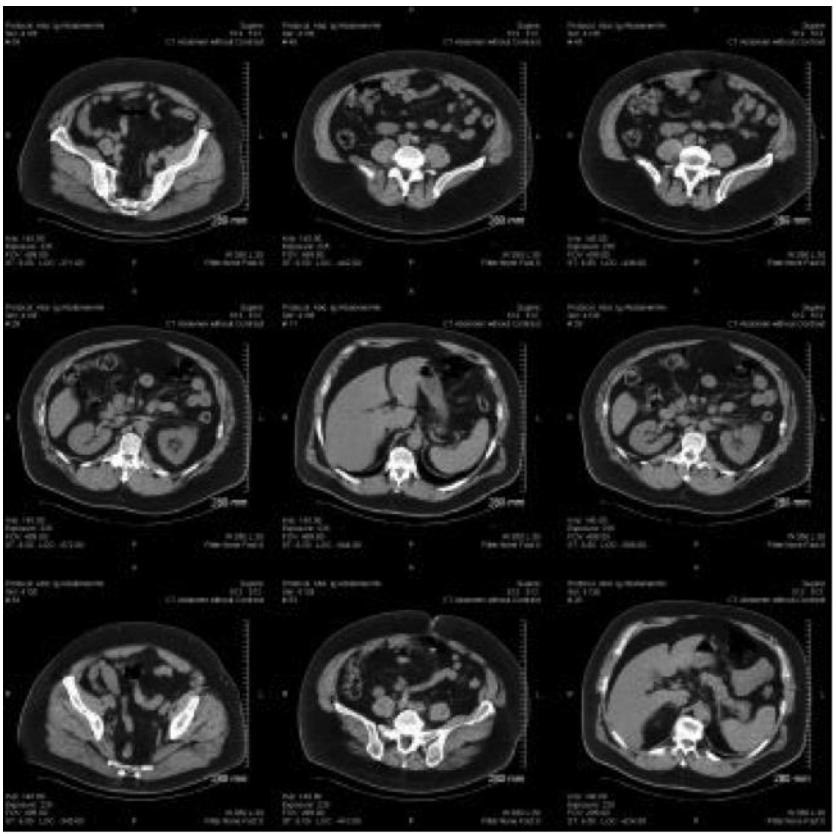

Figure 1.6 : Exemple d'image de tomodensitométrie

C- Imagerie par Résonance Magnétique (IRM)

L'imagerie par résonance magnétique (IRM) est un type d'imagerie médicale qui utilise un champ magnétique puissant et des ondes radio de très haute fréquence pour produire des images de haute résolution. Pendant une IRM, un ordinateur enregistre les changements dans le champ magnétique autour du corps d'une personne pour créer des images transversales détaillées. Contrairement à la TDM

(tomodensitométrie) et à la TEP (tomographie par émission de positons), l'IRM n'utilise pas de rayons X (rayonnements) et est d'ordinaire très sûre [8].

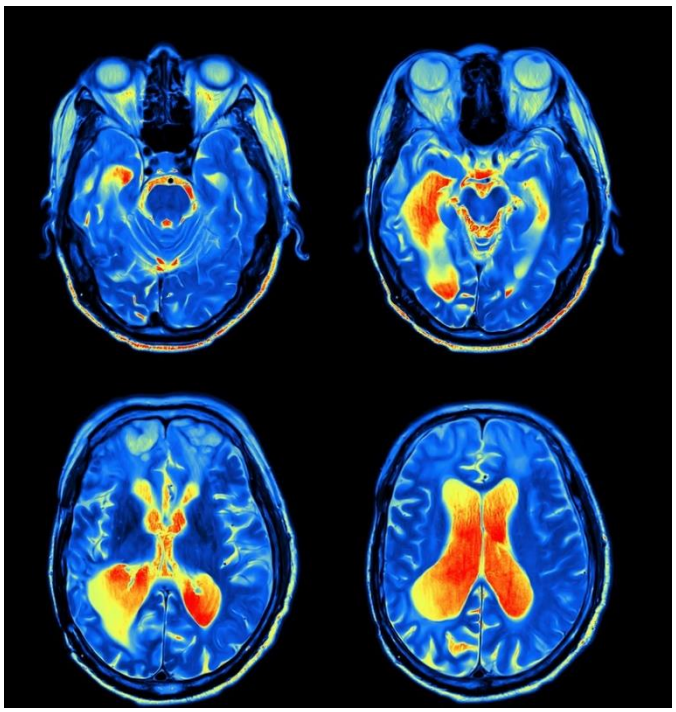

Figure 1.7 : Exemple d'IRM

Acquisition des images IRM :

L'imagerie par résonance magnétique (IRM) repose sur la détection du signal émis par les atomes d'hydrogène présents dans l'eau, qui est abondamment distribuée dans les tissus du corps humain. Comme ces tissus contiennent différents niveaux d'eau, plus la concentration en eau est élevée, meilleure est la qualité de l'image IRM obtenue.

Le processus de génération des images IRM commence par l'application d'un champ magnétique puissant sur le corps du patient. Ce champ aligne les molécules d'eau dans une direction précise (voir Figure).

Lorsque ce champ est interrompu, les molécules reviennent progressivement à leur état initial, un phénomène qui produit un signal mesurable. Ce signal est ensuite capté et interprété par un système informatique, qui reconstruit les données sous forme d'images volumétriques IRM.

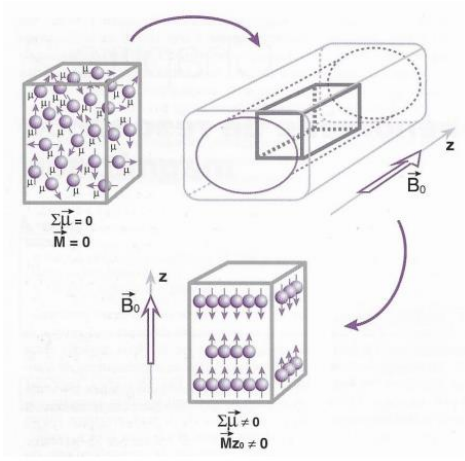

Figure 1.8 : Principe de la formation d'IRM

Contrairement à la tomodensitométrie (scanner), qui repose sur l'absorption des rayons X par les tissus traversés, l'IRM permet de visualiser avec précision les tissus mous grâce à leur forte teneur en eau. Les signaux émis varient en intensité selon les propriétés des tissus, ce qui permet d'obtenir des contrastes différenciés dans les images.

Les données issues de l'IRM forment un volume tridimensionnel de forme rectangulaire. Il est alors possible de les explorer selon les trois plans anatomiques de référence : le plan coronal, le plan axial et le plan sagittal (voir Figure), permettant une analyse complète et précise des structures internes du corps humain.

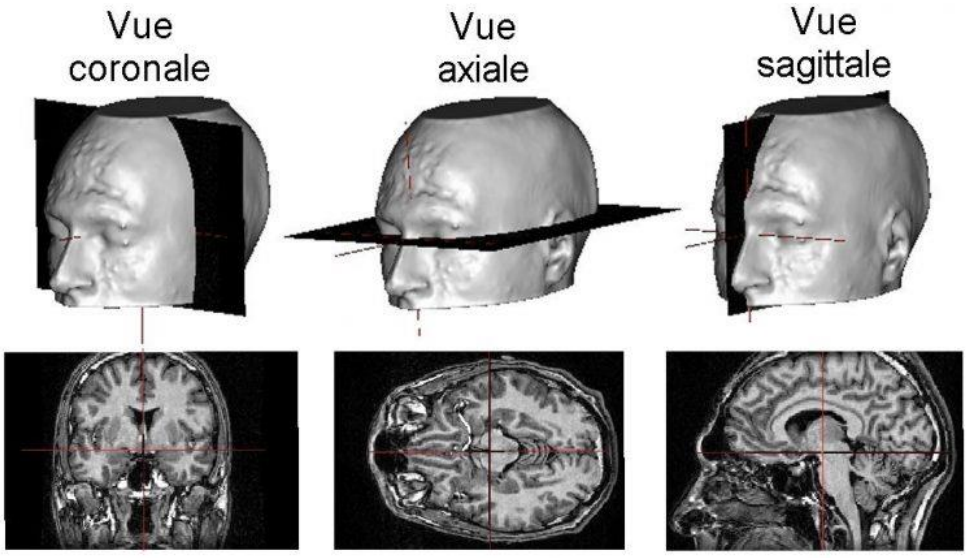

Figure 1.9 : Les 3 vues IRM selon les 3 principaux axes

D- Échographie (Ultrason)

L'échographie est une technique d'imagerie médicale qui repose sur l'utilisation d'ultrasons, des ondes sonores inaudibles, qui permettent de « visualiser » l'intérieur du corps.

Indolore et non invasive, elle est très largement utilisée pour le diagnostic de nombreuses pathologies mais aussi pour le guidage visuel lors d'autres examens, comme des biopsies.

L 'échographie est utilisée pour de très nombreuses indications, car elle peut être pratiquée sur n'importe quelle partie du corps : échographie cardiaque, rénale, pelvienne, des vaisseaux sanguins, du cerveau (échographie transfontanellaire du nouveau-né), de la prostate, de la thyroïde, etc [9].

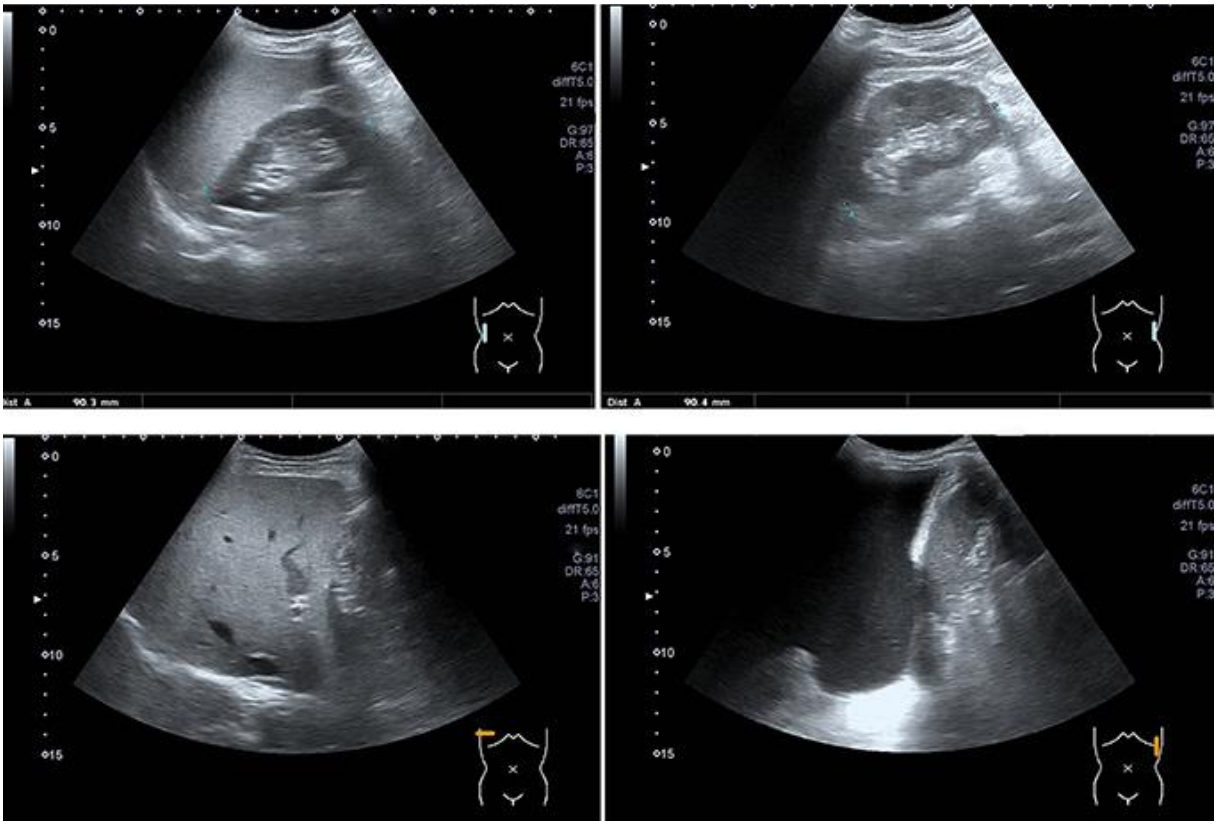

Figure 1.10 : Exemple d'échographie

E- La Médecine nucléaire

La médecine nucléaire est la spécialité médicale qui se consacre à l'utilisation de radioéléments (isotopes radioactifs) à des fins d'étude, de diagnostic et de traitement des pathologies affectant des organes ou des tissus.

Ces radioéléments émettent différents types de rayonnement, et notamment des photons, des positrons, des particules bêta négatives ou encore des rayons alpha. Cela permet de voir et d'analyser le fonctionnement de divers organes.

Ainsi, le médecin peut accéder à une imagerie fonctionnelle et métabolique pour détecter des lésions pouvant être profondes, surveiller leur évolution, mais aussi guider précisément un éventuel geste chirurgical à pratiquer.

La médecine nucléaire a recours à quatre types d'images :

- les images dynamiques : une série d'image capte le mouvement ou l'activité et permet par exemple d'observer le débit sanguin à proximité d'un organe ;
- les images statiques ou planaires : les images sont à deux dimensions et apparaissent une à la fois ;
- les images du corps entier : elles sont à deux dimensions et représentent l'avant ou l'arrière du corps ;
- les images par tomographies d'émission à photon unique : les images sont à trois dimensions et permettent de visualiser le fonctionnement de l'organe étudié.

Notons que la médecine nucléaire complète la radiologie, l'échographie ou encore l'imagerie par résonance magnétique (IRM).

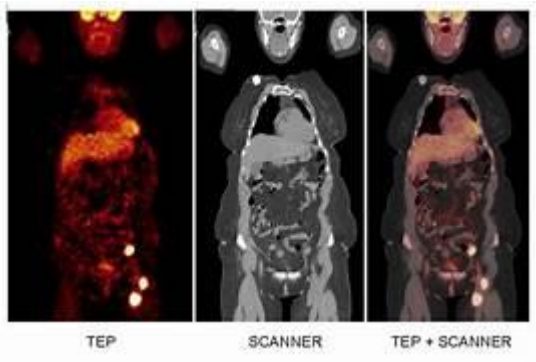

Figure 1.11 : Exemple d'image Scintigraphies (Médecine Nucléaire)

5-Segmentation d'image

5-1- Définition de la segmentation

La segmentation d'images est une technique de computer vision qui consiste à découper de façon automatique une image en zones de pixels appartenants à une même classe d'objets. La segmentation d'images a de nombreuses applications, notamment en imagerie médicale.

La segmentation d'images se divise en deux types, la segmentation sémantique (semantic segmentation) et la segmentation par instance (instance segmentation). Dans le cadre de la segmentation sémantique, on cherche à classifier les pixels de l'image comme appartenants ou non à une certaine catégorie. La segmentation par instance en revanche, permet d'avoir plus d'informations sur l'image, en divisant les différentes instances d'un même objet.

5-2-Fonctionnement de la segmentation d'image

Les méthodes de segmentation d'images ont longtemps reposé sur des techniques de détection de contours classiques. Il s'agit de méthodes mathématiques qui utilisent des opérateurs simples comme la dérivée et le gradient pour détecter les pixels ou un changement brusque d'intensité se produit. Le modèle détecte ainsi tous les points ou des changements brusques sont repérés, et reconstruit les contours en liant ces points.

Aujourd'hui, de plus en plus de techniques sont proposées et les modèles sont de plus en plus performants. Dans cette partie je présente quelques unes de ces techniques.

5-2-1- Clustering par k-means appliqué à la segmentation d'images

La segmentation d'images est à la base un problème de classification de pixels. Chaque pixel de l'image doit être associé à un ensemble d'autres pixels, que l'on affecte à une classe.

Une des méthodes envisageables pour séparer les pixels en classes est l'utilisation de l'algorithme k-means. Premièrement, on sélectionne un nombre de classes et un ensemble de centres de classes qui correspondent à des pixels sur l'image. Ensuite, une distance entre les centres de classe et chaque pixel de l'image est calculée. Enfin, chacun pixel est affecté à la classe qui minimise la distance entre ce pixel et le centre de la classe.

La distance euclidienne est souvent choisie pour trouver la classe d'un pixel donnée. Pour trouver la distance minimale, il faut passer par des méthodes heuristiques, surtout dans le cas d'images de grandes dimensions[10].

5-2-2- Découpage en régions et superpixel

Une autre méthode est souvent utilisée pour découper les images en régions, elle repose sur la construction de superpixels. Un superpixel est un assemblage de pixels proches, pour lesquels on a une variance très faible. Les superpixels sont des régions qui ont plus de sens visuellement qu'un pixel isolé.

Un des algorithmes qui permettent de construire ces superpixels est l'algorithme SLIC (Simple Linear Iterative Clustering). Cet algorithme est une variante de la méthode des k-means.

Les pixels sont d'abord encodés sous forme d'un vecteur de taille 5 (pour des images colorées). On a les intensité des différents canaux, R, G et B, et deux valeurs x et y qui

correspondent à la position du pixel sur l'image. On doit fixer le nombre de superpixels souhaité, plus il y a de superpixels, moins on perd d'informations. Ce nombre de superpixels va determiner les centres de clusters en les répartissant de façon uniforme sur toute l'image. L'approche basée sur la distance utilisée dans k-means permet ensuite d'affecter chaque pixel à un superpixel.

En pratique, la technique des superpixels est surtout utilisée comme une étape de preprocessing qui permet de réduire la complexité de l'image et simplifier les calculs[10].

5-2-3- Les réseaux de convolutions (CNN) pour la segmentation d'images

Les récents progrès en deep learning et la démocratisation des réseaux de convolutions ont permis d'améliorer considérablement les techniques de segmentation d'images. Les architectures couramment utilisées sont les mêmes que pour d'autres tâches de vision par ordinateur, comme VGG-16 ou les ResNet.

L'architecture la plus efficace pour la segmentation est mask R-CNN. Elle consiste en 3 réseaux modèles de convolutions indépendants. Le premier modèle permet d'extraire des caractéristiques de l'image. Le second réseau permet de détecter une bounding box (un rectangle) autour de l'instance d'un objet, on parle de region proposal. Le dernier modèle aura pour but d'affiner la prédiction pour trouver les contours de cet objet de façon plus fine[10].

Plus récemment, les transformers ont encore améliorer les performances. De manière général, il est clair que les transformers vont constituer la prochaine révolution en vision par ordinateur et en deep learning.

