

République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي



Université Mohamed El Bachir El Ibrahimi
Bordj Bou Arréridj
جامعة محمد البشير الإبراهيمي برج بوعريريج
Faculté des Mathématiques et d'Informatique
Département d'informatique

Mémoire de Projet de Fin d'Études

Pour obtenir un diplôme de master en informatique

Option :

Technologies de l'Information et de la Communication (TIC)

Classification des IRM cérébrales Pathologiques avec Optimisation

Préparé par :
M. DJEBARNI ilyes

Encadré par :
M. CHELAKH hafida

Promotion 2024/2025

Dédicace

الحمد لله على إحسانه، والشكر له على توفيقه وامتنانه، وأشهد أن لا إله إلا الله وحده لا شريك له تعظيمًا لشأنه، وأشهد أن محمدًا عبده ورسوله الداعي إلى رضوانه، صلى الله عليه وعلى آله وأصحابه. أما بعد، أهدي هذا العمل المتواضع:

إلى من جعل الله سبحانه وتعالى الجنة تحت قدميها، وكرّمها في كتابه العزيز. إلى مصدر الحنان الذي لا ينضب، إلى من سهرت الليالي لأهنا بنومي، وقامت بتربيتي تربية صالحة. إلى من كانت دعواتها ترافقني في كل أحوالي. إلى أمي الحبيبة، حفظك الله وأطال في عمرك.

إلى من مهد لي طريق العلم وأثار دربي بنور لا ينطفئ، إلى من علّمني وأكسبني شخصية قوية، إلى من لم ييخل عليّ بنصائحه وإرشاداته، إلى صاحب السيرة العطرة والفكر المستنير، إلى أبي الغالي، حفظك الله وأطال في عمرك.

إلى أخي العزيز اسلام، الذي كان دائمًا مصدر فخري. الذي وقف ومازال يقف بجانبني في كل خطوة من خطوات حياتي.

إلى أخي الصغير الحبيب اسحاق، الشاب الطموح المثابر، أنار الله دربه ووفقه.

إلى أختي المدللة فطيمة الزهراء، المنبع الثاني للحنان حفظك الله بحفظه ووفقك في امتحان شهادة التعليم المتوسط.

إلى أحبائي يوبا و منذر اللذان كانا خير معين لي بعد الله عز وجل في هذا البحث جزاهما الله خير الجزاء وجعل نصائحهم في ميزان حسناتهم

إلى رفقاء الدرب، إلى أصدقائي الأعزاء وزملائي الرائعين. وأخص بالذكر أخي فؤاد.. لم تكن هذه الرحلة لتكون مليئة بالنجاح والتحديات الجميلة لولا دعمكم المستمر ومساندتكم، أتمنى لكم كل التوفيق والنجاح.

وختامًا، أود أن أذكر نفسي وإياكم: "وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ"

- إلياس -

Remerciements

الحمد لله الذي بنعمته تتم الصالحات، وبتوفيقه تنجز الأعمال. أرفع أسمى آيات الشكر والامتنان لله سبحانه وتعالى على منحه لي القوة والصبر والإرادة لإتمام هذا العمل المتواضع.

كما أتقدم بخالص الشكر والتقدير إلى المشرفة الكريمة، الأستاذة شلاخ حفيظة، على توجيهاتها القيمة ودعمها المستمر طوال فترة إنجاز هذا المشروع. كانت نصائحها وإرشاداتها من العوامل الأساسية التي ساهمت في نجاحي.

أود أيضًا أن أعبر عن امتناني العميق لكل أساتذتي على جهودهم الكبيرة التي بذلوها في تزويدي بالمعرفة والعلم، وعلى تفانيهم في تقديم تعليم عالي الجودة.

أخص بالذكر أعضاء لجنة التحكيم الذين تفضلوا بقبول قراءة وتقييم هذا العمل، متوجها إليهم بأسمى عبارات الشكر على وقتهم وجهودهم.

إلى أحبائي يوبا و منذر اللذان كانا خير معين لي بعد الله عز وجل في هذا البحث جزاهما الله خير الجزاء وجعل نصائحهم في ميزان حسناتهم

ولا يفوتني أن أشكر أصدقائي وزملائي وكل من قدم لي الدعم والمساعدة خلال هذه الرحلة العلمية. دعمكم لي لا يُقدر بثمن.

Abstract

The automatic classification of brain MRI images represents a fundamental challenge in the medical field, as it effectively assists practitioners in diagnosing cerebral pathologies. In this study, we explored several artificial intelligence approaches for multi-class brain tumor classification by evaluating different models : a traditional Convolutional Neural Network (CNN), the ResNet-18 model with and without optimization, the same model combined with the symbolic DRB classifier, and the Vision Transformer (ViT) model in both its base and optimized versions. Model optimization was performed using the Adam algorithm, known for its rapid convergence. Experimental results demonstrated varying performance across the tested models. The optimized CNN achieved a notable accuracy of 98.02%.confirming its effectiveness despite its relative simplicity. The optimized ResNet-18 delivered excellent performance with 99.08% accuracy and F1-scores exceeding 0.98 for most classes, indicating strong generalization and optimal model tuning. In contrast, its DRB classifier-integrated version without optimization reached 91.61%, highlighting both the potential and limitations of this combination without prior refinement. The ViT model's case was particularly remarkable : without optimization, it achieved only 25.30% accuracy, reflecting poor initial learning capability. However, after optimization, its performance surged to 99.79%, making it the best-performing model in the study. This result clearly demonstrates the crucial importance of optimization in machine learning systems, particularly for advanced architectures. In summary, this work highlights the complementary roles of model architecture, classifier selection, and optimization techniques in developing high-performance intelligent systems for brain MRI classification. The findings open promising avenues for integrating these models into clinical diagnostic tools to enhance accuracy and efficiency in medical diagnosis.