5-3- Les techniques de segmentation d'image

5-3-1- Approche par contours (ou approche frontière)

L'approche par contours, aussi appelée approche frontière, repose sur la détection des discontinuités dans l'intensité de l'image, qui correspondent souvent aux bords des objets. L'objectif est de déterminer les lignes de séparation entre les régions distinctes en analysant les variations brutales des pixels. Cette méthode est particulièrement utile lorsque les objets dans l'image présentent des contours nets et bien définis, ce qui est souvent le cas dans certaines modalités d'imagerie médicale comme le scanner ou l'IRM.

A- Méthodes dérivatives (Edge Detection)

Les méthodes dérivatives consistent à calculer le gradient de l'image pour identifier les zones où les changements d'intensité sont les plus marqués. Ces changements sont interprétés comme des bords potentiels[11].

- Opérateurs de détection classiques :

Sobel : utilise deux masques de convolution pour calculer les dérivées premières de l'image selon les directions horizontale et verticale.

Prewitt : similaire à Sobel mais avec un masque différent, moins sensible au bruit.

Roberts : calcule la différence entre les pixels diagonaux voisins.

- Opérateur de Canny :

Considéré comme la référence en détection de bords, il combine plusieurs étapes :

Filtrage Gaussien pour réduire le bruit.

Calcul du gradient en intensité.

Suppression des non-maxima : élimine les réponses non maximales dans la direction du gradient.

Seuillage avec hystérésis : conserve uniquement les bords significatifs tout en évitant les détections isolées.

B- Méthodes déformables (Active Contours / Snakes)

Les modèles déformables, ou active contours, sont des courbes qui s'ajustent automatiquement pour épouser les contours d'un objet dans une image. Ils fonctionnent en minimisant une fonction d'énergie définie selon deux composantes [12] :

Énergie interne : contrôle la régularité et la continuité de la courbe.

Énergie externe : guide la courbe vers les contours en s'appuyant sur des données d'image (comme les gradients).

Deux grandes familles existent :

Snakes classiques (Kass et al., 1988) : requièrent une bonne initialisation proche des bords à détecter.

Modèles basés sur les level sets (Osher & Sethian, 1988) : permettent de gérer automatiquement les changements topologiques, comme la fusion ou la division d'objets.

Les level sets représentent la courbe comme une surface implicite dans un espace à dimension supérieure, ce qui offre une grande flexibilité dans le suivi des contours complexes.

5-3-2- Approche par région

L'approche par région est l'une des techniques fondamentales en segmentation d'image, utilisée pour extraire des objets ou des structures d'intérêt dans une image en se basant sur la similarité des pixels adjacents. Contrairement à l'approche par contours qui cherche les discontinuités dans les valeurs de pixels, l'approche par région vise à regrouper les pixels qui présentent une homogénéité selon un ou plusieurs critères : intensité, couleur, texture, ou encore statistiques locales.

Ce type de segmentation est particulièrement utile en imagerie médicale, car les différentes structures anatomiques (comme les tissus mous, les organes, ou les tumeurs) tendent à présenter des régions relativement homogènes dans certaines modalités d'imagerie comme l'IRM ou le scanner.

A- Croissance de région (*Region Growing*)

La méthode de croissance de région consiste à choisir un ou plusieurs pixels de départ, appelés graines (seeds), puis à étendre progressivement la région en ajoutant les pixels voisins qui satisfont un critère de similarité. Cette opération continue jusqu'à ce qu'aucun autre pixel voisin ne remplisse les conditions d'homogénéité [13].

Étapes typiques :

-Sélection des graines (manuellement ou automatiquement)

-Définition du critère de similarité (ex. : différence d'intensité < seuil)

-Agrandissement de la région jusqu'à stabilisation

Avantages :

-Méthode simple, intuitive et efficace lorsque les régions sont bien définies.

-Donne des résultats précis dans des images peu bruitées.

Inconvénients :

-Très sensible au bruit, aux artefacts, ou aux gradients d'intensité.

-Fortement dépendante au choix des graines et du critère.

B -Fusion de régions (*Region Merging*)

Dans cette approche, l'image est initialement divisée en de très petites régions (souvent chaque pixel ou blocs). Des tests de similarité sont effectués pour fusionner progressivement ces petites régions en plus grandes. Le processus continue jusqu'à ce que toutes les régions soient suffisamment homogènes[14].

Avantages :

-Moins sensible au choix initial des graines.

-Permet un contrôle plus rigoureux de l'uniformité des régions.

Inconvénients :

-Nécessite une stratégie de fusion rigoureuse (seuil, ordre des fusions...).

-Peut générer des régions trop étendues si les paramètres sont mal réglés.

C- Division de régions (*Region Splitting*)

La méthode de division part du principe inverse de la croissance : au lieu de partir de petites régions pour les agrandir, on considère l'image entière comme une seule région qu'on divise

récursivement. À chaque étape, une région est divisée (souvent en quatre) si elle ne respecte pas un critère d'homogénéité.

Avantages :

-Adapté à des images avec grandes zones homogènes et peu de bruit.

-Évite l'initialisation par graines.

Inconvénients :

-Génère souvent des formes carrées, ce qui peut ne pas correspondre aux structures naturelles.

-Peut nécessiter un grand nombre de divisions avant de converger.

D-Division-fusion (Split and Merge)

Cette approche hybride combine les avantages de la division et de la fusion. Elle commence par une division récursive de l'image (comme dans l'approche précédente), suivie d'une phase de fusion où les régions adjacentes similaires sont regroupées[15].

Processus :

Division : segmentation initiale en régions plus petites jusqu'à ce qu'un critère d'homogénéité soit respecté.

Fusion : regroupement des régions similaires adjacentes.

Avantages :

Meilleure robustesse et adaptabilité.

Peut produire des formes plus naturelles que la division seule.

Inconvénients :

Plus complexe à implémenter.

Peut nécessiter plusieurs passes pour converger vers un résultat stable.

5-3-3-Approche par classification

La segmentation par classification traite la segmentation d'image comme un problème de classification des pixels. Chaque pixel (ou ensemble de pixels) est classifié dans une catégorie ou une classe prédéfinie en fonction de ses caractéristiques (intensité, texture, position, etc.). Elle est très employée dans les systèmes intelligents et dans les applications médicales où la précision est cruciale.

On distingue généralement trois types de méthodes : les non supervisées, les supervisées, et les méthodes d'ensemble.

A- Les méthodes non supervisées

comme K-means, Fuzzy C-means ou Mean Shift permettent de segmenter une image sans connaissance préalable, en regroupant les pixels similaires dans des clusters. Elles sont particulièrement adaptées lorsque l'on ne dispose pas de données annotées. Leur principale

limite réside dans leur sensibilité au bruit et dans le choix des paramètres initiaux (comme le nombre de classes).

B- Les méthodes supervisées

Quant à elles, s'appuient sur un ensemble d'entraînement labellisé pour apprendre à classer les pixels. Des algorithmes tels que les Machines à Vecteurs de Support (SVM), les Random Forests, ou les réseaux de neurones convolutionnels (CNN) sont couramment utilisés. Ces approches offrent des résultats précis, surtout en présence d'une grande quantité de données d'entraînement, mais peuvent souffrir de sur-apprentissage (overfitting) si les données ne sont pas suffisamment variées ou bien préparées.

C- **Les méthodes d'ensemble**

Comme le bagging, boosting, ou Random Forest, cherchent à combiner plusieurs modèles faibles pour en produire un plus robuste. En traitement d'image médicale, elles permettent souvent d'améliorer la précision globale de la segmentation, en réduisant les erreurs commises par chaque modèle pris isolément. Leur principal inconvénient est la complexité computationnelle, qui peut devenir importante selon le nombre de classifieurs impliqués.

5-4-Difficultés de la segmentation des images médicales

La segmentation des images médicales présente plusieurs difficultés majeures dues à la complexité des données et aux contraintes du domaine médical. D'abord, **la variabilité anatomique inter-individuelle** constitue un obstacle important. Les structures anatomiques peuvent varier significativement d'un patient à un autre en termes de forme, taille, ou orientation, rendant difficile l'utilisation de modèles standards.

Ensuite, **les artefacts d'acquisition** et **le bruit** dans les images peuvent dégrader la qualité des données. Par exemple, en imagerie par résonance magnétique (IRM), des distorsions peuvent apparaître, tandis qu'en tomodensitométrie (CT), le bruit peut masquer des détails importants. Ces perturbations rendent la détection des contours ou la classification des tissus plus complexe.

Une autre difficulté réside dans la **faible différenciation des intensités** entre les tissus. Certains organes ou structures ont des niveaux de gris très similaires, ce qui empêche les algorithmes basés uniquement sur l'intensité de bien segmenter les régions. Cela impose

souvent le recours à des méthodes plus avancées, intégrant par exemple des informations de texture ou de forme.

De plus, **la disponibilité limitée de données annotées** représente une contrainte majeure. La segmentation supervisée repose sur des bases de données médicales annotées manuellement par des experts, un processus coûteux et long. Ce manque de données d'apprentissage complique le développement de modèles précis et généralisables.

Enfin, la **validation clinique** des méthodes de segmentation reste un enjeu crucial. Les algorithmes doivent être à la fois précis, robustes et fiables dans un contexte réel. Toute erreur de segmentation peut avoir des conséquences médicales graves, ce qui exige des procédures d'évaluation rigoureuses avant toute intégration dans les systèmes cliniques.

6-Conclusion

Ce chapitre a permis de passer en revue les concepts fondamentaux liés à la représentation numérique des images, en particulier dans le contexte de l'imagerie médicale. Nous avons distingué les types d'images, leurs principales caractéristiques techniques (résolution, bruit, histogramme, luminance, etc.), et détaillé les différentes modalités d'acquisition d'images médicales, notamment l'IRM, la tomodensitométrie, et l'échographie. L'accent a été mis sur l'importance de la segmentation d'images, indispensable pour extraire des informations utiles à partir d'images complexes. Les différentes approches, allant des techniques traditionnelles de détection de contours aux méthodes avancées basées sur l'intelligence artificielle comme les CNN et les transformers, ont été présentées. Ces connaissances constituent une base solide pour les développements ultérieurs de notre mémoire, orientés vers l'amélioration des performances de la classification d'images IRM à l'aide d'algorithmes intelligents.

Chapitre 02 : Apprentissage automatique

1-Introduction

L'évolution rapide des technologies informatiques a permis l'émergence de disciplines puissantes, dont l'intelligence artificielle (IA), qui redéfinit les limites de l'automatisation et de la prise de décision autonome. Parmi les branches les plus prometteuses de l'IA figure l'apprentissage automatique (machine learning), qui permet aux systèmes informatiques d'apprendre à partir des données et d'améliorer leurs performances sans intervention humaine explicite. Ce chapitre s'intéresse aux concepts fondamentaux de l'intelligence artificielle, aux différentes approches de l'apprentissage automatique, ainsi qu'aux algorithmes les plus couramment utilisés dans ce domaine, comme les arbres de décision, les machines à vecteurs de support et les réseaux de neurones. Une attention particulière est également portée à l'apprentissage profond et aux réseaux de neurones convolutifs (CNN), qui sont à la base des avancées récentes dans des domaines tels que la vision par ordinateur ou le traitement automatique du langage naturel.

2-Intelligence Artificielle

L'intelligence artificielle (IA) est l'étude des méthodes permettant aux ordinateurs d'accomplir des tâches considérées autrefois comme exclusivement réservées aux êtres humains [16].

Une autre définition décrit l'intelligence artificielle comme un ensemble de technologies et de théories issues de l'informatique, mais aussi comme un domaine d'intérêt lié à la pensée, à la logique et à l'intelligence. Elle vise à développer des programmes capables de résoudre des problèmes, de traiter le langage et d'exécuter diverses tâches complexes, autrefois propres à l'intelligence humaine [17].

3-Apprentissage automatique « machine-learning »

L'apprentissage automatique est une branche de l'intelligence artificielle. Il constitue aujourd'hui l'un des outils les plus utilisés pour résoudre de nombreux problèmes concrets. Cette technique permet à un ordinateur d'exécuter des tâches complexes et de faire des prédictions à partir des données, sans intervention humaine directe ni programmation explicite.

Contrairement aux méthodes traditionnelles qui reposent sur des règles codées manuellement, les algorithmes d'apprentissage automatique offrent rapidité et efficacité accrues. Ils sont particulièrement utiles pour concevoir des programmes capables de traiter, analyser et

interpréter de grandes quantités de données complexes, ou encore de résoudre des problèmes impossibles à gérer manuellement.

L'essor de l'apprentissage automatique s'explique notamment par l'augmentation massive des volumes et de la diversité des données, par l'accès facilité à une puissance de calcul élevée, ainsi que par la généralisation de l'Internet haut débit.

3-1-Les types d'apprentissage automatique

Il existe plusieurs méthodes pour effectuer l'apprentissage automatique. Elles peuvent être classées en quatre catégories principales selon leur besoin d'intervention de l'acteur humain. Cela conduit à l'apprentissage supervisé, à l'apprentissage non supervisé, à l'apprentissage semi-supervisé, et à l'apprentissage par renforcement que nous expliquerons ci-dessous. L'auto-apprentissage et la détection d'anomalies sont également considérés comme des types d'apprentissage avec certaines particularités.

3-1-1-Apprentissage supervisé

L'apprentissage supervisé consiste à prédire une valeur cible à partir d'observations fournies en entrée. En apprentissage automatique, ces données d'entrée sont appelées caractéristiques, tandis que les valeurs que le modèle apprend à prédire sont généralement désignées par le terme étiquettes[18-19].

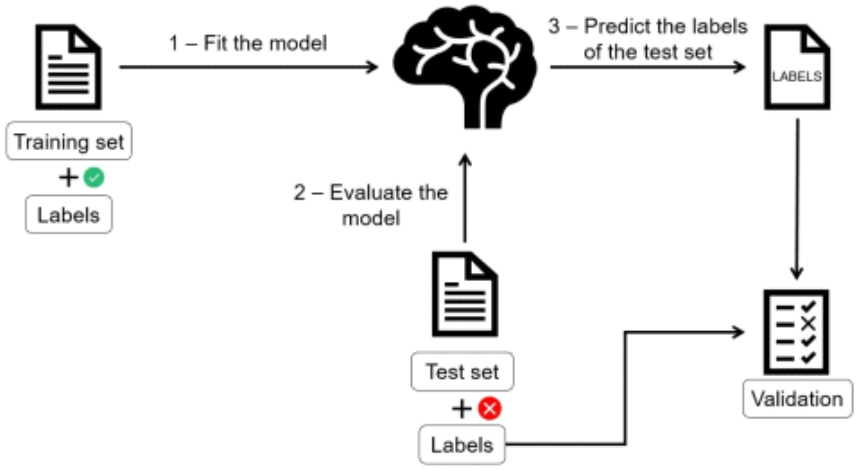

Figure 2.1 : L'apprentissage supervisé [20].

3-1-2-Apprentissage non supervisé

Ce type d'apprentissage repose sur l'utilisation de données non étiquetées, fournies à l'algorithme sans indication préalable. L'objectif est qu'il identifie, par lui-même, des similarités ou des structures cachées au sein de ces données. Cette approche est précieuse, car

les données brutes non étiquetées sont généralement beaucoup plus abondantes que les données annotées.

L'apprentissage non supervisé permet ainsi de découvrir des motifs sous-jacents dans les ensembles de données ou de déterminer automatiquement les caractéristiques des entrées. Par exemple, il peut être utilisé pour générer des étiquettes de manière automatique.

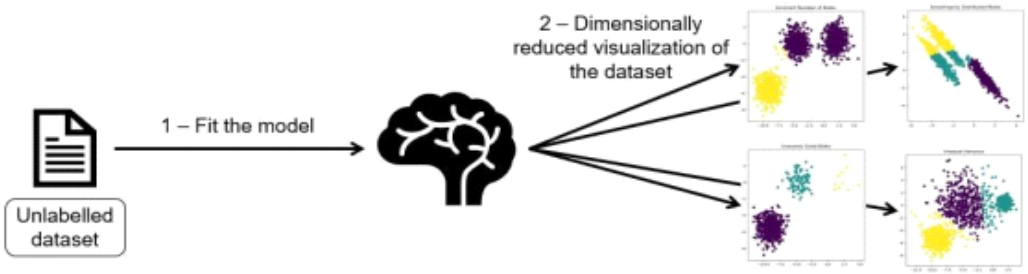

Figure 2.2 : Apprentissage non supervisé [20].

3-1-3-Apprentissage semi-supervisé

L'apprentissage semi-supervisé combine les deux approches précédentes, en exploitant à la fois des données étiquetées et non étiquetées. Cette méthode peut, dans certains cas, améliorer les performances d'un algorithme. Toutefois, son efficacité dépend fortement de la sensibilité de l'algorithme aux données utilisées ; elle peut donc aussi entraîner une dégradation des résultats si les données non étiquetées influencent négativement l'apprentissage.

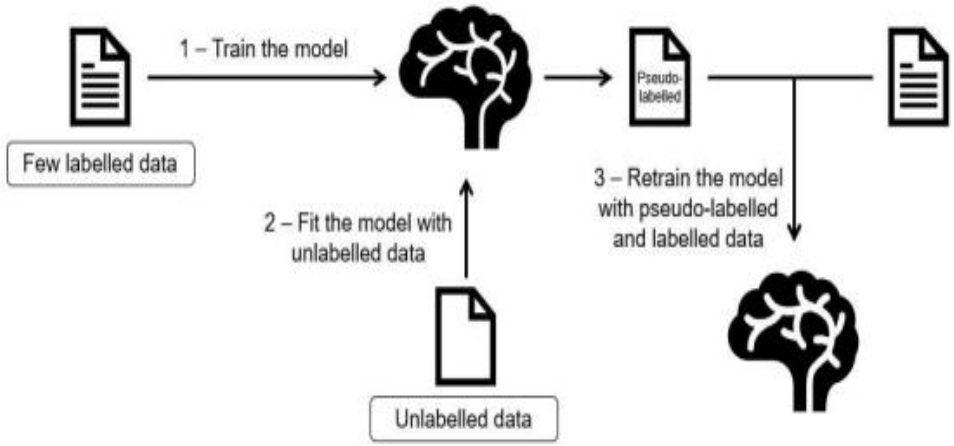

Figure 2.3 : Apprentissage semi-supervisé [20].

3-1-4-Apprentissage par renforcement

L'apprentissage par renforcement est une forme d'apprentissage automatique qui permet à une machine d'apprendre, de manière autonome, le comportement optimal à adopter dans un environnement donné afin d'optimiser ses performances.