Keywords : Artificial Intelligence, Deep Learning, Convolutional Neural Networks (CNN), ResNet-18, Vision Transformer (ViT), Optimization, Deep Rule-Based (DRB), Magnetic Resonance Imaging (MRI), Image Classification.

Résumé

La classification automatique des images IRM du cerveau représente un enjeu fondamental dans le domaine médical, car elle permet d'assister efficacement les praticiens dans le diagnostic des pathologies cérébrales. Dans le cadre de ce travail, nous avons exploré plusieurs approches d'intelligence artificielle pour la classification multi-classes des tumeurs cérébrales, en évaluant différents modèles : un réseau de neurones convolutif classique (CNN), le modèle ResNet-18 avec et sans optimisation, ce même modèle combiné au classificateur symbolique DRB, ainsi que le modèle Vision Transformer (ViT) dans ses versions brute et optimisée. L'optimisation des modèles a été réalisée à l'aide de l'algorithme Adam, réputé pour sa rapidité de convergence. Les résultats expérimentaux ont montré des performances variées selon les modèles utilisés. Le CNN optimisé a atteint une précision notable de 98.02%, confirmant son efficacité malgré sa relative simplicité. Le ResNet-18 optimisé a offert une excellente performance avec une précision de 99.08% , et des f1-scores supérieurs à 0.98 pour la majorité des classes, témoignant d'une bonne généralisation et d'un réglage optimal du modèle. En revanche, sa version associée au classificateur DRB, sans optimisation, a atteint 91.61%, illustrant l'intérêt mais aussi les limites de cette combinaison sans affinement préalable. Le cas du modèle ViT est particulièrement remarquable : sans optimisation, il n'a atteint que 25.30%, ce qui reflète une mauvaise capacité d'apprentissage initial. Toutefois, grâce à l'optimisation, ses performances ont bondi jusqu'à 99.79%, devenant ainsi le meilleur modèle de l'étude. Ce résultat démontre de manière claire l'importance cruciale de l'optimisation dans les systèmes d'apprentissage automatique. En somme, ce travail met en lumière le rôle complémentaire entre l'architecture choisie, le classificateur utilisé, et les techniques d'optimisation appliquées, pour obtenir un système intelligent performant pour la classification des IRM cérébrales.

Mots clés : Intelligence Artificielle, Apprentissage Profond, Réseaux de Neurones Convolutifs (CNN), ResNet-18, Vision Transformer (ViT), Optimisation, Deep Rule-Based (DRB), Imagerie par Résonance Magnétique (IRM), Classification d'Images.

ملخص

يُعتبر التصنيف التلقائي لصور الرنين المغناطيسي للدماغ تحديًا أساسيًا في المجال الطبي، حيث يُساهم بشكل فعال في مساعدة الأطباء على تشخيص الأمراض الدماغية. في هذا البحث، قمنا بدراسة منهجيات متعددة للذكاء الاصطناعي لتصنيف أورام الدماغ بأنواعها، من خلال تقييم نماذج مختلفة تشمل: الشبكة العصبية الالتفافية التقليدية (CNN)، نموذج ResNet-18 مع التحسين وبدونه، نفس النموذج مع المصنف الرمزي، DRB بالإضافة إلى نموذج Vision Transformer (ViT) بنسخته الأساسية والمحسنة. كما استخدمنا خوارزمية Adam لتحسين النماذج نظرًا لسرعة تقاربها المعروفة. أظهرت النتائج التجريبية تباينًا في الأداء بين النماذج المختلفة. حيث حقق نموذج CNN المحسّن دقة عالية بلغت 98.02%، مما يؤكد فعاليته رغم بساطته النسبية، بينما سجّل نموذج ResNet-18 المحسّن أداءً مميّزًا بدقة 99.08% مع تحقيق درجات F1 أعلى من 0.98 لمعظم الفئات، مما يعكس قدرة تصميم جيد وضبطًا دقيقًا للنموذج. أما النسخة المدمجة مع المصنف DRB دون تحسين، فقد حققت دقة 91.61% فقط، ما يُظهر إمكانيات هذا التركيب مع إبراز محدودية أدائه عند غياب التحسين المناسب. وكانت نتائج نموذج ViT الأكثر إثارة للاهتمام، حيث سجل دقة 25.30% بدون تحسين، ما يعكس ضعفًا في التعلّم الأولي، لكن بعد التحسين ارتفعت النتيجة بشكل كبير إلى 99.79%، ليصبح بذلك أفضل نموذج في هذه الدراسة. هذه النتيجة تؤكد بشكل واضح أهمية التحسين في أنظمة التعلّم الآلي. باختصار، يبرز هذا البحث الأهمية التكاملية بين البنية المعمارية المستخدمة، وتقنيات التحسين المطبقة، وإدماج نظم ذكاء عالي الكفاءة في مهام التصنيف الطبي. ويفتح بذلك آفاقًا جديدة لاستخدام هذه النماذج في الأنظمة الطبية للمساعدة، مما قد يُساهم في تسريع ودقة تشخيص الأمراض الدماغية.

كلمات مفتاحية : الذكاء الاصطناعي، التعلّم العميق، الشبكات العصبية التلافيفية، (CNN)، ResNet-18، محول الرؤية، (ViT) التحسين، القواعد العميقة (DRB)، التصوير بالرنين المغناطيسي، (MRI) تصنيف الصور.

Liste des abréviations

IA : Intelligence Artificielle

IRM : Imagerie par Résonance Magnétique

CNN : Convolutional Neural Network

ViT : Vision Transformer

ResNet-18 : Residual Networks

DRB : Deep Rule-Based Classifier

ML : Machine Learning

DL : Deep Learning

RNN : Recurrent Neural Network

Adam : Adaptive Moment Estimation

SGD : Stochastic Gradient Descent

F1 : F1-score

GPU : Graphics Processing Unit

TP : True Positives

TN : True Negatives

FP : False Positives

FN : False Negatives

Table des matières

Dédicace	I
Remerciements	II
Abstract	III
Résumé	IV
V	ملخص
Liste des abréviations	VI
Introduction Générale	1
I Notion d'étude	3
1 L'imagerie médicale	4
1.1 Introduction	4
1.2 L'image	4
1.3 L'image numérique	4
1.4 Les caractéristiques d'une image numérique	5
1.4.1 Le pixel	5
1.4.2 La résolution	6
1.5 L'imagerie médicale	6
1.5.1 Les modalités d'images médicales	6
1.5.2 Imagerie par résonance magnétique(IRM)	7
1.5.3 IRM cérébrale	7
1.6 Imagerie par résonance magnétique	7
1.6.1 Définition	7
1.6.2 Principe et fonctionnement de l'IRM	8
1.7 Anatomie du cerveau	9
1.8 Pathologies cérébrales	10
1.8.1 Gliomes	10
1.8.2 tumeurs pituitaires	10
1.8.3 Tumeur de méningiome	10
1.9 Conclusion	11
2 L'intelligence artificielle	12

2.1	Introduction	12
2.2	L'utilisation de l'intelligence artificielle	12
2.3	Les domaines d'applications de L'IA	13
2.4	Machine Learning (ML)	14
2.5	Apprentissage Profond (Deep Learning)	15
2.6	Les réseaux de neurones	15
2.6.1	Réseau de neurones récurrents (RNN)	15
2.6.2	Mémoire à long terme (LSTM)	16
2.7	conclusion	17
3	La classification	18
3.1	Introduction	18
3.2	Définition	18
3.3	Les étapes d'une classification	18
3.4	Mesure de la similarité	19
3.5	Les méthodes de la classification :	19
3.5.1	Méthodes non supervisé	19
3.5.2	Méthodes supervisée	20
3.6	Quelques algorithmes de la classification supervisée	20
3.6.1	Arbre de décision	20
3.6.2	Le classifieur naïf de Bayes	21
3.6.3	Les machines à vecteur de support (SVM)	21
3.6.4	Deep Rule-Based Classifier (DRB)	22
3.7	Présentation de notre méthodologie	25
3.8	Conclusion	26
4	Conception, Implémentation	27
4.1	Introduction	27
4.2	Explication du choix des modèles adoptés	28
4.2.1	CNN (Le Réseau de Neurones Convolutifs)	29
4.2.2	ResNet18 (Residual Network)	32
4.2.3	Vision Transformer (ViT)	35
4.3	Les algorithmes d'optimisation pour l'apprentissage profond	38
4.3.1	L'algorithme d'optimisation Adam	38
4.3.2	Avantages de l'algorithme d'optimisation Adam	38
4.3.3	Le fonctionnement fondamental de l'algorithme d'optimisation Adam	39
4.3.4	Réglage des hyperparamètres	40
4.4	Implémentation	41
4.4.1	Matériel Outils	41
4.5	Technologies et outils utilisés	41
4.5.1	Google Colab	41
4.5.2	Visual Studio Code	42
4.5.3	Langage de programmation Python	42
4.5.4	Dataset	45
4.6	Mesures d'évaluation des algorithmes de classifications	46
4.7	Implémentation de code et explication	48

4.8	Conclusion	51
II	Étude comparative	52
5	Évaluation	53
5.1	Introduction	53
5.2	CNN (Convolution Neural Network)	54
5.2.1	Avec optimisation	54
5.2.2	Sans optimisation	55
5.3	Res-Net18 (Residual Network)	57
5.3.1	Avec optimisation	57
5.3.2	Sans optimisation	58
5.4	ViT (vision transformer)	60
5.4.1	Avec optimisation	60
5.4.2	Sans optimisation	61
5.5	Res-net18 avec DRB (deep rule-based)	63
5.5.1	Avec optimization	63
5.5.2	Sans optimisation	64
5.6	Analyse Comparative de la Performance des Modèles avec et sans Optimisation	65
5.7	Conclusion	68
	Conclusion générale et perspectives	69
	Bibliography	74