Cette méthode repose sur l'analyse des interactions entre l'agent (le programme intelligent) et son environnement. En observant les conséquences de ses actions, l'agent cherche à maximiser les récompenses obtenues ou à minimiser les risques. Ainsi, il affine progressivement sa stratégie pour atteindre ses objectifs de manière efficace [21].

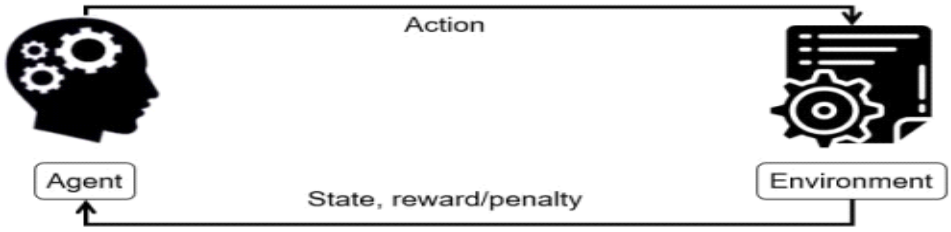

Figure 2.4 : Apprentissage par renforcement [20].

4-Quelques Algorithmes d'apprentissage automatique

4-1-Les machines à vecteur de support SVM

Les machines à vecteurs de support (SVM) sont largement utilisées en apprentissage automatique pour résoudre des tâches de classification et de régression. Leur principe consiste à définir des frontières de décision permettant de séparer les données en différentes classes au sein d'un espace de caractéristiques, souvent de haute dimension.

Les SVM cherchent à maximiser la marge entre ces classes, c'est-à-dire la distance entre la frontière de décision et les points les plus proches de chaque classe, afin d'assurer une meilleure généralisation du modèle.

L'un des atouts majeurs des SVM réside dans leur capacité à gérer les problèmes de classification non linéaire grâce à l'utilisation de la méthode du noyau. Cette technique permet de projeter implicitement les données dans un espace de dimension supérieure, facilitant ainsi la séparation entre les classes.

Cependant, le choix du noyau approprié joue un rôle crucial dans les performances du modèle. Il peut nécessiter plusieurs essais afin de l'adapter aux spécificités de l'ensemble de données utilisé.

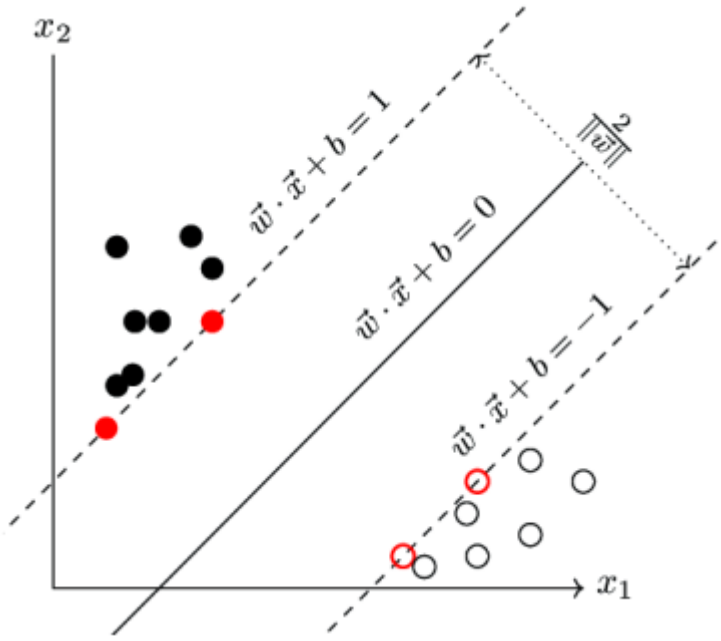

Figure 2.5 : Exemple simple représentant la classification par SVM [22]

4-2- L'Arbre de décision

Un arbre de décision est une représentation graphique des choix possibles et de leurs conséquences, organisée sous forme d'une structure arborescente. Les branches (ou arêtes) de l'arbre correspondent aux critères ou règles de décision, tandis que les nœuds symbolisent des événements ou des options. Chaque nœud contient un ensemble de caractéristiques servant à la classification, et chaque branche indique une valeur possible associée à l'attribut du nœud [23].

Lors de la classification d'une instance à l'aide d'un arbre de décision, l'algorithme démarre au niveau du nœud racine et progresse en évaluant successivement les attributs définis par les nœuds. Il suit ensuite la branche correspondant à la valeur observée de l'attribut pour l'instance à classer. Le processus se poursuit jusqu'à atteindre un nœud feuille, lequel indique la classe finale attribuée à l'instance.

Pour décider comment diviser les nœuds, deux mesures sont fréquemment utilisées : l'indice de Gini (ou "gini") pour évaluer l'impureté, et l'entropie pour mesurer le gain d'information. Ces critères permettent de choisir l'attribut optimal à chaque étape, en maximisant la qualité de la séparation des données.

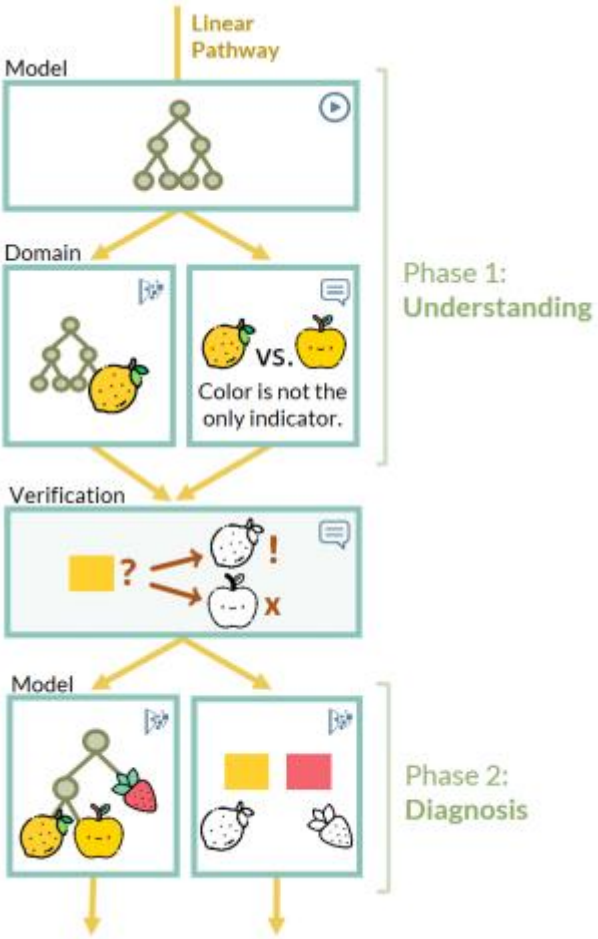

Figure 2.6 : Un arbre de décision pour un apprentissage non linéaire. [24]

4-3- Forêt aléatoire

La forêt aléatoire est une méthode d'apprentissage par ensemble qui repose sur la combinaison de plusieurs arbres de décision, dont les prédictions sont ensuite agrégées par vote (pour la classification) ou moyennage (pour la régression). Cette approche vise à limiter le surapprentissage et à améliorer la précision globale du modèle.

Elle repose sur la construction de nombreux arbres de décision présentant une certaine diversité, obtenue grâce à des techniques comme le bagging (bootstrap aggregating) et la sélection aléatoire des caractéristiques à chaque nœud.

Les forêts aléatoires sont capables de traiter efficacement des variables aussi bien catégorielles que numériques, ce qui en fait une solution polyvalente, adaptée à des tâches de classification comme de régression.

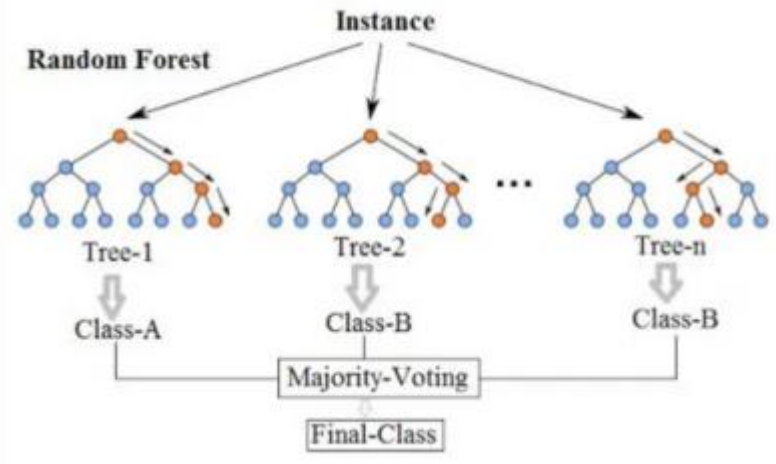

Figure 2.7 : Un exemple de la structure d'une forêt aléatoire prenant en compte plusieurs arbres de décision [25]

4-4-Le clustering K-moyennes mobiles

L'algorithme des K-moyennes (k-means), introduit par MacQueen en 1967, est une méthode d'apprentissage automatique non supervisé largement utilisée pour le clustering ou regroupement de données. Son but est de diviser un ensemble de données en k groupes (ou clusters), le nombre k étant généralement défini à l'avance par l'analyste.

Le principe de base de cette méthode consiste à regrouper les données de manière à maximiser la similarité entre les éléments d'un même cluster (forte similarité intra-classe) tout en minimisant la ressemblance entre les clusters différents (faible similarité inter-classe).

Chaque cluster est représenté par un centroïde, qui correspond à la moyenne des points qui lui sont assignés. Ce centroïde sert de point de référence et reflète les caractéristiques principales du groupe.

5-Apprentissage profond (Deep Learning)

L'apprentissage profond (Deep Learning) est un sous-ensemble des méthodologies et techniques d'apprentissage automatique qui utilisent plusieurs couches cachées de réseaux neuronaux artificiels (ANNs). Il s'agit de l'adaptation de réseaux neuronaux qui imitent la structure du cerveau humain. La force de l'apprentissage profond réside dans le fait que la machine peut extraire des caractéristiques et apprendre par elle-même, indépendamment de l'intervention d'un expert. La méthodologie de l'apprentissage profond applique des transformations non linéaires et des abstractions de modèle de haut niveau dans de grandes bases de données [26].

6-Réseaux De Neurones (RNs)

Pratiquement tous les algorithmes d'apprentissage profond sont des réseaux de neurones, également appelés ANNs. Les ANNs sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique. Cela est similaire à la manière dont le cerveau traite l'information au niveau fonctionnel. Tous les réseaux neuronaux sont composés de neurones interconnectés organisés en couches [27].

Les détails du neurone et la description de son fonctionnement seront expliqués dans la prochaine partie.

6-1- Neurone biologique

Le neurone, élément fondamental du système nerveux central, se compose d'un corps cellulaire, appelé soma, d'où partent de nombreuses ramifications appelées dendrites. Ces dernières acheminent l'information vers le soma. L'influx nerveux est ensuite transmis à d'autres neurones via une longue extension appelée axone.

La communication entre les neurones s'effectue au niveau de la synapse, un espace microscopique séparant l'axone d'un neurone des dendrites d'un autre. Cette transmission se fait par l'intermédiaire de messagers chimiques appelés neurotransmetteurs[28].

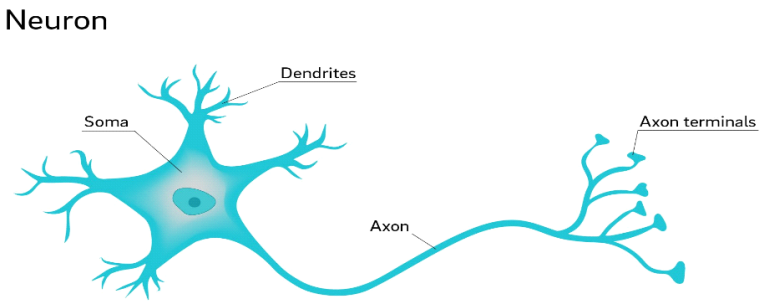

Figure 2.8 : Neurone biologique.

Le cerveau est composé d'environ un trillion de cellules nerveuses appelées neurones [29]. La sortie de chaque neurone est liée à des milliers d'autres neurones. Chaque neurone effectue un traitement local. Il recueille les signaux des dendrites et les additionne. Si l'amplitude résultante dépasse un certain seuil interne, un signal est envoyé à d'autres neurones.

6-2- Neurone artificiel

Un neurone artificiel est un modèle mathématique inspiré du fonctionnement des neurones biologiques. Il peut être vu comme un opérateur recevant plusieurs entrées, issues de l'environnement ou d'autres neurones, chacune étant associée à un poids synaptique qui reflète son importance. Le neurone artificiel calcule alors une somme pondérée de ces entrées et ne génère une sortie que si cette somme dépasse un seuil d'activation prédéfini.

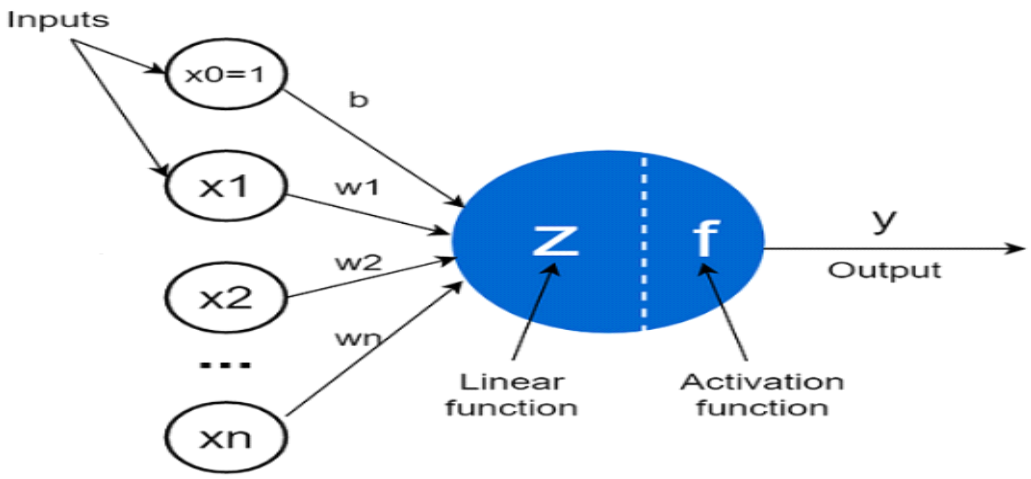

Figure 2.9 : Neurone artificielle.

Le neurone a plusieurs entrées ($x1$, $x2$,……, xn), associées à leurs poids ($w1$, $w2$,……, wn), en plus d'une entrée de biais $x0 = 1$ et son poids $w0$, et une seule sortie y est calculée via l'application d'une fonction d'activation, par exemple, la fonction sigmoïde (σ) sur z qui est la somme de la multiplication des entrées par leurs poids.

Les réseaux de neurones artificiels se composent de couches, qui peuvent être les suivantes :

- **Couche d'entrée** : La première couche est la couche d'entrée ; la couche d'entrée se compose d'un vecteur représentant les données sous forme vectorisée.
- **Couche cachée** : Les couches cachées sont des couches intermédiaires entre la couche d'entrée et la couche de sortie.
- **Couche de sortie** : Elle produit les résultats du réseau neuronal en fonction des entrées données.

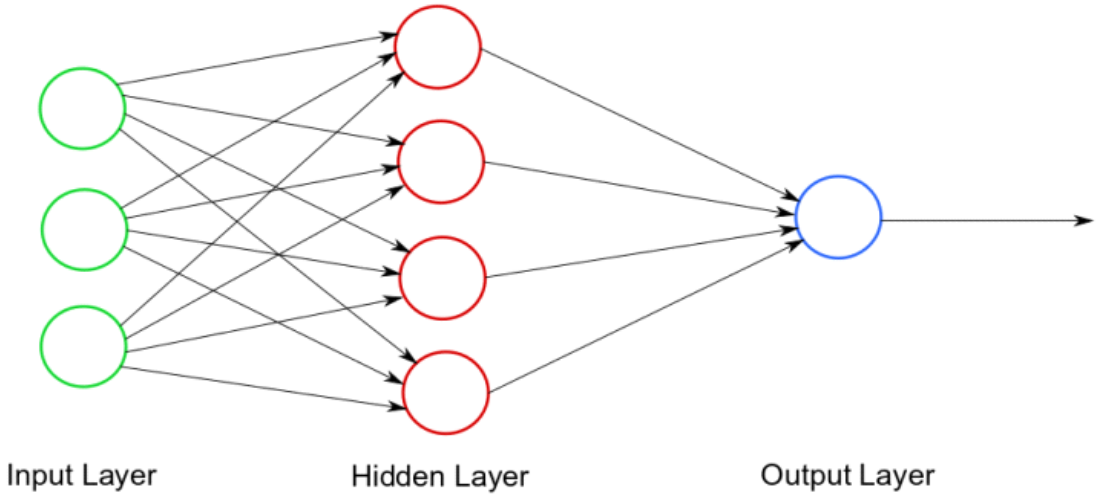

Figure 2.10 : Couches de réseaux de neurones.

6-3- Fonctions d'activation

Les fonctions d'activation jouent un rôle essentiel dans le bon fonctionnement des réseaux neuronaux, en particulier lorsqu'ils comportent plusieurs couches. Bien que les poids soient ajustés dans chaque couche, leur apprentissage dépend du calcul du gradient de la fonction de perte globale par rapport à chaque vecteur de poids.

Ce gradient doit être transmis efficacement à travers toutes les couches du réseau. Sa valeur dépend directement de la manière dont l'entrée de chaque couche est transformée, c'est-à-dire de la fonction d'activation utilisée. Pour maintenir un apprentissage stable et efficace, il est préférable d'utiliser des fonctions d'activation non saturantes, qui évitent la disparition du gradient [30].

Parmi les fonctions les plus couramment utilisées, on peut citer plusieurs exemples illustrés dans la Figure 15.

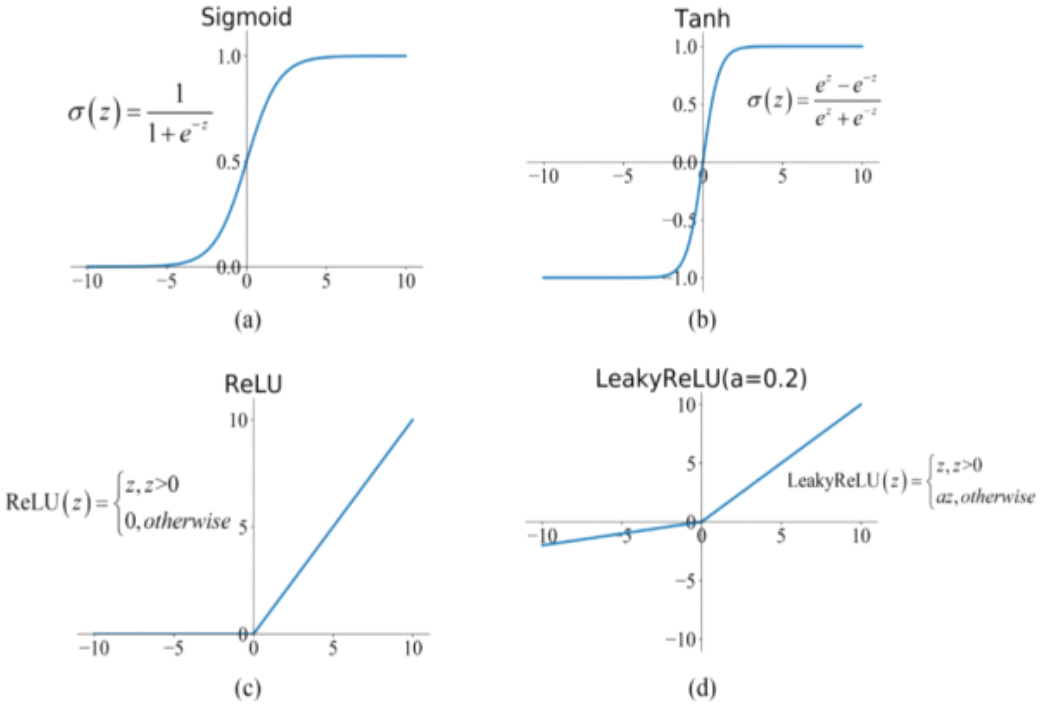

Figure 2.11 : Fonctions d'activation couramment utilisées [31].

A. Fonction logistique (Sigmoid) : Il s'agit de l'une des fonctions les plus fréquemment utilisées. Elle est communément appelée fonction logistique ou sigmoïde logistique, et elle est limitée entre 0 et 1. Elle peut être comprise de manière stochastique comme la probabilité que le neurone s'active. Elle est définie comme suit :

$$\sigma(z) = \frac{1}{1+e^{-z}} (1)$$

B. Tangente hyperbolique (TanH) : La fonction TanH est une fonction trigonométrique dont la forme d'onde lui confère le privilège d'être choisie parmi les fonctions d'activation. Elle est similaire à la fonction sigmoïde logistique, mais sa plage de valeurs est entre 1 et -1. Après différenciation, la valeur de cette fonction devient inférieure à 1. Elle est définie comme suit :

$$\sigma(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}} (2)$$

- La relation entre la fonction TanH et la fonction Sigmoid est :

$$(z) = 2\ s(2z) - 2 \quad (3)$$

C. Rectified Linear Unit (ReLU): La fonction ReLU est probablement celle qui est la plus similaire à son homologue biologique. Récemment, de nombreux emplois (en particulier ceux liés à la vision par ordinateur) ont commencé à privilégier cette fonction. Comme dans la formule ci-dessous, cette fonction renvoie 0 si l'entrée z est inférieure à 0 et renvoie z lui-même si elle est supérieure à 0, comme illustré dans la Figure 07. Elle est définie comme suit :

$$(z) = (0, z) \quad (4)$$

D. Leaky-ReLU : Leaky Rectified Linear Unit est un type de fonction d'activation basé sur un ReLU qui tente de résoudre le problème connu sous le nom de ReLU mourant.

Une unité rectifiée linéaire traditionnelle ReLU(z) renvoie 0 lorsque $z \leq 0$. Le problème du ReLU mourant fait référence au fait que l'unité reste bloquée de cette manière, renvoyant toujours 0 pour toute entrée. Le Leaky ReLU vise à résoudre ce problème en renvoyant une petite valeur négative non nulle à la place de 0, comme suit :

$$LeakyReLUa\,(z) = (az, z) \quad (5)$$

E. UNITÉ LINÉAIRE D'ERREUR GAUSSIENNE (GELU) : Cette fonction résout la plupart des problèmes des fonctions d'activation précédentes et, plus important encore, évite le problème des gradients qui disparaissent. Elle fournit un gradient bien défini dans la zone négative et empêche les neurones de mourir ; de plus, elle fonctionne mieux dans les modèles de transformateurs [29]. Le GELU est une formule approximative :

$$\text{GELU}(z) = 0.5z(1+\tanh[\sqrt{2/\pi}\,(z + 0.044715z^3) \quad (6)$$

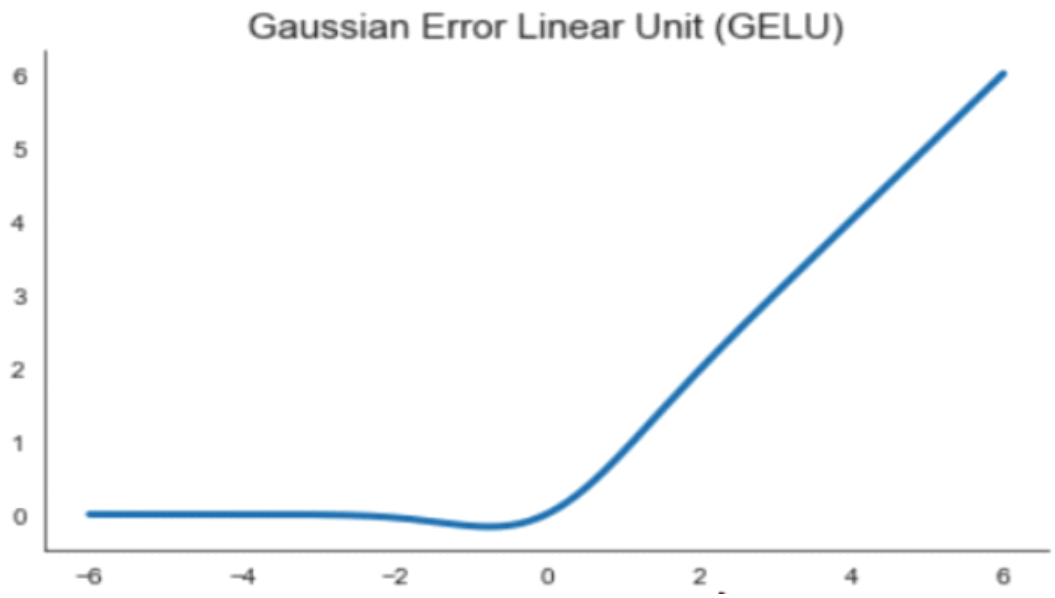

Figure 2.12 : GELU

7-Types de réseaux de neurones

7-1- réseau de neurones récurrents(RNNs)

Les réseaux de neurones récurrents (RNN) constituent une catégorie de réseaux neuronaux profonds dotés d'une mémoire interne, également appelée état de cellule, leur permettant d'analyser les caractéristiques d'entrée ainsi que les dépendances temporelles dans des données séquentielles, dans le but de prédire les sorties futures. Ainsi, la sortie de chaque neurone dépend non seulement de l'entrée actuelle, mais aussi des états cachés précédents, c'est-à-dire de l'historique du réseau [32].

La Figure 17 illustre la structure d'un RNN déplié dans le temps, avec sa boucle de rétroaction caractéristique. Selon le nombre d'étapes temporelles prises en compte, les RNN peuvent mémoriser efficacement des informations passées. Grâce à cette capacité, ils se sont révélés particulièrement performants dans des domaines tels que le traitement du langage naturel, la reconnaissance vocale ou encore la traduction automatique, où les entrées sont souvent sous forme de mots ou de caractères [32].

Par ailleurs, les RNN sont bien adaptés à la prévision de séries temporelles, comme l'estimation du State of Charge (SOC) et du State of Health (SOH), en fournissant des prédictions fondées à la fois sur l'entrée actuelle et l'historique des états précédents [30].

Toutefois, les RNN classiques ainsi que leurs variantes, telles que les LSTM (Long Short-Term Memory) et les GRU (Gated Recurrent Unit), sont sensibles au problème de

dépendance à long terme. Ce dernier provoque souvent une disparition ou une explosion du gradient lors de la rétropropagation. Cette limitation est en partie liée à l'utilisation de fonctions d'activation comme la sigmoïde ou la tangente hyperbolique, qui peuvent entraîner une décroissance progressive du gradient à travers les couches [32].

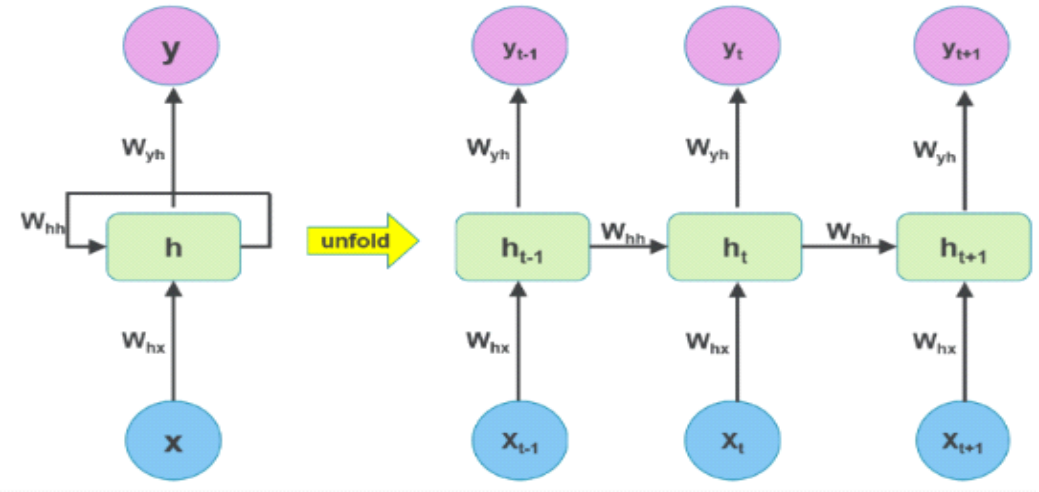

Figure 2.13 : Structure d'un réseau de neurones récurrents simple (RNN) et RNN déplié.

7-2- Réseaux de Neurones Convolutifs (CNN)

Le réseau de neurones convolutionnel (CNN) est une forme particulière de réseau de neurones à propagation avant, introduite en 1989 [33]. Il a d'abord été utilisé dans des domaines comme la vision par ordinateur et les systèmes de recommandation.

L'architecture d'un CNN se compose typiquement d'une succession de couches de convolution et de couches de regroupement (pooling), qui agissent conjointement pour extraire automatiquement les caractéristiques pertinentes d'une image. Ces couches sont généralement suivies par une ou plusieurs couches entièrement connectées, permettant la classification des images à partir des caractéristiques extraites (voir Figure 18).

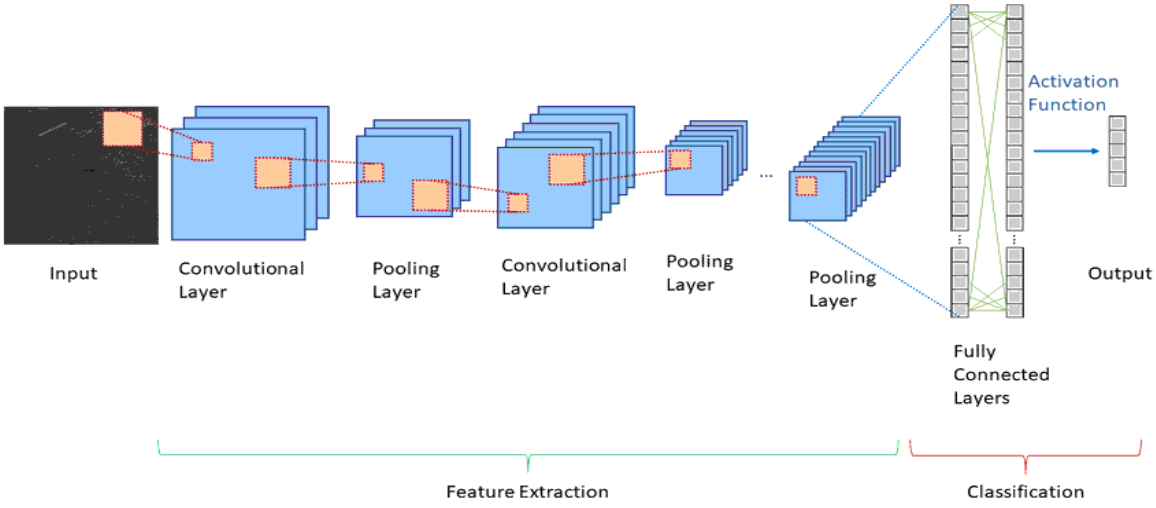

Figure 2.14 : Architecture typique d'un CNN.

Les différentes couches d'un réseau de neurone convolutif sont les suivantes :

A. Couche de convolution (CONV)

La convolution est une opération mathématique qui consiste à appliquer un filtre (ou noyau) sur un tableau à n dimensions, tel qu'une image. Ce filtre est lui-même représenté par une matrice de valeurs numériques, appelées poids ou paramètres.

Lors de son déplacement — ou convolution — sur l'image d'entrée, le filtre effectue une multiplication élément par élément entre ses propres valeurs et les pixels correspondants de l'image. Le résultat de ces multiplications est ensuite sommé pour générer un seul point de la carte de caractéristiques (ou carte d'activation). En parcourant ainsi toute l'image, le filtre produit une nouvelle représentation filtrée.

Il est important de noter que plusieurs filtres sont généralement appliqués à la même image, chacun apprenant un ensemble spécifique de poids pendant l'entraînement. Ces filtres permettent d'extraire différentes caractéristiques (bords, textures, motifs, etc.). Enfin, la couche de convolution est souvent suivie d'une fonction d'activation, telle que la ReLU, qui introduit de la non-linéarité dans le modèle.

Pour mieux comprendre le fonctionnement de la couche de convolution, voici un exemple simple. Nous appliquons un filtre à une image de 5×5 pixels.

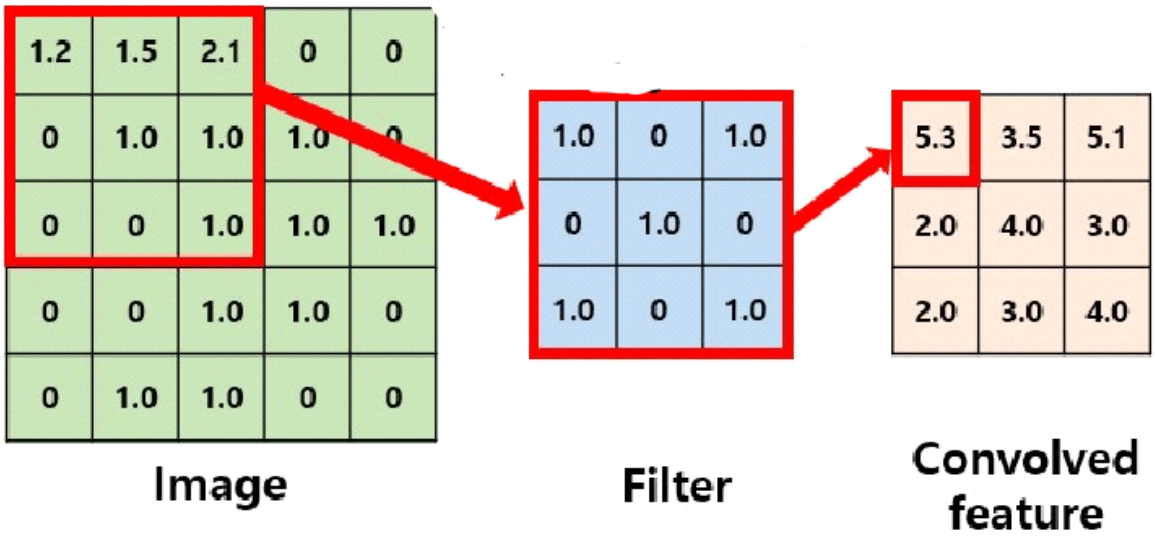

Figure 2.15 : Explication de l'opération de convolution.

B. Couche de regroupement (Pooling)

Le pooling est une opération qui consiste à résumer des régions locales d'une carte de caractéristiques pour produire une nouvelle carte de plus petite dimension, appelée carte de caractéristiques échantillonnée (ou poolée).

Concrètement, cette opération fait glisser une fenêtre (ou noyau) sur la carte d'entrée et applique une fonction de réduction, comme le maximum (max pooling) ou la moyenne (average pooling), à chaque sous-région couverte.

Les opérations de pooling jouent un rôle essentiel dans les réseaux neuronaux convolutionnels (CNN), car elles permettent de réduire la taille des données, de diminuer la complexité computationnelle et d'améliorer la robustesse du modèle face aux variations mineures dans l'image d'entrée.

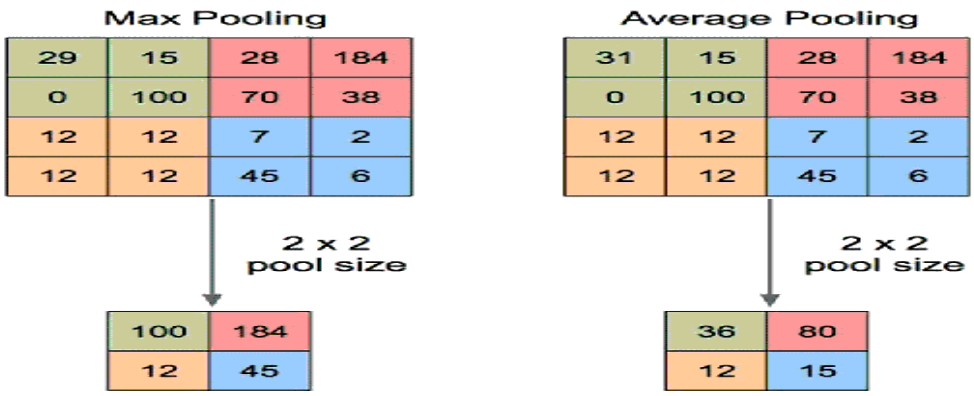

Figure 2.16 : Exemple de couche de regroupement.

C. Couche entièrement connectée (Fully Connected)

Il s'agit de la dernière couche d'un réseau de neurones convolutionnel, chargée de réaliser une classification précise en combinant l'ensemble des caractéristiques extraites par les couches précédentes à partir des données d'entrée.

À cette étape, les cartes de caractéristiques en deux dimensions sont aplaties pour former un vecteur de caractéristiques unidimensionnel (1D), un processus appelé flattening.