Table des figures

1.1	Représentation d'une image numérique (HORRI Taqyedine, 2021)	5
1.2	Image et pixels (HORRI Taqyedine, 2021)	5
1.3	Quelques exemples de résolutions (HORRI Taqyedine, 2021)	6
1.4	Exemple d'image IRM	8
1.5	L'anatomie du cerveau (Merck Manuals, 2023)	9
1.6	Tumeurs cérébrales	10
2.1	L'intelligence artificielle	13
2.2	IA, Machine Learning et Deep Learning (Canonar, 2023)	14
2.3	Réseau neuronal récurrent	16
2.4	Structure d'une cellule LSTM (Abirami & Chitra, 2018)	17
3.1	Structure d'un arbre de décision	21
3.2	Structure graphique du classifieur naïf de Bayes	21
3.3	Principe de séparation maximale en SVM	22
3.4	Architecture du classificateur DRB (P. P. Angelov & Gu, 2018)	23
4.1	Architecture du système de détection et classification des tumeurs cérébrales	28
4.2	Architecture du Réseau de Neurones Convolutionnel (CNN) (Alzubaidi et al., 2021)	30
4.3	Opération de convolution (Kimura et al., 2019)	31
4.4	Types de Pooling (Alzubaidi et al., 2021)	31
4.5	Couche entièrement connectée (Alzubaidi et al., 2021)	32
4.6	Opération ReLU (Prabhu, 2018)	32
4.7	Original ResNet-18 Architecture (Ramzan et al., 2019)	33
4.8	Architecture ResNet18 (Ramzan et al., 2019)	34
4.9	Architecture du transformateur de vision (ViT)(Dosovitskiy et al., 2020)	35
4.10	L'évolution des algorithmes d'optimisation (Ahmed, 2020)	39
4.11	Interface de Google Colab	41
4.12	logo Visual Studio Code	42
4.13	Logo Python	42
4.14	Pandas logo	43
4.15	Matplotlib logo	43
4.16	PIL logo	43
4.17	NumPy logo	44
4.18	Scikit-learn logo	44
4.19	Seaborn logo	44
4.20	PyTorch logo	45
4.21	Exemples d'images IRM par classe dans le dataset	46

Table des figures

5.1	Courbes de performance du modèle CNN avec optimisation	54
5.2	Courbes de performance du modèle CNN sans optimisation	55
5.3	Courbes de performance du modèle ResNet18 Avec optimisation	57
5.4	Courbes de performance du modèle ResNet18 sans optimisation	58
5.5	Courbes de performance du modèle ViT avec optimisation	60
5.6	Courbes de performance du modèle ViT sans optimisation	61
5.7	Performance du modèle DRB avec optimisation	63
5.8	Performance du modèle DRB sans optimisation	64

Liste des tableaux

4.1	Description des principaux packages Python	45
4.2	Matrice de confusion multi-classe	47
5.1	Rapport de classification CNN-A	54
5.2	Rapport de classification CNN-S	56
5.3	Rapport de classification Res-Net18 A	57
5.4	Rapport de classification Res-Net18 S	59
5.5	Rapport de classification ViT-A	60
5.6	Rapport de classification ViT-S	62
5.7	Rapport de classification ResNet-DRB A	63
5.8	Rapport de classification ResNet-DRB S	65

Introduction générale

Les tumeurs cérébrales constituent l'une des pathologies les plus complexes et dangereuses affectant le système nerveux central, représentant une menace sérieuse pour la vie humaine en raison de leur caractère agressif et de leur impact destructeur sur les tissus cérébraux. Malgré les avancées médicales et technologiques, leur diagnostic demeure un défi majeur, notamment face à l'augmentation préoccupante des cas au cours des dernières décennies. Dans ce contexte, le recours à des technologies avancées telles que l'intelligence artificielle (IA) et le Deep Learning s'avère indispensable pour optimiser les procédures de diagnostic. L'imagerie par résonance magnétique (IRM) joue un rôle clé dans la localisation et l'évaluation des tumeurs, mais l'interprétation manuelle de ces images peut être lente et sujette à des erreurs. D'où l'intérêt croissant pour les systèmes de diagnostic assisté par ordinateur (CAD), qui exploitent des algorithmes d'apprentissage profond pour analyser automatiquement les images médicales, offrant ainsi une aide précieuse aux médecins dans la prise de décision.

Dans ce travail, nous nous sommes fixés comme objectif de développer un système intelligent capable de classier automatiquement les tumeurs cérébrales à partir d'images IRM. Pour cela, nous avons exploré plusieurs architectures d'apprentissage profond, notamment les réseaux de neurones convolutifs (CNN), le modèle ResNet-18, le Vision Transformer (ViT), ainsi qu'une combinaison de ResNet-18 avec un classificateur symbolique basé sur des règles (DRB), avec ou sans optimisation. Une problématique essentielle se pose alors : parmi tous ces modèles, quel est celui qui offre les meilleures performances en termes de précision, de robustesse et de fiabilité pour la classification des IRM cérébrales ?

Afin d'y répondre, nous avons mis en œuvre différentes techniques d'extraction de caractéristiques, de réduction de dimension et d'optimisation des hyperparamètres pour chaque modèle, avant de les évaluer rigoureusement selon des indicateurs standards comme la précision (accuracy), le rappel (recall) et la précision (precision). Cette étude vise donc non seulement à améliorer le diagnostic assisté par IA, mais aussi à orienter les futurs choix technologiques vers l'approche la plus adaptée pour la classification automatique des tumeurs cérébrales, contribuant ainsi à une prise en charge plus rapide et plus fiable des patients.

Ce rapport est structuré comme suit :

- Le premier chapitre est consacré aux notions médicales de base. Il présente l'imagerie médicale, avec un accent particulier sur l'imagerie par résonance magnétique (IRM), ainsi que l'anatomie du cerveau et les principales pathologies cérébrales, notamment les types de tumeurs étudiées.