Les couches entièrement connectées fonctionnent de manière similaire aux couches d'un réseau de neurones classique, et ont pour objectif de générer des scores de classe à partir des activations précédentes, permettant ainsi la classification finale [34].

8-Conclusion

Ce chapitre a permis de poser les bases théoriques indispensables à la compréhension des techniques d'intelligence artificielle, en mettant l'accent sur l'apprentissage automatique et ses diverses déclinaisons. Nous avons exploré les principaux types d'apprentissage, allant du supervisé au non supervisé, en passant par les approches semi-supervisées et par renforcement. Les algorithmes classiques tels que les SVM, les arbres de décision et les forêts aléatoires ont été présentés, avant d'aborder les concepts avancés de l'apprentissage profond, notamment à travers les réseaux neuronaux artificiels et convolutifs. Ces notions constituent le socle fondamental sur lequel repose notre travail de recherche, et seront appliquées dans les chapitres suivants pour le développement, l'évaluation et l'optimisation de modèles intelligents dédiés à la classification automatique des données complexes.

Chapitre 03 : Modèles d’apprentissage automatique en classification d’IRM avec intégration

1-Introduction

Le choix des modèles d'apprentissage en imagerie médicale est crucial. En effet, tout système de classification d'IRM à but d'orientation diagnostic doit assurer à la fois une fiabilité de classification et un temps d'exécution raisonnable. Il est aussi nécessaire que les modèles utilisés peuvent être entrainés avec les données d'apprentissage dont nous disposons et qui ne sont pas forcément volumineuses. Selon cette problématique, je me suis intéressé à deux modèles bien connus pour leur performance, à savoir les arbres de décisions qui ne sont pas gourmant en données et en temps d'apprentissage et les réseaux de neurones convolutifs, qui sont connus pour leur adéquation aux données images, mais qui nécessitent des volumes importants de données d'apprentissage. Je mène également une étude d'intégration de ces deux modèles dans le but de mutualiser leurs avantages et minimiser leurs inconvénients.

2-Aperçu du système

2-1- Principe du fonctionnement

Le système de classification proposé a pour objectif de distinguer automatiquement les images d'IRM cérébrales en deux classes : normal et tumeur. Il repose sur une approche hybride combinant deux types de modèles complémentaires : un Arbre de Décision (DT), performant sur des données tabulaires simples, et un réseau de neurones convolutif (CNN) de type AlexNet, capable d'extraire des caractéristiques visuelles complexes. Chaque modèle est entraîné indépendamment à partir des mêmes images, avec un prétraitement adapté à ses spécificités : les images sont converties en niveaux de gris et réduites en vecteurs 1D pour le DT, tandis qu'elles sont conservées en couleur et redimensionnées à 227x227 pixels pour le CNN. Par la suite, une technique d'intégration permet de combiner les prédictions issues des deux modèles, dans le but d'exploiter au mieux leurs forces respectives. Cette stratégie permet d'obtenir un système de classification plus précis, robuste, et fiable, comme le démontrent les métriques d'évaluation utilisées tel que, la précision.

2-2-Organigramme

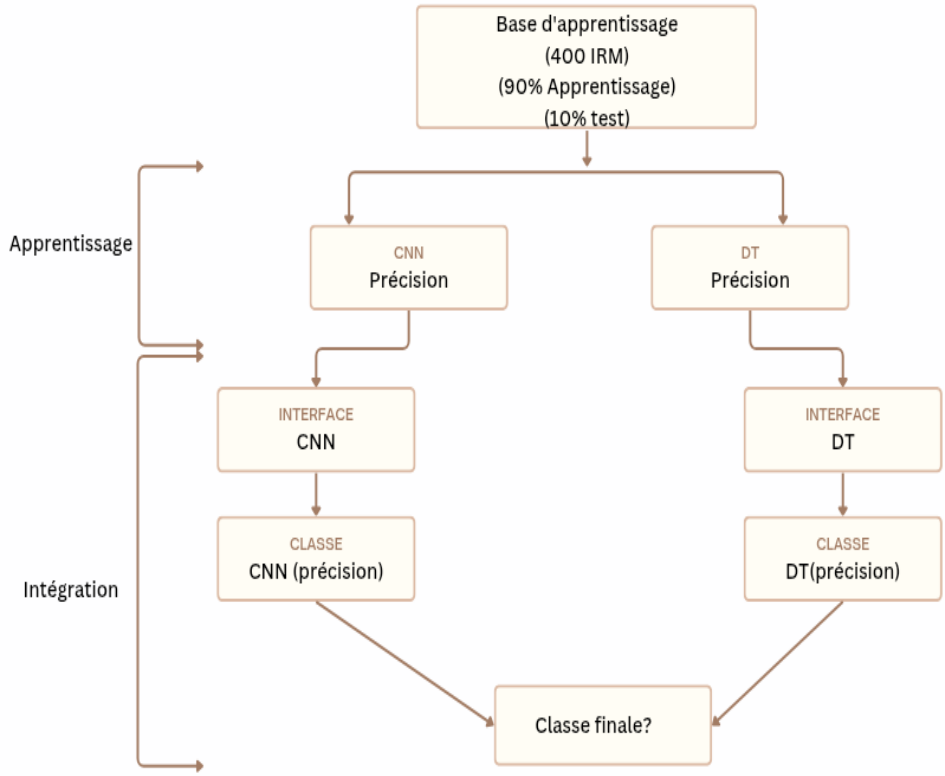

Figure 3.1 : Organigramme du système

3-Modèles IA utilisées

3-1- Les réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs (CNN) sont une technique d'apprentissage profond composée de plusieurs couches, chacune représentant une carte de caractéristiques. En image rie, ces cartes sont des matrices 3D d'intensités de pixels pour différents canaux de couleur. Chaque couche interne représente une image induite à plusieurs canaux, où chaque "pixel" est une caractéristique spécifique. Les neurones sont connectés à une petite partie des neurones ad jacents de la couche précédente, et des transformations telles que la convolution et la mise en commun peuvent être effectuées sur les cartes. Dans la phase de convolution, une opération convolue une matrice de filtre (poids appris) avec les valeurs d'un champ récepteur

de neurones, suivie d'une fonction non linéaire comme la sigmoïde ou ReLU pour obtenir les réponses. Ensuite, l'opération de mise en commun ré sume les réponses d'un champ récepteur en une seule valeur, produisant des descriptions de fonctionnalités plus robustes [35]. Une hiérarchie initiale des caractéristiques est construite, pouvant être affinée de manière supervisée en ajoutant une couche de neurones entièrement connectés pour s'adapter à différentes tâches. Selon la tâche, une couche finale avec différentes fonctions d'activation est ajoutée pour obtenir une probabilité spécifique pour chaque neurone de sortie. Le réseau est ensuite optimisé sur une fonction objective via la descente de gradient stochastique. Un exemple typique de CNN est le VGG16, composé de 13 couches convolutionnelles, 3 couches entièrement connectées, 3 couches de max-pooling et une couche finale de classifi cation softmax. Les filtres de convolution sont produits par des fenêtres de filtre 3x3, et les résolutions des cartes sont réduites avec 2 couches de max-pooling. Pour tester le modèle, une image de la même taille que les échantillons d'apprentissage peut être traitée, avec éventuelle ment une opération de remise à l'échelle ou de recadrage si différentes tailles sont fournies [36].

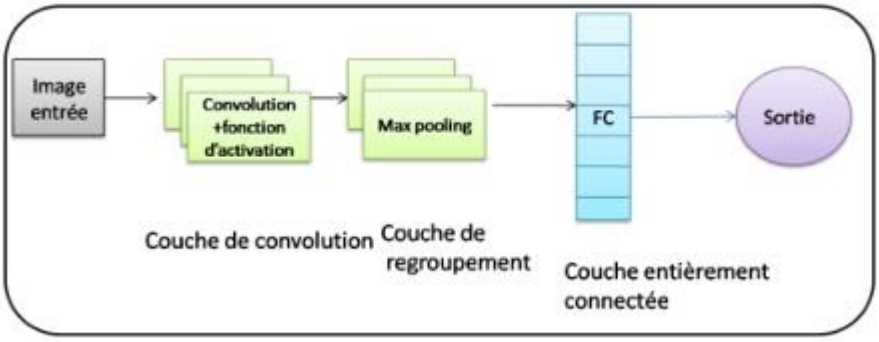

Figure 3.2 : Une architecture typique d'un CNN

3-1-1-Les avantages de CNN

Les avantages des réseaux de neurones convolutifs (CNN) par rapport aux méthodes de classification traditionnelles sont les suivants :

1-**Représentation hiérarchique des caractéristiques** : Les réseaux de neurones convolutifs (CNN) construisent une représentation hiérarchique des caractéristiques, en partant des pixels bruts jusqu'aux attributs sémantiques de haut niveau, à travers une série de couches successives [37].

2-**Optimisation conjointe de multiples tâches** : L'architecture des CNN permet de traiter simultanément plusieurs tâches liées. Par exemple, des modèles tels que *Fast R-CNN*

combinent à la fois la classification des objets et la régression des boîtes englobantes dans un cadre d'apprentissage multitâches.

3-**Grande capacité d'apprentissage** : Les CNN possèdent une forte capacité à apprendre, ce qui permet de reformuler certains problèmes classiques de la vision par ordinateur en tâches de transformation de données à haute dimension, et de les aborder sous un angle différent.

Grâce à ces atouts, les CNN sont largement utilisés dans divers domaines de recherche, notamment la super-résolution d'images, la reconstruction et la classification d'images, la recherche d'images, la reconnaissance faciale, ainsi que la détection et l'analyse vidéo [38].

3-1-2-Les inconvénients de CNN

Malgré les progrès significatifs par rapport aux méthodes traditionnelles et l'importance des réseaux de neurones convolutifs (CNN) dans la détection pratique d'objets, il existe encore quelques inconvénients :

1-**Contrainte sur la taille de l'image d'entrée** : En raison de la présence de couches entièrement connectées, les CNN requièrent des images d'entrée de dimensions fixes. Cela peut entraîner la nécessité de recalculer le modèle pour chaque région d'intérêt, allongeant ainsi considérablement le temps d'inférence.

2-**Complexité de l'apprentissage** : L'apprentissage du modèle ALEXNET (Deep Convolutional Neural Network) se déroule en plusieurs étapes, ce qui le rend coûteux en temps et en ressources. Il comprend le raffinement du réseau convolutif sur les propositions d'objets, le remplacement du classifieur *softmax* par des SVM (Support Vector Machines) pour mieux exploiter les caractéristiques extraites, ainsi que l'entraînement des régresseurs de boîtes englobantes. Ce processus est long et exige une capacité de stockage importante pour conserver les caractéristiques extraites.

3-**Redondance des propositions de régions** : Bien que les méthodes comme la recherche sélective puissent générer un grand nombre de propositions de régions avec un taux de rappel élevé, ces propositions restent souvent redondantes et leur génération peut être chronophage.

Malgré ces limites, les réseaux de neurones convolutifs demeurent très utilisés et activement étudiés en raison de leurs performances remarquables dans de nombreux domaines d'application.

3-1-3-L'opération de convolution

L'opération de convolution est comparable aux filtres de convolution que l'on trouve dans le traitement d'images, comme les filtres de Sobel, de Roberts et gaussien. Le noyau du filtre parcourt l'image, multipliant ses poids avec chaque pixel correspondant à travers ce filtre, puis ajoutant le biais au résultat [39].

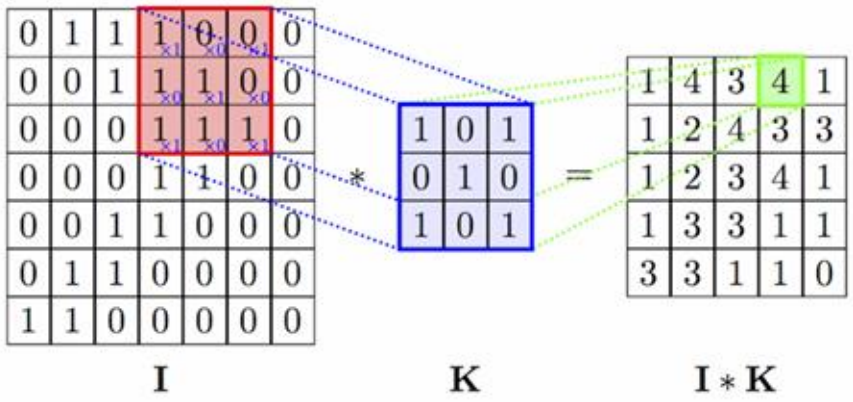

Figure 3.3 : Exemple d'une opération de convolution [40]

Dans la couche de convolution, trois hyperparamètres sont utilisés pour produire la carte des caractéristiques :

1. Pas (Stride S) : Cela représente le déplacement à chaque fois que le filtre de convolution est glissé. Un pas de 1 signifie que les filtres se déplacent d'un pixel à chaque fois. Avec un pas de 2 (ou plus), les filtres sautent 2 pixels à la fois pendant le glissement [41].

2. Rembourrage (Padding P) : Les images d'entrée peuvent être remplies avec une bor dure de taille spécifique en utilisant le rembourrage. Le rembourrage zéro (zéro-padding) est le plus couramment utilisé. La taille du rembourrage est un hyperparamètre contrôlant la taille spatiale des volumes de sortie dans les frameworks de réseaux de neurones [42].

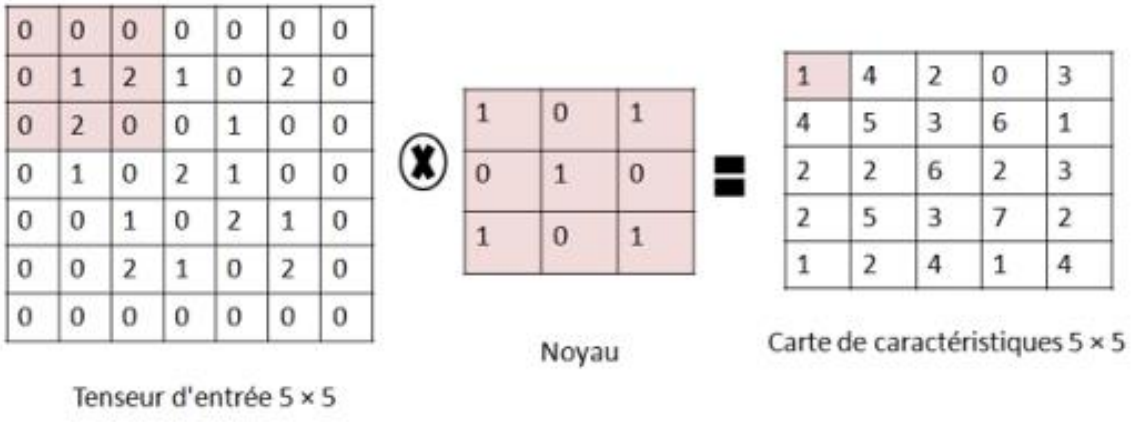

Figure 3.4: Exemple d'opération de convolution avec padding, un tenseur d'entrée de taille 5×5, noyau de 3×3 et un stride de 1. [43]

3. Profondeur (Depth D) : C'est le nombre de filtres utilisés dans une couche de convo lution, correspondant à la profondeur du volume de sortie. Ce paramètre est également un hyperparamètre et est déterminé par le nombre de filtres que nous choisissons d'appliquer [44].

3-1-4-Mise en commun (Pooling)

De nos jours, les réseaux de neurones à convolution utilisent toujours un partage de poids étendu pour réduire le nombre de paramètres des modèles. Une couche de mise en commun (pooling) est souvent utilisée pour réduire le temps de calcul et construire progressivement une invariance spatiale et configurative dans le processus d'apprentissage. Lacouchedemiseencommunaideàextrairedavantagedesenssémantiquedesimages.Par exemple, la mise en commun maximale (Max-pooling) renvoie simplement la valeur maximale des valeurs sur lesquelles l'opération du noyau est appliquée. Voici un exemple pour illustrer les sorties d'une mise en commun maximale (max-pooling) et d'une mise en commun moyenne (average pooling), respectivement, avec un noyau de taille 2 et un pas de 2[45].

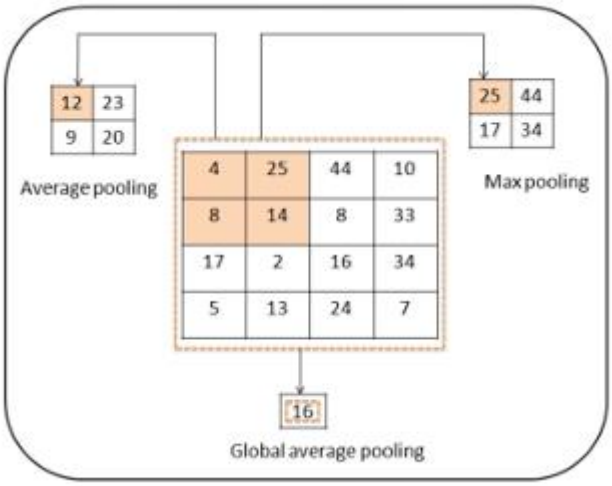

Figure 3.5 : Types d'opération de pooling

3-1-5-Aplanissement (Flattening)

Pour exploiter pleinement les caractéristiques de haut niveau extraites par la couche de convolution, une couche entièrement connectée est ajoutée. Cette couche est cruciale car elle permet de combiner de manière non linéaire les caractéristiques de haut niveau représentées par la sortie de la couche de convolution. En aplatissant les images en un vecteur (colonne), nous les convertissons dans une forme appropriée pour notre Perceptron multi-couches. Cette sortie aplatie est ensuite alimentée à un réseau de neurones feed-forward, et la rétropropagation est appliquée à chaque itération du processus d'apprentissage. Après plusieurs itérations, le modèle est capable de distinguer les caractéristiques dominantes et certaines caractéristiques de bas niveau dans les images, ce qui permet de les classer à l'aide de la technique de classification des réseaux de neurones [46].