- Le deuxième chapitre introduit les concepts fondamentaux de l'intelligence artificielle, du Machine Learning et du Deep Learning. Il explore également les différents types d'apprentissage, les architectures de réseaux de neurones, et les avantages et limites de l'IA dans le domaine médical.
- Le troisième chapitre porte sur la classification automatique. Il décrit les étapes d'un processus de classification, les méthodes supervisées et non supervisées, les mesures de similarité, ainsi que plusieurs algorithmes classiques de classification, dont les arbres de décision, K-NN, les classificateurs bayésiens, et le modèle DRB (Deep Rule-Based).
- Le quatrième chapitre est dédié à la partie expérimentale. Il détaille la conception du système proposé, les modèles de Deep Learning implémentés (tels que CNN, ResNet18, ViT, etc.), les étapes de prétraitement, l'entraînement, ainsi que l'évaluation comparative des performances à l'aide de métriques telles que l'exactitude, le rappel et la précision. Ce chapitre conclut avec une analyse critique des résultats obtenus.
- le cinquième chapitre est consacré à une analyse approfondie des résultats et à une discussion des différents modèles proposés. Il met en évidence les forces et les faiblesses de chaque méthode en évaluant sa capacité de généralisation sur des images initiées par IRM afin de déterminer quel modèle est le plus approprié pour une catégorisation fiable et précise des tumeurs cérébrales.

partie I

Notion d'étude

Chapitre 1

L'imagerie médicale

1.1 Introduction

Grâce aux progrès scientifiques, nous disposons aujourd'hui d'une grande variété de systèmes d'imagerie : radiographie traditionnelle, échographie, imagerie par résonance magnétique (IRM), tomodensitométrie (TDM), etc. Ces développements permettent d'inspecter un organe et de suivre son fonctionnement grâce à l'imagerie médicale. Dans ce chapitre, nous allons présenter les notions fondamentales de l'image. Nous commencerons par quelques définitions générales, puis nous décrirons les systèmes précédemment cités et enfin nous présenterons les bases de données que nous avons utilisées pour notre projet.

1.2 L'image

On définit l'image comme étant la reproduction fidèle ou analogique d'une scène réelle. C'est également un regroupement organisé d'informations qui, une fois présentées, portent une signification pour l'œil humain. On peut la caractériser comme une fonction $I(x, y)$ à brillance analogique continue, établie dans un domaine limité, où x et y représentent les coordonnées spatiales d'un point de l'image et I correspond à une fonction d'intensité lumineuse et de couleurs. Dans cette perspective, l'image n'est pas exploitable par la machine, ce qui rend nécessaire sa conversion en format numérique (EL-HACHEMI, 2018).

1.3 L'image numérique

Une image numérique est une image dont la surface est segmentée en unités de taille standard, appelées cellules ou pixels. Chaque unité possède des attributs particuliers : la teinte (ou le ton de gris) sélectionnée et la position correspondante dans l'image réelle. La numérisation d'une image consiste à transformer son état analogique en une représentation numérique sous forme de matrice bidimensionnelle de valeurs numériques $f(x, y)$. x, y : positions cartésiennes d'un point sur l'image. $f(x, y)$: degré d'intensité. Les images données sont habituellement organisées en un tableau I de n lignes et p colonnes

pour faciliter leur représentation et leur adresse. Chaque élément $I(x,y)$ symbolise un pixel de l'image, sa valeur étant liée à un niveau de gris codé sur m bits. La valeur à chaque point dénote l'intensité lumineuse détectée par le capteur.

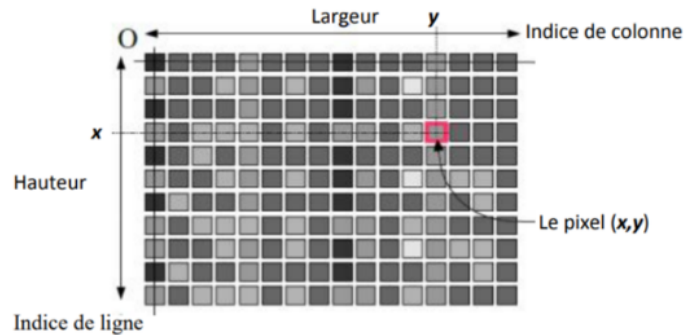


FIG. 1.1 : Représentation d'une image numérique (HORRI TAQYEDDINE, 2021)

1.4 Les caractéristiques d'une image numérique

1.4.1 Le pixel

Un pixel est l'élément le plus petit d'une surface d'affichage, tel qu'un écran d'ordinateur. On le décrit fréquemment comme un petit carré de couleur. Quand on effectue un zoom sur une image numérique matricielle (les images vectorielles ne reposent pas sur les pixels), il est évident que celle-ci est constituée d'un grand nombre de petits carrés, comme démontré dans l'illustration (TAQYEDDINE et al., 2020).

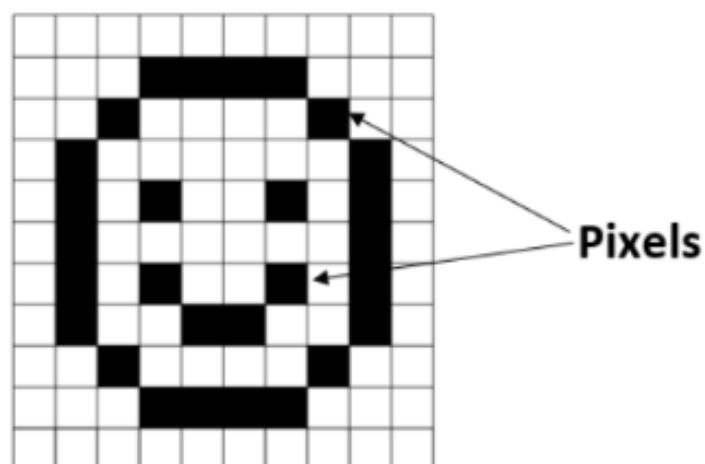


FIG. 1.2 : Image et pixels (HORRI TAQYEDDINE, 2021)

1.4.2 La résolution

La résolution se caractérise par une quantité de pixels par unité de longueur d'image à numériser, exprimée en dpi (dots per inch) ou ppp (points par pouce). On évoque la définition en ce qui concerne un écran, tandis qu'on se réfère à la résolution pour une image. Plus il y a de pixels par unité de longueur, plus l'image est riche en détails et plus sa qualité s'améliore (mais cela entraîne une augmentation de la taille du fichier). En d'autres termes, la résolution d'une image se réfère au degré de détails qui sera représenté sur celle-ci. Voir l'exemple ci-dessous pour plus de clarté (TAQYEDDINE et al., 2020).

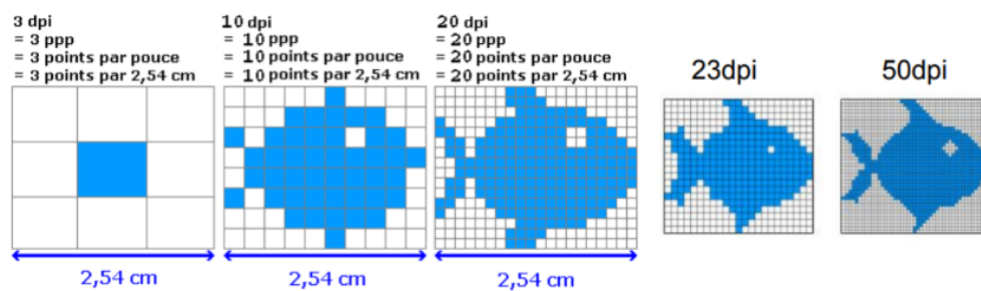


FIG. 1.3 : Quelques exemples de résolutions (HORRI TAQYEDDINE, 2021)

1.5 L'imagerie médicale

L'imagerie médicale fait référence à l'emploi du rayonnement électromagnétique et d'autres technologies spécifiques pour réaliser des images des structures internes du corps humain dans le but de poser un diagnostic médical. On distingue diverses catégories d'images en fonction de la technologie utilisée dans la zone concernée et/ou de l'affection spécifique.

1.5.1 Les modalités d'images médicales

Les principes physiques impliqués dans diverses sources d'images sont très variés et offrent des explications sur le corps humain, tant du point de vue morphologique que fonctionnel. Ces plateformes d'images sont également en concurrence sur le marché, ce qui pourrait expliquer les progrès constants qu'elles réalisent. Les techniques les plus couramment utilisées comprennent l'imagerie par résonance magnétique (IRM), le tomodensitométrie (TDM), la tomographie par émission, l'échographie, entre autres. Dans cette partie, notre attention est principalement dirigée vers les images obtenues par IRM et TDM.

1.5.2 Imagerie par résonance magnétique(IRM)

Le principe de l'IRM est applicable à l'organisme, car il renferme une profusion de petits « aimants atomiques », le proton du noyau de l'atome d'hydrogène étant le plus répandu et réactif. Le concept de l'IRM s'appuie sur la répartition aléatoire des protons dotés de caractéristiques magnétiques. Cela se déroule en trois phases. Initialement, l'IRM expose le corps à un champ magnétique extrêmement fort (30 000 fois plus puissant que celui de la Terre), orientant ainsi tous les protons dans une direction uniforme. Par la suite, les protons sont stimulés par des ondes radioélectriques, ce qui change leur alignement. Finalement, l'excitation est abruptement arrêtée et l'appareil capte une onde qualifiée de « résonance » via des antennes spécifiquement élaborées. L'analyse informatique du signal transmis permet d'établir les images des organes internes en utilisant des méthodes similaires à celles qui ont été mises au point pour la radiographie aux rayons X ou les scanners

L'IRM offre également la possibilité de créer des contrastes d'image en employant diverses séquences d'impulsions et en ajustant les paramètres d'imagerie liés au temps de relaxation longitudinal (T1) et au temps de relaxation transversal (T2). L'IRM corporelle est réalisée afin d'obtenir des précisions structurelles sur les organes (comme le cerveau, le foie, etc.) au sein du crâne, du thorax, de l'abdomen et du pelvis. Cela facilite l'établissement d'un diagnostic, la surveillance de la progression de la maladie et même l'évaluation de l'efficacité du traitement. (CROISÉ-LAURENT, 2008)

1.5.3 IRM cérébrale

L'IRM est particulièrement efficace pour repérer les irrégularités cérébrales lors des phases initiales de certaines pathologies et excelle dans la détection précoce d'accidents vasculaires cérébraux, d'infections et de néoplasmes cérébraux. L'IRM est particulièrement efficace pour détecter les affections de la substance blanche, comme la sclérose en plaques et l'encéphalite post-infectieuse.