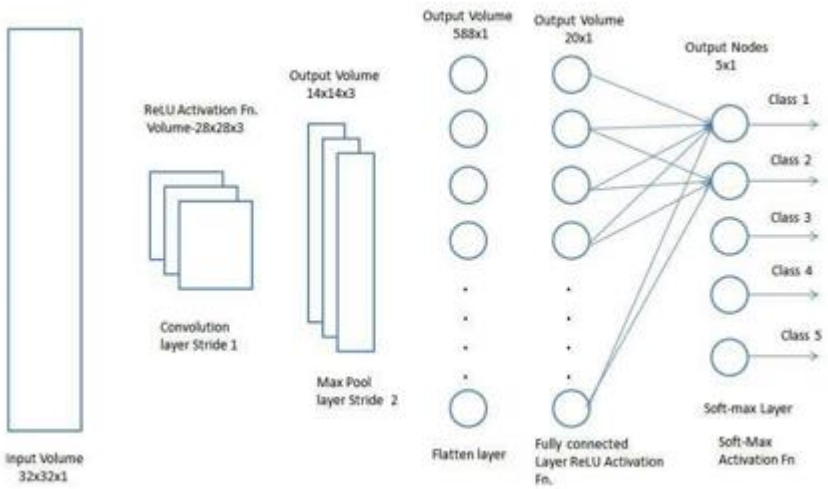

Figure 3.6 : schéma d'un réseau à convolution avec la couche Flattening

3-1-6-Deep Convolutional Neural Networks (AlexNet)

A-Définition

AlexNet est un réseau de neurones convolutif (CNN) introduit en 2012. Il a été conçu pour reconnaître des images et a été utilisé dans de nombreux domaines, notamment la robotique, l'imagerie médicale et les véhicules autonomes.

AlexNet a été créé pour traiter efficacement des tâches complexes de reconnaissance d'images.

Par exemple, imaginez qu'on enseigne à un enfant à reconnaître des objets comme des pommes et des oranges. Au début, on lui montre de nombreuses images de ces objets.

Avant AlexNet, il fallait décrire manuellement les caractéristiques de chaque objet à « l'enfant-machine ». AlexNet a changé cela en permettant à la machine d'apprendre elle-même ces caractéristiques, de la même manière qu'un enfant apprend [47].

B-Contexte et historique

Avant l'apparition d'AlexNet, l'ingénierie des caractéristiques (feature engineering) constituait une étape essentielle des techniques d'apprentissage automatique. Les ingénieurs concevaient manuellement les caractéristiques que les algorithmes devaient utiliser pour identifier des motifs. Ce processus était long, sujet aux erreurs et difficile à faire évoluer à grande échelle. En raison des limitations en matière de données et de puissance de calcul, l'idée d'un apprentissage profond (deep learning) basé sur des réseaux de neurones artificiels pour apprendre automatiquement les caractéristiques n'était pas nouvelle, mais elle restait spéculative et peu exploitée en pratique [47].

Tout a changé en 2009 avec le lancement du jeu de données ImageNet, développé par Fei-Fei Li et son équipe. ImageNet a offert une immense collection d'images annotées couvrant 1 000 catégories, créant ainsi un environnement idéal pour tester les modèles d'apprentissage automatique. Cependant, l'échelle et la complexité de ce jeu de données dépassaient les capacités des méthodes traditionnelles [47].

En 2012, trois chercheurs , Alex Krizhevsky, Geoffrey Hinton et Ilya Sutskever — ont présenté le modèle AlexNet, inspiré du système visuel humain. En utilisant des réseaux de neurones convolutifs (CNN) et en exploitant la puissance de calcul des GPU, AlexNet a démontré que les machines pouvaient atteindre une performance surhumaine en classification d'images, à condition d'être conçues de manière efficace [47].

C-Architecture d'AlexNet

Analysons la conception et le fonctionnement d'AlexNet pour découvrir ce qui l'a rendu révolutionnaire.

AlexNet comporte huit couches : trois couches entièrement connectées pour la classification et cinq couches convolutives pour l'extraction des caractéristiques. Chaque couche convertit progressivement l'image d'entrée en prédictions pertinentes [47].

L'activation ReLU, la régularisation par dropout et le pooling avec chevauchement (overlapping pooling) font partie des innovations qui distinguent AlexNet et qui ont permis de corriger les faiblesses des modèles précédents [47].

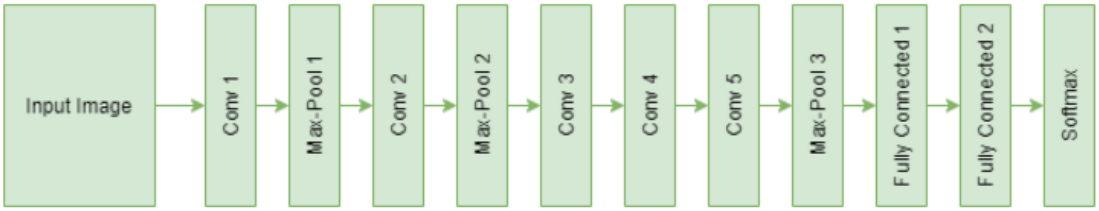

Figure 3.7 : exemple architecture d'alexnet

D-Explication couche par couche

- **Couche d'entrée (Input Layer) :**

Taille : AlexNet utilisait des images RGB (couleur) de dimensions 227×227×3.

Prétraitement : Afin d'améliorer l'extraction des caractéristiques et de réduire la redondance, les images étaient normalisées.

- **Couches convolutives (Convolutional Layers) :**

Ces couches utilisaient des filtres (kernels) pour extraire des éléments visuels tels que les formes, les contours et les textures.

Les couches profondes se concentraient sur des détails fins, tandis que les couches initiales capturaient des motifs plus larges.

De grands filtres (11×11) ont été utilisés dans la première couche convolutive d'AlexNet, un choix audacieux pour capter des informations globales.

Activation ReLU (Rectified Linear Unit) :

AlexNet employait la fonction d'activation ReLU, qui introduisait de la non-linéarité et accélérait l'apprentissage, contrairement aux fonctions traditionnelles comme le sigmoïde ou le tanh.

ReLU a joué un rôle crucial dans la possibilité d'entraîner des réseaux profonds, en atténuant le problème du gradient qui disparaît.

- **Couches de Pooling :**

Pour réduire les dimensions spatiales tout en conservant les caractéristiques essentielles, AlexNet utilisait des couches de max-pooling avec recouvrement (overlapping pooling).

Ce type de pooling apportait un effet de régularisation léger, améliorant la généralisation du modèle et réduisant le surapprentissage (overfitting).

Couches entièrement connectées (Fully Connected Layers) :

Les dernières couches servaient à convertir les caractéristiques de haut niveau en probabilités de classes.

Ces couches fortement connectées permettaient d'identifier des schémas complexes dans les données.

- **Couche de sortie (Output Layer) :**

Un algorithme softmax générait des probabilités pour les 1000 classes d'ImageNet.

La prédiction finale correspondait à la classe ayant la probabilité la plus élevée.

En combinant ces différents composants, AlexNet a surpassé ses concurrents lors de la compétition ImageNet, atteignant un taux d'erreur top-5 record de 15,3 %.

E-Innovations clés et leur importance

L'apprentissage profond moderne a été rendu possible grâce aux innovations révolutionnaires suivantes :

Activation ReLU : Contrairement aux fonctions sigmoïde ou tanh, l'introduction de la ReLU a permis d'ajouter de la non-linéarité, de résoudre le problème du gradient qui disparaît et d'accélérer considérablement l'apprentissage.

Dropout : L'un des principaux problèmes des réseaux profonds était le surapprentissage (overfitting). Le Dropout, une technique utilisée par AlexNet consistant à désactiver aléatoirement des neurones pendant l'entraînement, a permis d'améliorer la généralisation et d'ajouter un mécanisme de régularisation.

Accélération par GPU : En raison des limitations des processeurs (CPU), il aurait été impossible d'entraîner AlexNet sur le vaste jeu de données ImageNet. Toutefois, en utilisant des GPU pour le traitement parallèle, AlexNet a considérablement réduit le temps d'entraînement et démontré l'importance du matériel dans l'intelligence artificielle.

Pooling avec chevauchement (Overlapping Pooling) : Contrairement au pooling classique, le pooling avec chevauchement a introduit un léger effet de régularisation, ce qui a amélioré la capacité du modèle à se généraliser à de nouvelles données.

Augmentation de données (Data Augmentation) : Pour renforcer la robustesse et réduire le surapprentissage, AlexNet a artificiellement élargi le jeu de données en utilisant des techniques telles que le retournement (flipping) et le rognage aléatoire (random cropping).

Ces avancées ont élargi les possibilités de l'apprentissage profond, dépassant largement de simples progrès technologiques.

F-Défis rencontrés par AlexNet

Bien qu'AlexNet ait connu un succès remarquable, plusieurs défis ont été identifiés :

Exigences computationnelles : L'entraînement d'AlexNet nécessitait des GPU haut de gamme, coûteux et inaccessibles pour de nombreux chercheurs à l'époque.

Surapprentissage (Overfitting) : Bien que l'augmentation de données et le dropout aient permis de réduire le surapprentissage, la généralisation sur de petits jeux de données restait difficile.

Dépendance aux données : L'importance de la curation et de l'annotation des données était souvent sous-estimée, alors que le succès d'AlexNet dépendait fortement de jeux de données volumineux et annotés, comme ImageNet.

Consommation énergétique : L'énergie computationnelle requise pour l'entraînement de modèles massifs comme AlexNet a soulevé des préoccupations liées à la scalabilité et à l'impact environnemental.

Interprétabilité : Malgré ses performances, AlexNet a mis en évidence un problème fondamental de l'apprentissage profond : la compréhension des décisions prises par les modèles.

Cela a donné naissance à de nombreux travaux sur l'intelligence artificielle explicable (Explainable AI).

Ces difficultés ont souligné la nécessité d'une innovation continue afin d'améliorer l'accessibilité, l'efficacité et la compréhension des réseaux de neurones profonds.

L'héritage d'AlexNet

La victoire d'AlexNet au concours ImageNet de 2012 a constitué un tournant majeur dans l'histoire de l'intelligence artificielle.

Son architecture novatrice, incluant l'accélération par GPU, le dropout et l'activation ReLU, a permis de réduire l'écart entre la recherche théorique et les applications concrètes.

AlexNet n'a pas seulement remporté une compétition : il a posé les bases de la révolution de l'IA.

Les méthodes qu'il a introduites continuent d'influencer le deep learning moderne, que ce soit pour les modèles génératifs, la reconnaissance d'objets, ou d'autres domaines.

3-2-Les arbres de décision

3-2-1-Définition

Un arbre de décisions est un algorithme d'apprentissage supervisé non paramétrique, utilisé à la fois pour les tâches de classification et de régression. Il possède une structure hiérarchique et arborescente, qui se compose d'un nœud racine, de branches, de nœuds internes et de nœuds feuille [48].

3-2-2-Principe de Fonctionnement

Le fonctionnement des arbres de décision repose sur la **division récursive** des données en sous-ensembles homogènes.

- **Construction de l'arbre**

1. **Choix de la meilleure caractéristique** :
   À chaque étape, le modèle cherche la variable qui permet de **mieux séparer les classes cibles**. Cette sélection repose sur un **critère d'impureté** comme :

- **Indice de Gini** : $G=1-\sum p_i^2$
- **Entropie** : $H=-\sum p_i \log_2 p_i$

  Ces critères mesurent le degré de "pureté" des données : une impureté nulle signifie que toutes les données appartiennent à une même classe.

2. **Division récursive** :

   Une fois la meilleure variable et le meilleur seuil trouvés, le jeu de données est divisé en deux sous-groupes. Le processus se répète **récursivement** sur chacun de ces sous-groupes.

3. **Arrêt de la croissance de l'arbre** :

   La construction s'arrête si :

   - Toutes les données sont dans la même classe.
   - Un seuil maximal de profondeur est atteint.
   - Le nombre de données restantes est insuffisant.

- **Avantages et inconvénients des arbres de décision**

Bien que les arbres de décision puissent être utilisés dans de nombreux cas d'utilisation, d'autres algorithmes sont généralement plus performants. Cela dit, les arbres de décision sont particulièrement utiles pour les tâches de fouille des données et de découverte de connaissances. Découvrons ci-dessous les principaux avantages et les principaux défis de l'utilisation des arbres de décision : Bien que les arbres de décision puissent être utilisés dans de nombreux cas d'utilisation, d'autres algorithmes sont généralement plus performants. Cela dit, les arbres de décision sont particulièrement utiles pour les tâches de fouille des données et de découverte de connaissances. Découvrons ci-dessous les principaux avantages et les principaux défis de l'utilisation des arbres de décision :

A-Avantages

Facilité d'interprétation : la logique booléenne et les représentations visuelles des arbres de décision facilitent leur compréhension et leur utilisation. La nature hiérarchique d'un arbre de décision permet également de voir facilement quels sont les attributs les plus importants, ce qui n'est pas toujours évident avec d'autres algorithmes, comme les réseaux neuronaux.

Peu voire aucune préparation des données requise : les arbres de décision présentent un certain nombre de caractéristiques qui les rendent plus flexibles que les autres classificateurs.

Ils peuvent gérer différents types de données, par exemple les valeurs discrètes ou continues et les valeurs continues peuvent être converties en valeurs catégorielles grâce à l'utilisation de seuils. De plus, ils peuvent gérer des valeurs avec des valeurs manquantes, ce qui peut être problématique pour d'autres classificateurs, comme la méthode de classification naïve bayésienne.

Plus de flexibilité : les arbres de décision peuvent être utilisés à la fois pour les tâches de classification et de régression, ce qui les rend plus flexibles que certains autres algorithmes. Ils sont également insensibles aux relations sous-jacentes entre les attributs ; cela signifie que si deux variables sont hautement corrélées, l'algorithme ne choisira que l'une des fonctionnalités sur lesquelles se baser.

B-Inconvénients

Sujets au surajustement : les arbres de décision complexes ont tendance à être surdimensionnés et ne se généralisent pas correctement aux nouvelles données. Ce scénario peut être évité grâce aux processus de pré-élagage ou de post-élagage. Le pré-élagage arrête la croissance de l'arbre en cas de données insuffisantes, tandis que le post-élagage supprime les branches dont les données sont inadéquates après la construction de l'arbre.

Estimateurs de variance élevés : de petites variations dans les données peuvent produire un arbre de décision très différent. Le bagging ou moyenne des estimations peut être une méthode permettant de réduire la variance des arbres de décision. Cependant, cette approche est limitée car elle peut conduire à des prédicteurs hautement corrélés.

Coût plus élevé : étant donné que les arbres de décision induisent une phase de recherche exigeante pendant la construction, leur formation peut être plus coûteuse que d'autres algorithmes.

3-2-3-Le modèle DecisionTreeClassifier

Le modèle choisi pour la classification des données dans cette partie du travail est le DecisionTreeClassifier, qui fait partie des algorithmes d'apprentissage supervisé les plus utilisés en apprentissage automatique. Ce modèle appartient à la catégorie des arbres de décision, une méthode basée sur des règles logiques hiérarchiques permettant de prendre des décisions à partir de caractéristiques observées dans les données.

Le DecisionTreeClassifier est disponible dans la bibliothèque **Scikit-learn,** une référence en matière de machine learning en Python. Ce modèle est capable de gérer à la fois des variables numériques et catégorielles, et il peut être utilisé pour résoudre des problèmes de classification binaire ou multiclasse.

Le principe du modèle repose sur une structure en arborescence, dans laquelle :

Chaque nœud interne représente un test sur une caractéristique

Chaque branche correspond à une issue possible de ce test (oui ou non),

Chaque feuille contient une classe de sortie (la prédiction finale).

Lors de l'apprentissage, le modèle explore les meilleures divisions possibles des données à chaque étape, en se basant sur des mesures d'impureté, telles que :

L'indice de Gini

L'entropie de Shannon

Ces mesures permettent d'évaluer le degré d'homogénéité des sous-groupes créés. Le modèle cherche à minimiser cette impureté pour obtenir des groupes de données les plus purs possibles (idéalement appartenant à une seule classe).

L'arbre est construit de manière récursive, en divisant le jeu de données en sous-groupes de plus en plus homogènes, jusqu'à ce qu'un critère d'arrêt soit atteint (par exemple : profondeur maximale atteinte, ou nombre minimal d'échantillons dans un nœud).

Le modèle DecisionTreeClassifier présente plusieurs avantages : il est simple à interpréter, rapide à entraîner, et ne nécessite pas de mise à l'échelle des données. Toutefois, il peut être sujet à un phénomène de surapprentissage (overfitting) s'il n'est pas bien régulé. C'est pourquoi il est souvent conseillé de contrôler sa profondeur maximale et d'autres paramètres pour éviter qu'il ne s'ajuste trop fortement aux données d'apprentissage.

Dans le cadre de ce travail, ce modèle a été sélectionné pour sa capacité à classer efficacement des images prétraitées, en tenant compte des caractéristiques extraites de manière simple (ex. :

niveaux de gris, intensités moyennes), tout en permettant une interprétation claire du processus de décision.

4-La stratégie d'intégration

Afin d'améliorer la performance globale du système de classification d'IRM, nous avons adopté une stratégie d'intégration reposant sur deux méthodes complémentaires. Ces approches permettent de combiner les prédictions de deux modèles : un Arbre de Décision (DT) et un réseau de neurones convolutif (CNN – AlexNet)

4-1-Intégration basée sur la précision (Vote pondéré)

Cette méthode repose sur l'utilisation de la précision globale de chaque modèle pour pondérer son influence dans la décision finale.

Chaque modèle se voit attribuer un poids proportionnel à sa précision sur les données de validation.

Lors de la prédiction, les scores ou classes prédites par chaque modèle sont pondérés par leur précision, et la classe finale correspond à celle ayant le score le plus élevé.

4-2-Intégration basée sur l'historique (par classe)

Cette méthode d'intégration repose sur l'exploitation des performances historiques détaillées de chaque modèle, non pas globalement, mais par classe (par exemple : tumeur et normal).

Contrairement aux approches classiques qui utilisent une précision globale, cette méthode analyse les taux de précision obtenus par chaque modèle pour chaque classe spécifiquement.

Pour chaque classe, on identifie quel modèle est historiquement le plus fiable pour la reconnaître correctement. Ainsi, lorsqu'une nouvelle image est testée, et que les deux modèles donnent des prédictions différentes, la décision finale est guidée par le modèle ayant historiquement mieux performé sur la classe prédite.

Cette stratégie permet une prise de décision fine et dynamique, qui exploite les forces spécifiques de chaque modèle pour chaque type de classe. Elle est particulièrement avantageuse dans les contextes déséquilibrés ou avec des classes difficiles à distinguer, car elle adapte la prédiction à la compétence historique démontrée.

5-Implémentation et tests

5-1-le langage python

1-Langage Python

Développé par Guido Van Rossum au début des années 1980, Python est un langage de programmation polyvalent utilisé dans de nombreux domaines tels que le développement web, l'intelligence artificielle, le machine learning, les systèmes d'exploitation, les applications mobiles, les jeux vidéo, et plus encore. Typé dynamiquement et successeur du langage de programmation ABC, Python est rapidement devenu populaire et adopté dans divers secteurs. Il a véritablement pris son essor au début des années 2000 avec la sortie de Python 2.0. Orienté objet, ce langage est idéal tant pour les projets de grande envergure que pour les programmes plus modestes [49].