1.6 Imagerie par résonance magnétique

1.6.1 Définition

L'imagerie par résonance magnétique, qui est une méthode d'imagerie médicale sans intrusion, fait appel à des radiations non ionisantes pour produire des images anatomiques très précises en 2D ou 3D, facilitant ainsi l'identification des tissus pathologiques. Il repose sur le principe de la résonance magnétique nucléaire. Habituellement, l'IRM est perçue comme un examen crucial pour le diagnostic et le suivi des affections des tissus mous (AHMAD et al., 2013).

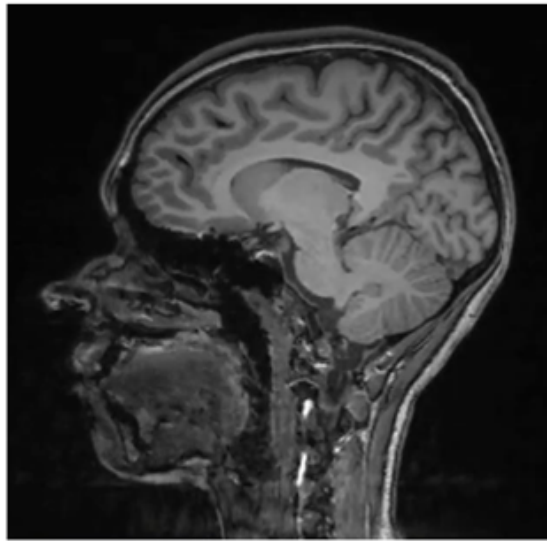


FIG. 1.4 : Exemple d'image IRM

1.6.2 Principe et fonctionnement de l'IRM

Utilisant principalement les protons d'hydrogène présents dans le corps humain, y compris dans l'eau des tissus muqueux, l'imagerie par résonance magnétique (IRM) est une approche non invasive basée sur le phénomène de résonance magnétique nucléaire. Un aimant supraconducteur produisant un champ magnétique fort et uniforme, des bobines de radiofréquence (RF) qui envoient et reçoivent des signaux, et des bobines de gradient assurant le codage spatial du signal sont les principaux composants du système IRM. Ces éléments sont gérés par un ordinateur central contrôlé par un logiciel spécialisé permettant la visualisation et la reconstruction des images. (AHMAD et al., 2013).

Le fonctionnement de l'IRM suit un processus en plusieurs étapes : d'abord, les protons s'alignent selon le champ magnétique principal (B_0). Ces protons sont ensuite stimulés par une radiofréquence (RF) à la fréquence de Larmor, ce qui entraîne leur basculement vers un plan transversal. Lorsque cette excitation cesse, les protons retournent à leur état initial et émettent un signal que les bobines peuvent détecter. Pour créer une image, ce signal est ensuite soumis à une transformée de Fourier.

La qualité et le contraste de l'image dépendent de paramètres tels que le temps de répétition (TR), le temps d'écho (TE) et l'angle de bascule (α), qui définissent différentes séquences d'imagerie adaptées aux structures à analyser. Ainsi, l'IRM permet de produire des images anatomiques précises en 2D ou 3D, utiles pour la détection des pathologies, le suivi de traitement et l'analyse fine de l'anatomie interne. The picture quality and contrast are determined by parameters like the repetition time (TR), echo time (TE), and bascule angle (α), which define various image sequences appropriate for the structures to be analysed. As a result, IRM makes it possible to create precise 2D or 3D anatomical images that are helpful for pathology detection, treatment monitoring, and fine-grained internal anatomy analysis (KOZÁK, 2024) .

1.7 Anatomie du cerveau

Le cerveau, centre de commande du corps humain, est un organe complexe composé à 60 pour cent de lipides et à 40 pour cent d'eau, de protéines, de glucides et de sels minéraux. Il fait partie du système nerveux central, aux côtés du cervelet et du tronc cérébral, ce dernier étant connecté à la moelle épinière. Ensemble, ces structures reçoivent, analysent et coordonnent les informations sensorielles et motrices essentielles aux fonctions vitales (AHMAD et al., 2013).

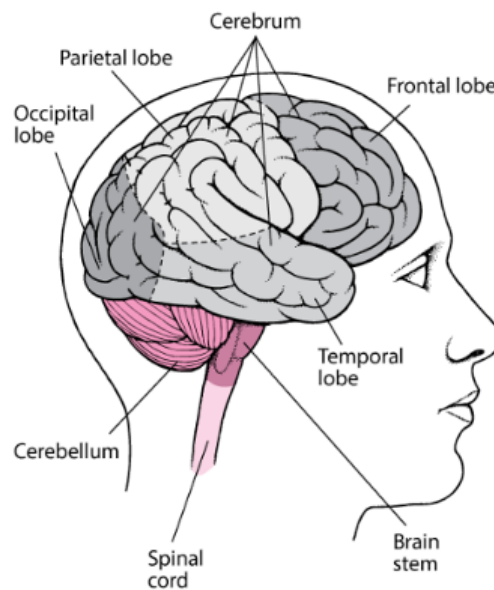


FIG. 1.5 : L'anatomie du cerveau (MERCK MANUALS, 2023)

Le cerveau est divisé en deux hémisphères, chacun contrôlant le côté opposé du corps. Chaque hémisphère est subdivisé en quatre lobes (frontal, pariétal, temporal, et occipital), jouant chacun un rôle clé dans le langage, la mémoire, le mouvement, ou encore la vision. Le cervelet, situé sous le cerveau, est responsable de la coordination des mouvements et de l'équilibre. Quant au tronc cérébral, il assure la régulation automatique de fonctions vitales comme la respiration, le rythme cardiaque ou la température corporelle .