Figure 3.8 **:** Logo Phyton

2-Keras

Keras est une bibliothèque open source conçue pour le prototypage rapide de modèles de deep learning. Accessible aux débutants en intelligence artificielle, elle offre une API de haut niveau qui prend en charge différentes bibliothèques de réseaux de neurones artificiels récurrents ou convolutifs, telles que TensorFlow, Microsoft Cognitive Toolkit, PlaidML et Theano [49]

Figure 3.9 : Logo Kears

3-Tensorflow

Tensorflow est une bibliothèque de machine learning, offrant une boîte à outils pour résoudre aisément des problèmes mathématiques extrêmement complexes. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimental et de les transformer en logiciels fonctionnels [50].

Figure 3.10 : Logo Tensorflow

4-OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque open-source très populaire en Python utilisée pour le traitement d'images, la vision par ordinateur et la reconnaissance de formes. Elle a été développée par Intel et est maintenant maintenue par la communauté open-source.

OpenCV fournit un large éventail de fonctionnalités et d'algorithmes pour travailler avec des images et des vidéos [51].

Figure 3.11 : Logo OpenCV

5-2-Google colab

Google-Colaboratory

Pendant de nombreuses années, Google a développé un outil de développement nommé Google Colaboratory, plus connu sous le nom de Google Colab. Aujourd'hui, cet environnement de programmation est mis à disposition gratuitement pour le public.

Google Colab est un notebook interactif entièrement basé sur le cloud, similaire à Google Docs, qui permet à plusieurs utilisateurs de collaborer et de modifier simultanément un document. Il prend en charge de nombreuses bibliothèques populaires de machine learning et d'intelligence artificielle, facilement intégrables dans les notebooks.

Grâce à Colab, il est possible, avec seulement quelques lignes de code, d'importer un jeu de données d'images, d'entraîner un classificateur, puis d'évaluer le modèle. Le code s'exécute directement sur les serveurs cloud de Google, offrant ainsi la possibilité d'utiliser des ressources matérielles puissantes comme les GPU et TPU, indépendamment de la performance de l'ordinateur de l'utilisateur.

Enfin, Google Colab prend en charge des langages de programmation tels que Python ou C++, facilitant ainsi le développement d'applications avancées en science des données et intelligence artificielle. [52].

Figure 3.12 : L`environnement Google Colaboratory

Pourquoi utiliser Google Colab ?

Il existe plusieurs raisons pour lesquelles Google Colab est préférable à une simple instance de Jupyter Notebook locale :

- **Bibliothèques préinstallées**

Alors que la distribution Anaconda de Jupyter Notebook propose déjà plusieurs bibliothèques de traitement de données telles que Pandas, NumPy ou Matplotlib, Google Colab va plus loin. Il intègre nativement de nombreuses bibliothèques de deep learning comme TensorFlow, Keras ou PyTorch, ce qui facilite le développement de projets d'intelligence artificielle sans avoir à installer manuellement ces outils.

- **Sauvegarde dans le Cloud**

Contrairement aux carnets Jupyter classiques qui enregistrent les fichiers en local, Google Colab enregistre automatiquement vos notebooks dans Google Drive. Cela permet d'y accéder facilement depuis n'importe quel appareil connecté à votre compte Google. Bien que le stockage local puisse convenir pour des raisons de confidentialité, le stockage cloud offre flexibilité et accessibilité.

- **Collaboration facilitée**

Un des grands atouts de Google Colab est sa fonction de collaboration en temps réel. À l'instar de Google Docs, plusieurs utilisateurs peuvent modifier simultanément un même notebook, ce qui est idéal pour les projets en équipe. De plus, le partage de vos travaux est simple, via un lien ou une invitation.

- **Accès gratuit aux GPU et TPU**

Google met à disposition des GPU et TPU gratuits pour les projets personnels en machine learning. Cette puissance de calcul permet d'accélérer considérablement l'entraînement des modèles, même pour des projets de petite taille. En utilisant l'infrastructure cloud de Google, les performances de votre machine locale ne sont plus une limitation, car le traitement se fait entièrement sur les serveurs de Google. [53].

5-3-Codes de modeles

Le dataset d'IRM

Afin de développer une application basée sur des modèles d'intelligence artificielle pour la classification d'images médicales, il est indispensable de disposer de deux ensembles de

données : l'un dédié à l'entraînement du modèle, et l'autre réservé au test pour évaluer les performances du système.

Dans notre cas, le dataset utilisé a été téléchargé depuis une plateforme de référence en science des données, Kaggle. Il s'agit d'un ensemble contenant des images IRM ""Brain_tumor_images", réparties en deux classes principales :

Tumor (présence de tumeur cérébrale)

Normal (absence de tumeur)

Le dataset contient environ 400 images, de dimensions variables, que nous avons prétraitées, normalisées et redimensionnées en fonction du modèle utilisé.

Le processus de classification est basé sur deux approches complémentaires :

-Un modèle Decision Tree, qui traite les images converties en niveaux de gris et aplaties

-Un modèle CNN (AlexNet), qui traite les images couleur redimensionnées à 227×227

Les résultats des deux modèles sont ensuite fusionnés grâce à la méthode du vote majoritaire, permettant d'augmenter la fiabilité des prédictions.

Voici les catégories utilisé dans notre base de données :

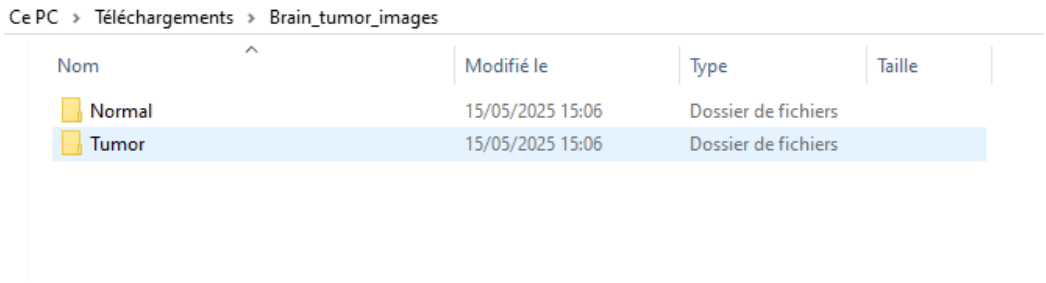

Figure 3.13 : Les 2 classes dans le data set.

Code Arbre de décision

- **Prétraitement**

```python
import os
import cv2
import numpy as np
data_dir = "/content/Brain_tumor_images/Brain_tumor_images"
categories = ["Normal", "Tumor"]

X = []
y = []

for label, category in enumerate(categories):
    folder = os.path.join(data_dir, category)
    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)
        try:
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
            img = cv2.resize(img, (64, 64))
            X.append(img.flatten())
            y.append(label)
        except:
            pass
X = np.array(X)
y = np.array(y)
print("✅ Images chargées :", X.shape)
```

Figure 3.14 **:** code prétraitement des images

Ce bloc de code a pour objectif le chargement et la préparation des images IRM pour l'entraînement d'un modèle d'arbre de décision. Le script parcourt les deux classes définies, à savoir "Normal" et "Tumor", contenues dans le dossier principal du dataset. Pour chaque image, les étapes suivantes sont réalisées :

Lecture en niveaux de gris (cv2.IMREAD_GRAYSCALE) pour simplifier les données et réduire la complexité de traitement.

Redimensionnement des images à une taille standard de 64×64 pixels afin d'uniformiser les entrées du modèle.

Aplatissement des images en un vecteur unidimensionnel (flattening) pour les rendre compatibles avec le format d'entrée requis par un algorithme d'arbre de décision (vecteurs de caractéristiques).

Étiquetage automatique des images selon leur classe (0 pour "Normal", 1 pour "Tumor").

- **Normalisation**

```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Figure 3.15 : code normalisation des images

Ce code permet de normaliser les données d'entrée afin de faciliter l'apprentissage du modèle et d'améliorer sa performance, en utilisant la méthode StandardScaler qui standardise les pixels des images.

- **La division**

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.10, random_state=42, stratify=y
)

print(" Taille du jeu d'entraînement :", X_train.shape)
print(" Taille du jeu de test        :", X_test.shape)
```

Figure 3.16 : code division des images

Ce code permet de diviser les données en deux ensembles : 90 % pour l'entraînement et 10 % pour le test, tout en conservant la proportion des classes grâce à l'option stratify.

*Entrainement

```python
from sklearn.tree import DecisionTreeClassifier

model_dt = DecisionTreeClassifier(
    criterion="entropy",
    max_depth=20,
    min_samples_split=5,
    random_state=42
)
model_dt.fit(X_train, y_train)
```

Figure 3.17 : code Entrainement du modèle DT

Ce code entraîne un modèle d'arbre de décision avec des paramètres optimisés pour améliorer la précision tout en limitant le surapprentissage.

- **Évaluation des performances du modèle**

```
from sklearn.metrics import accuracy_score, classification_report
import pandas as pd
y_pred = model_dt.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(" Précision du modèle :", round(accuracy * 100, 2), "%")
report = classification_report(y_test, y_pred, target_names=["Normal", "Tumor"], output_dict=True)
df_report = pd.DataFrame(report).transpose()
```

Figure 3.18 : code Évaluation du modèle DT

Ce code permet d'évaluer les performances du modèle en calculant la précision globale et en générant un rapport détaillé de classification.

Code CNN

- **Préparation des données pour CNN**

```
import os
import cv2
import numpy as np

data_dir = "/content/Brain_tumor_images/Brain_tumor_images"
categories = ["Normal", "Tumor"]

X = []
y = []
for label, category in enumerate(categories):
    folder = os.path.join(data_dir, category)
    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)
        try:
            img = cv2.imread(img_path)
            img = cv2.resize(img, (227, 227))
            X.append(img)
            y.append(label)
        except:
            pass
X = np.array(X) / 255.0
y = np.array(y)

print(" Données chargées :", X.shape)
```

Figure 3.19 : code préparation des données .

Ce bloc de code permet de charger les images du dataset en couleur, de les redimensionner au format 227x227 pixels, ce qui est compatible avec l'architecture AlexNet, puis de les

normaliser. Chaque image est associée à une étiquette correspondant à sa classe (Normal ou Tumor), afin de constituer les données d'entrée pour l'entraînement du modèle CNN.

- **La Division**

```
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
y_cat = to_categorical(y, num_classes=2)
X_train, X_test, y_train, y_test = train_test_split(
    X, y_cat, test_size=0.1, random_state=42, stratify=y_cat
)
print(" Données divisées :")
print("    X_train :", X_train.shape)
print("    X_test  :", X_test.shape)
print("    y_train :", y_train.shape)
print("    y_test  :", y_test.shape)
```

Figure 3.20 : code de la Division des images pour cnn

Ce code effectue la division du dataset en deux sous-ensembles : 90 % pour l'apprentissage et 10 % pour les tests. Les étiquettes sont encodées au format one-hot pour être compatibles avec le réseau de neurones convolutionnel.

- **Préparation des données d'entraînement avec augmentation d'images**

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(
    rotation_range=20,
    zoom_range=0.15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True
)
datagen.fit(X_train)
print(" Générateur d'images avec augmentation prêt.")
```

Figure 3.21 : code préparation des images pour cnn

Ce code utilise un générateur ImageDataGenerator pour appliquer des transformations aléatoires (rotations, zooms, décalages, retournements) afin d'augmenter la diversité des images d'entraînement et améliorer la robustesse du modèle CNN.

- **Conception du modèle AlexNet**

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout
model = Sequential()
model.add(Conv2D(filters=96, kernel_size=(11,11), strides=(4,4), activation='relu', input_shape=(227, 227, 3)))
model.add(MaxPooling2D(pool_size=(3,3), strides=(2,2)))
model.add(Conv2D(filters=256, kernel_size=(5,5), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(3,3), strides=(2,2)))
model.add(Conv2D(filters=384, kernel_size=(3,3), padding="same", activation='relu'))
model.add(Conv2D(filters=384, kernel_size=(3,3), padding="same", activation='relu'))
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation='relu'))
model.add(MaxPooling2D(pool_size=(3,3), strides=(2,2)))
model.add(Flatten())
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(2, activation='softmax'))
model.summary()
```

Figure 3.22 : code conception du modele cnn

Ce bloc de code définit une architecture AlexNet adaptée à la classification d'images médicales (IRM cérébrales) en deux classes : « Normal » et « Tumeur ».

Le modèle utilise plusieurs couches convolutives pour extraire des caractéristiques visuelles complexes, suivies de couches de pooling pour réduire la dimensionnalité.

Les couches Dense (fully connected) permettent d'apprendre des combinaisons non linéaires de caractéristiques extraites, avec des fonctions d'activation ReLU.

Un Dropout est appliqué pour éviter le surapprentissage.

Enfin, la couche de sortie utilise softmax pour donner une probabilité à chaque classe.

- **Compilation du modèle CNN (AlexNet)**

```
from tensorflow.keras.optimizers import Adam
model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Figure 3.23 : code Compilation du modèle CNN

Ce code compile le modèle en utilisant l'optimiseur Adam avec un taux d'apprentissage faible (0.0001) pour une convergence progressive et stable.

La fonction de perte choisie est **categorical_crossentropy**, appropriée pour un problème de classification binaire avec étiquettes encodées en one-hot.

L'évaluation du modèle se fait avec la métrique accuracy, permettant de suivre la performance au fil de l'entraînement.

- **Entraînement du modèle AlexNet avec augmentation et régularisation**

```
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
early_stopping = EarlyStopping(monitor='val_loss', patience=7, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min_lr=1e-6)
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=32),
    validation_data=(X_test, y_test),
    epochs=50,
    callbacks=[early_stopping, reduce_lr],
    verbose=1
)
```

Figure 3.24 : code Entraînement du modèle AlexNet

Ce bloc lance l'entraînement du modèle CNN (AlexNet) en utilisant les images d'entraînement augmentées dynamiquement via ImageDataGenerator.

Deux callbacks sont utilisés :

• **EarlyStopping** pour arrêter l'entraînement automatiquement si la performance de validation ne s'améliore plus (évite le surapprentissage).

• **ReduceLROnPlateau** pour réduire le taux d'apprentissage lorsque la fonction de perte de validation stagne (améliore la convergence).

L'entraînement est réalisé sur un maximum de 50 époques avec batch_size=32, et la validation est effectuée à chaque époque avec les données de test.

- **Évaluation des performances du modèle CNN (AlexNet)**

```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_test, axis=1)
accuracy = np.mean(y_pred_classes == y_true)
print(f" Précision globale du modèle : {accuracy * 100:.2f}%")
cm = confusion_matrix(y_true, y_pred_classes)
plt.figure(figsize=(6, 5))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=categories, yticklabels=categories)
plt.xlabel("Classe prédite")
plt.ylabel("Classe réelle")
plt.title("Matrice de confusion")
plt.show()
print(" Rapport de classification :")
print(classification_report(y_true, y_pred_classes, target_names=categories))
```

Figure 3.25 : code Évaluation des performances du modèle CNN

Ce bloc évalue la performance du modèle entraîné à l'aide de plusieurs métriques :

La précision globale est calculée en comparant les classes prédites (y_pred_classes) aux vraies étiquettes (y_true).

Une matrice de confusion est générée pour visualiser les erreurs de classification et vérifier si le modèle confond les classes "Normal" et "Tumor".

Le rapport de classification fournit des métriques détaillées comme la précision, le rappel et le F1-score pour chaque classe.

Ces outils sont indispensables pour juger la qualité du modèle, au-delà de la simple précision.

Code Intégration des deux modeles (DT+CNN)

- **Intégration basé précision**

Méthode d'intégration

```
from sklearn.metrics import accuracy_score
precision_dt = precision_score(y_true, y_pred_dt)
precision_cnn = precision_score(y_true, y_pred_cnn)
total = precision_dt + precision_cnn
weight_dt = precision_dt / total
weight_cnn = precision_cnn / total
y_pred_ensemble = []
for prob_cnn, pred_dt in zip(y_pred_cnn_probs, y_pred_dt):
    prob_class1_cnn = prob_cnn[1] if len(prob_cnn) > 1 else 0
    prob_class1_dt = float(pred_dt)
    score = weight_cnn * prob_class1_cnn + weight_dt * prob_class1_dt
    y_pred_ensemble.append(1 if score >= 0.5 else 0)
y_pred_vote = y_pred_ensemble.copy()
```

Figure 3.26 : code Méthode d'intégration basée précision

Cette étape permet de mesurer la performance globale du système d'intégration en combinant les prédictions du modèle Arbre de Décision et du modèle CNN (AlexNet). L'objectif est de vérifier si la fusion des deux modèles améliore réellement la précision par rapport à chaque modèle pris individuellement.

Évaluation du modèle intégré

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
import pandas as pd
from tabulate import tabulate

accuracy = accuracy_score(y_true, y_pred_ensemble) * 100
precision = precision_score(y_true, y_pred_ensemble) * 100
recall = recall_score(y_true, y_pred_ensemble) * 100
f1 = f1_score(y_true, y_pred_ensemble) * 100
cm = confusion_matrix(y_true, y_pred_ensemble)
tn, fp, fn, tp = cm.ravel()
specificity = tn / (tn + fp) * 100

metrics_ensemble = pd.DataFrame([{
    "Modèle": "Vote Pondéré (DT + CNN)",
    "Accuracy (%)": round(accuracy, 2),
    "Precision (%)": round(precision, 2),
    "Recall (%)": round(recall, 2),
    "F1-Score (%)": round(f1, 2),
    "Spécificité (%)": round(specificity, 2)
}])

print(tabulate(metrics_ensemble, headers='keys', tablefmt='fancy_grid'))
```

Figure 3.27 : code Évaluation du modèle intégré basé précision

Ce bloc permet de calculer et d'afficher les métriques classiques d'évaluation du modèle d'intégration pondéré (CNN + Arbre de Décision), telles que l'accuracy, la précision, le

rappel, le F1-score et la spécificité, afin d'estimer la qualité globale de la classification. Ces indicateurs sont essentiels pour vérifier l'amélioration apportée par l'approche hybride.

- **Intégration basé historique**

Méthode d'intégration

```
weight_cnn=0.6
weight_dt=0.4

y_pred_ensemble = []
for val_cnn, val_dt in zip(mean_cnn, mean_dt):
    score = weight_cnn * val_cnn + weight_dt * val_dt
    y_pred_ensemble.append(1 if score >= 0.5 else 0)

y_pred_hist = np.array(y_pred_ensemble)
```

Figure 3.28 : code Méthode d'intégration basé historique

La méthode d'intégration basée sur l'historique de précision permet d'ajuster les prédictions de chaque modèle selon leur fiabilité mesurée sur les données d'entraînement. Cela renforce l'influence du modèle le plus performant tout en conservant une contribution équilibrée du second, améliorant ainsi la robustesse de la classification finale.

Évaluation du modèle

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matri
import pandas as pd
from tabulate import tabulate

accuracy = accuracy_score(y_true, y_pred_ensemble) * 100
precision = precision_score(y_true, y_pred_ensemble) * 100
recall = recall_score(y_true, y_pred_ensemble) * 100
f1 = f1_score(y_true, y_pred_ensemble) * 100
cm = confusion_matrix(y_true, y_pred_ensemble)
tn, fp, fn, tp = cm.ravel()
specificity = tn / (tn + fp) * 100

metrics_ensemble = pd.DataFrame([{
    "Modèle": "Base historique (DT + CNN)",
    "Accuracy (%)": round(accuracy, 2),
    "Precision (%)": round(precision, 2),
    "Recall (%)": round(recall, 2),
    "F1-Score (%)": round(f1, 2),
    "Spécificité (%)": round(specificity, 2)
}])

print(tabulate(metrics_ensemble, headers='keys', tablefmt='fancy_grid'))
```

Figure 3.29 : code Évaluation du modèle intégré basé historique

Ce bloc calcule les métriques de performance (accuracy, précision, rappel, F1-score et spécificité) du modèle résultant de l'intégration pondérée selon la précision historique par classe des modèles CNN et DT. Il permet d'évaluer l'efficacité globale de cette méthode d'intégration.

5-4-Resultats

5-4-1-Resultats DT

Tableau 01 : Résultat du modèle DT

	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Spécificité(%)
Exécution 01	82.5	80.77	91.3	85.71	70.59
Exécution 02	82.5	80.77	91.3	85.71	70.59
Exécution 03	82.5	80.77	91.3	85.71	70.59
Exécution 04	82.5	80.77	91.3	85.71	70.59
Exécution 05	82.5	80.77	91.3	85.71	70.59
Exécution 06	82.5	80.77	91.3	85.71	70.59
Exécution 07	82.5	80.77	91.3	85.71	70.59
Exécution 08	82.5	80.77	91.3	85.71	70.59
Exécution 09	82.5	80.77	91.3	85.71	70.59
Exécution 10	82.5	80.77	91.3	85.71	70.59
La moyenne	82.5	80.77	91.3	85.71	70.59

5-4-2- Resultats CNN

Tableau 02 : Résultat du modèle CNN

	Accuracy(%)	**Precision(%)**	**Recall(%)**	**F1-Score(%)**	**Spécificité(%)**
Exécution 01	97.5	95.83	100	97.87	94.12
Exécution 02	97.5	95.83	100	97.87	94.12
Exécution 03	90	95.24	86.96	90.91	94.12
Exécution 04	90	95.24	86.96	90.91	94.12
Exécution 05	90	95.24	86.96	90.91	94.12
Exécution 06	97.5	100	95.65	97.78	100
Exécution 07	97.5	100	95.65	97.78	100
Exécution 08	100	100	100	100	100
Exécution 09	100	100	100	100	100
Exécution 10	100	100	100	100	100
La moyenne	96	97.74	95.22	96.4	97.06

5-4-3- Resultat CNN+DT

- **Intégration avec vote pondéré**

Tableau 03 : Résultat de l'intégration basé précision (vote pondéré)

	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Spécificité(%)
Exécution 01	95	92	100	95.83	88.24
Exécution 02	95	92	100	95.83	88.24
Exécution 03	90	85.19	100	92	76.47
Exécution 04	90	85.19	100	92	76.47
Exécution 05	90	85.19	100	92	76.47
Exécution 06	100	100	100	100	100
Exécution 07	100	100	100	100	100
Exécution 08	97.5	95.83	100	97.87	94.12
Exécution 09	97.5	95.83	100	97.87	94.12
Exécution 10	100	100	100	100	100
La moyenne	95.5	93,62	100	96.34	89.41

- **Intégration avec base d'historique**

Tableau 04 : Résultat de l'intégration basé historique (par classe)

	Accuracy(%)	**Precision(%)**	**Recall(%)**	**F1-Score(%)**	**Spécificité(%)**
Exécution 01	97.5	95.83	100	97.87	94.12
Exécution 02	97.5	95.83	100	97.87	94.12
Exécution 03	90	95.24	86.96	90.91	94.12
Exécution 04	90	95.24	86.96	90.91	94.12
Exécution 05	90	95.24	86.96	90.91	94.12
Exécution 06	97.5	100	95.65	97.78	100
Exécution 07	97.5	100	95.65	97.78	100
Exécution 08	100	100	100	100	100
Exécution 09	100	100	100	100	100
Exécution 10	100	100	100	100	100
La moyenne	96	97.74	95.22	96.4	97.06

5-5-Discussion

Dans cette section, nous analysons et comparons les performances des différents modèles de classification implémentés pour la détection de tumeurs cérébrales à partir d'images IRM : le **réseau de neurones convolutifs (CNN)**, l'**arbre de décision (Decision Tree)** et deux méthodes d'intégration, basées respectivement sur la précision globale et sur l'historique des prédictions.

- **Performances du modèle CNN**

D'après le tableau des résultats, le modèle CNN présente des performances globalement très élevées. En effet, la moyenne de ses métriques sur les 10 exécutions successives est la suivante :

- **Accuracy : 96 %**
- **Précision : 97,74 %**

- **Recall : 95,22 %**
- **F1-Score : 96,4 %**
- **Spécificité : 97,06 %**

On constate également que le CNN atteint, dans certaines exécutions, une performance parfaite avec **100 % de précision, de rappel et d'accuracy**, traduisant sa forte capacité à distinguer les images normales de celles présentant une tumeur. Cette performance reste relativement stable à travers les exécutions, bien que quelques légères variations apparaissent, notamment dans les exécutions 3 à 5.

- **Performances du modèle Arbre de Décision (Decision Tree)**

Le modèle Arbre de Décision, quant à lui, obtient des résultats plus modestes, avec des valeurs constantes à travers les 10 exécutions :

- **Accuracy : 82,5 %**
- **Précision : 80,77 %**
- **Recall : 91,3 %**
- **F1-Score : 85,71 %**
- **Spécificité : 70,59 %**

Il est à noter que si le modèle parvient bien à identifier les cas positifs (rappel élevé), il est moins performant pour détecter les cas négatifs (spécificité de 70,59 %). Ce déséquilibre peut s'expliquer par la simplicité du modèle par rapport au CNN, et une moindre capacité à capter des caractéristiques complexes dans les images.

- **Performances de l'intégration basée sur la précision (Precision-Based)**

La méthode d'intégration pondérée selon la précision globale des deux modèles (base précision) améliore sensiblement les performances par rapport au Decision Tree seul, et dans certains cas s'approche de celles du CNN :

- **Accuracy moyenne : 95,5 %**
- **Précision : 93,62 %**
- **Recall : 100 %**
- **F1-Score : 96,34 %**

- **Spécificité : 89,41 %**

Cette méthode démontre ainsi que la combinaison des modèles, en tenant compte de leurs précisions respectives, permet de capitaliser sur les points forts de chacun, notamment en maintenant un rappel parfait tout en rehaussant la spécificité et l'accuracy par rapport au Decision Tree seul.

- **Performances de l'intégration basée sur l'historique (History-Based)**

Enfin, la méthode d'intégration historique, qui repose sur l'analyse des prédictions sur plusieurs itérations, donne des résultats identiques à ceux obtenus avec le CNN seul. La moyenne des métriques est ainsi exactement la même :

- **Accuracy : 96 %**
- **Précision : 97,74 %**
- **Recall : 95,22 %**
- **F1-Score : 96,4 %**
- **Spécificité : 97,06 %**

Ce résultat indique que dans ce cas précis, l'intégration historique n'apporte pas de gain significatif par rapport au CNN, et suit même exactement son comportement. Cela peut s'expliquer par la forte domination du CNN dans les prédictions, éclipsant l'apport du Decision Tree dans la pondération.

6-Conclusion

Dans ce chapitre, j'ai présenté mes contributions à travers ce mémoire de master. L'essentiel de ma contribution est l'étude et l'intégration de deux modèles d'apprentissage automatique pour la classification d'IRM cérébrales, dont un /-but diagnostic (IRM saine, et IRM tumorale). Les modèles étudiés sont les arbres de décision, et les réseaux de neurones convolutifs (CNN). Il en a résulté que les CNN domine largement les arbre de décision pour ce problème de segmentation d'IRM.

Conclusion générale

Le diagnostic médical moderne repose principalement sur l'exploration radiologique du corps humain. L'imagerie médicale est devenue ainsi une technologie d'excellence d'aide à l'établissement des diagnostics médicaux en un temps court, comparé à au temps de l'interprétation manuelles des données volumineuses en imagerie médicale, notamment en imagerie par résonnance magnétique.

Par ailleurs, les techniques modernes d'intelligence artificielle, à travers les modèles d'apprentissage automatique qui ne cessent d'être développés, ont permet des interprétations fiables et en un temps court d'images médicales, et ainsi l'établissement de diagnostic corrects des pathologies qui y existent.

Dans ce mémoire de master, je me suis intéressé à deux modèles d'apprentissage automatique et à leur utilisation pour la classification d'IRM cérébrales, dont l'objectif est de diagnostiquer des tumeurs si elles y existent. Le premier modèle est celui des arbres de décision et le second est celui des réseaux de neurones convolutifs. J'ai mené également une étude sur les approches d'intégration de ces deux modèle dont l'objectif et d'améliorer la performance du système de classification.

En perspective à mon travail, il sera judicieux d'étudier d'autres modèles est d'intégrer des modèles opérant à la fois sur des données images et sur des données cliniques factuelles. Dans cette situation des données cliniques vont consolider la décision qui serait prise en se basant sur les modèles opérant sur les données IRM.

Références

[1] https://www.larousse.fr/dictionnaires/francais/image/41604

[2] Nicolas Thome, Bases du traitement des images, Introduction et fondements , 13 Septembre 2016.

[3] Valérie Gouet-Brunet, Image Numérique, Cours 1 : les bases de l'image, 2006-2007.

[4] Diane Lingrand. Introduction au Traitement d'Images. Vuibert, 2008, Théories, algorithmes, applications, ISBN-13: 978-2711748662. http://www.vuibert.fr /ouvrage9782711748662- introduction-autraitement-d-images. html. hal-00463907 Livre.

[5] Vision numérique et interaction Sylvia Chalençon Licence Pro OCI - Licence MIME Université Paris 8.

[6] https://www.doctissimo.fr/html/sante/imagerie/radiographie_standard.htm

[7] https://www.passeportsante.net/fr/Maux/examens-medicaux-operations/Fiche.aspx?doc=tomodensitometrie-savoir-sur-cet-examen-medical

[8] https://www.msdmanuals.com/fr/accueil/sujets-particuliers/examens-d-imagerie-courants/imagerie-par-r%C3%A9sonance-magn%C3%A9tique-irm

[9] https://www.passeportsante.net/fr/Maux/examens-medicaux-operations/Fiche.aspx?doc=examen-echographie

[10] https://larevueia.fr/quest-ce-que-la-segmentation-dimages/

[11] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), 679-698.

[12] Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. International journal of computer vision, 1(4), 321-331.\nOsher, S., & Sethian, J. A. (1988).

Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi

formulations. Journal of computational physics, 79(1), 12-49.

[13] Adams, R., & Bischof, L. (1994). Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), 641-647.

[14] Nock, R., & Nielsen, F. (2004). Statistical region merging. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11), 1452–1458.

[15] Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. Computer

Vision, Graphics, and Image Processing, 29(1), 100–132.

[16] Caiming.Z et Yang.L "Study on artificial intelligence: The state of the art and future prospects" ScienceDirect, Journal of Industrial Information Integration page 01 (23) 8 Mai 2021.

[17] Marco. A « Comme Exigence Partielle de la Maitrise en Philosophie » Mémoire présenté à l'université du Québec à Trois-Rivières, page 04, Mars 1992.

[18] Emmert-Streib, F., Yli-Harja, O., & Dehmer, M. (2020). A clarification of misconceptions, myths and desired status of artificial intelligence. Doi: https://doi.org/10.48550/arXiv.2008.05607

[19] Ford, M. (2018). Architects of Intelligence: The truth about AI from the people building it. Packt Publishing Ltd.

[20] Asenjo Carvajal, A. (2020). Time to land prediction based on machine learning models (Bachelor's thesis, Universitat Politècnica de Catalunya). URL: http://hdl.handle.net/2117/329210

[21]. Vansh Jatana. ResearchGate. (2018). Dive into Machine Learning (PDF). URL: https://www.researchgate.net/publication/328614973_Dive_Into_Machine_Learning

[22] J. Yang and A. J. Awan, "Support vector machines on noisy intermediate scale quantum computers," September 2019.

[23] B. Mahesh, “Machine learning algorithms -a review,” 2019.

[24] M. El-Assady, W. Jentner, R. Kehlbeck, and U. Schlegel, “Towards xai : Structuring the processes of explanations,” in Human-Centered Machine Learning Perspectives Workshop, Universität Konstanz, University of Oklahoma, May 2019.

[25] W. Koehrsen, “Random forest simple explanation,” Dec 27 2017.

[26] Vargas, R., Mosavi, A., & Ruiz, R. (2018). Deep learning: a review. Doi: https://doi.org/10.20944/preprints201810.0218.v1

[27] Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd.

[28] Trahi, F. (2011). Prédiction de l'irradiation solaire globale pour la région de Tizi ouzou

par les réseaux de neurones artificiels : Application pour le dimensionnement d'une

installation photovoltaique pour l'alimentation du laboratoire de recherche LAMPA (Doctoral

dissertation, Université Mouloud Mammeri). URL:

https://www.ummto.dz/dspace/handle/ummto/623

[29] Kasri, S. (2010). Etude et modélisation de déclin de potentiel de surface par les réseaux de neurones (Doctoral dissertation, Université de Annaba-Badji Mokhtar).

[30] Feng, J., & Lu, S. (2019, June). Performance analysis of various activation functions in

artificial neural networks. In Journal of physics: conference series (Vol. 1237, No. 2, p.

022030). IOP Publishing. Doi: https://dx.doi.org/10.1088/1742-6596/1237/2/022030

[31] Feng, J., He, X., Teng, Q., Ren, C., Chen, H., & Li, Y. (2019). Reconstruction of porous

media from extremely limited information using conditional generative adversarial networks. Physical Review E, 100(3), 033308. Doi: https://doi.org/10.1103/PhysRevE.100.033308

[32] Venugopal, P. Vigneswaran, T. (2019). State-of-Health estimation of li-ion batteries in electric vehicle using IndRNN under variable load condition. Energies, 12(22), 4338. Doi: https://doi.org/10.3390/en1222433

[33] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541- 551. Doi: https://psycnet.apa.org/doi/10.1162/neco.1989.1.4.541

[34] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011, June). Flexible, high performance convolutional neural networks for image classification. In Twentysecond international joint conference on artificial intelligence, 1237–1242. Doi: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-210

[35] K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. LeCun, “Learning invariant features through topographic filter maps,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1605–1612, IEEE, June 2009.

[36] A.Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convo lutional neural networks,” Advances in Neural Information Processing Systems, vol. 25, 2012.

[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, ..., and T. Darrell, “Caffe : Convolutional architecture for fast feature embedding,” in Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678, November 2014.

[38] Y.-G. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, and S.-F. Chang, “Modeling multimodal clues in a hybrid deep learning framework for video classification,” IEEE Transactions on

Multimedia, vol. 20, no. 11, pp. 3137–3147, 2018.

[39] T. T. Teoh and Z. Rong, Artificial Intelligence with Python. Springer Nature, 2022.

[40] Z. Rezgui, “Rapport de stage : Détection et classification de visages pour la descrip tion de l’égalité femme-homme dans les archives télévisuelles,” tech. rep., University of Carthage, University of Twente, November 2019.

[41] C. Kong and S. Lucey, “Take it in your stride : Do we need striding in cnns?,” arXiv preprint arXiv :1712.02502, 2017.

[42] S. Mittal, “A survey of accelerator architectures for 3d convolution neural networks,” Journal of Systems Architecture, vol. 115, p. 102041, 2021.

[43] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks : Anoverview and application in radiology,” Insights into Imaging, vol. 9, no. 4, pp. 611 629, 2018.

[44] A. Gridi and T. Guerram, “Un outil de deep learning pour les données textuelles,” Mas ter’s thesis, Université Oum El Bouaghi, 2020.

[45] X. Wang, G. Liang, Y. Zhang, H. Blanton, Z. Bessinger, and N. Jacobs, “Inconsistent performance of deep learning models on mammogram classification,” Journal of the American College of Radiology, vol. 17, no. 6, pp. 796–803, 2020.

[46] M.Coşkun,.YILDIRIM,U..A.R.Ayşegül,andY.Demir,“Anoverviewofpopular deep learning methods,” European Journal of Technique (EJT), vol. 7, no. 2, pp. 165 176, 2017.

[47] https://www.mygreatlearning.com/blog/alexnet-the-first-cnn-to-win-image-net/

[48] https://www.ibm.com/fr-fr/think/topics/decision-trees

[49] mobiskill. « Qu’est Ce Que Le Python et Quelle Est Son Utilisation ? », Mobiskill. (18 déc. 2020), adresse : https : / / mobiskill . fr / blog / conseils - emploi - tech/quest-ce-que-le-python-et-quelle-est son-utilisation/ (visité le 04/07/2023).

[50] B. L. « TensorFlow : tout savoir sur la bibliothèque Machine Learning open source ». (), adresse : https : / / www . lebigdata . fr / tensorflow - definition - tout - savoir#:~:text=Pour%20faire%20simple%2C%20TensorFlow%20est%20une%20biblioth%C3%A8que%20de,d%E2%80%99apprentissage%20exp%C3%A9rimentales%20et%20de%20les%20transformer%20en%20logiciels. (visité le 04/07/2023).

[51] https://www.tresfacile.net/la-bibliotheque-opencv-python/

[52] [Enligne].https://www.medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg googlenet-resnet-and-more-666091488df5.

[53] F. Nacer «Reconnaissance d'expression faciale à partir d'un visage réel» Mémoire de master, Université de 8/Mai/19450, Guelma , 2019