Cette structure délicate est protégée par plusieurs couches successives : le crâne, les méninges (dure-mère, arachnoïde et pie-mère), et le liquide céphalo-rachidien (LCR), qui amortit les chocs et assure le transport des nutriments, des hormones et des déchets métaboliques.

Au niveau cellulaire, le cerveau contient deux principaux types de tissus : la matière grise, composée des corps cellulaires neuronaux (soma), qui traite et interprète l'information, et la matière blanche, formée des axones myélinisés qui transmettent rapidement les signaux entre les différentes régions cérébrales (KOZÁK, 2024).

1.8 Pathologies cérébrales

À l'instar de tous les organes du corps humain, le cerveau est aussi à l'origine de diverses affections et troubles, avec leurs causes, manifestations et soins associés comme la maladie d'Alzheimer, la maladie de Parkinson, l'épilepsie, la sclérose en plaques[9] et les tumeurs cérébrales. Cette recherche s'est focalisée sur trois catégories de tumeurs du cerveau : les gliomes, les tumeurs hypophysaires et les méningiomes (SWANBERG et al., 2022).

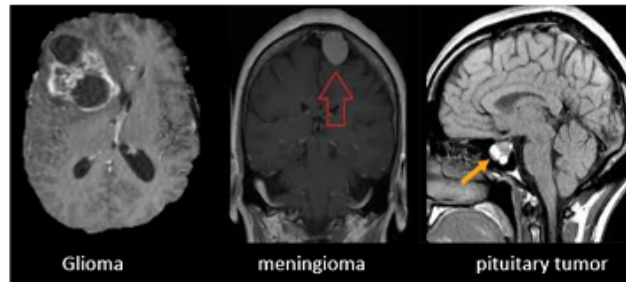


FIG. 1.6 : Tumeurs cérébrales

1.8.1 Gliomes

Les gliomes figurent parmi les formes de cancer les plus mortelles, touchant le cerveau ou la moelle épinière. C'est une forme fréquente de tumeurs cérébrales primaires, se développant majoritairement dans les cellules gliales, contrairement aux cancers métastatiques qui peuvent se former à divers endroits du corps et se répandre au cerveau (OBERHEIM BUSH & CHANG, 2016).

1.8.2 tumeurs pituitaires

Une tumeur hypophysaire est une croissance anormale localisée au niveau de l'hypophyse, une petite glande située à la base du cerveau. Bien que la majorité de ces tumeurs soient bénignes, elles peuvent affecter la sécrétion hormonale en provoquant une production excessive ou insuffisante. L'hypophyse, aussi appelée glande maîtresse, joue un rôle central dans la régulation du système endocrinien, influençant la thyroïde, les glandes surrénales et reproductrices (THAPAR et al., 2000).

1.8.3 Tumeur de méningiome

C'est une forme de tumeur qui se forme dans les méninges, ces trois enveloppes protectrices qui protègent le cerveau et la moelle épinière.

Classification du méningiome : Les méningiomes, qui présentent habituellement une nature bénigne, se distinguent par une progression lente de leur croissance. Toutefois, certains individus peuvent montrer une agressivité et une croissance plus rapide. Ainsi,

ils sont classés en trois catégories basées sur leurs propriétés histologiques (BUERKI et al., 2018).

- Grade 1 : grade faible généralement bénin présentant un taux de croissance lent.
- Grade 2 : tumeurs atypiques de niveau intermédiaire avec un taux de retour plus important après une intervention chirurgicale.
- Grade 3 : l'anaplastique est classée comme maligne (cancéreuse) et se caractérise par une croissance plus rapide.

1.9 Conclusion

Ce chapitre a permis de poser les bases de la compréhension des fondamentaux de l'image numérique et de son rôle crucial dans l'imagerie médicale. En nous concentrant sur l'imagerie par résonance magnétique (IRM), nous avons démontré sa capacité à produire des images précises, contrastées et non invasives des structures internes, en particulier du cerveau. Cette méthode, devenue essentielle dans le domaine du diagnostic neurologique, offre des perspectives diagnostiques avancées, notamment l'identification, la catégorisation et le suivi des maladies cérébrales telles que les tumeurs. Ainsi, la compréhension des concepts de l'image et l'exploitation de l'IRM constituent une étape cruciale de l'analyse médicale moderne qui allie rigueur scientifique et innovation technologique au service de la santé.

Chapitre 2

L'intelligence artificielle

2.1 Introduction

L'émergence d'Internet et la numérisation croissante de notre vie quotidienne ont transformé le monde en un vaste réseau interconnecté où chaque objet utilisé, qu'il s'agisse d'une voiture, d'un appareil électrique ou d'une plateforme de médias sociaux, génère une quantité importante de données. Cette explosion d'informations ouvre la voie à des services intelligents, rapides et personnalisés. Cependant, un défi de taille se pose face à ce flux continu de données : comment extraire des connaissances pertinentes en temps réel ? C'est dans ce contexte que l'intelligence artificielle (IA) apparaît comme un remède nécessaire. L'objectif du domaine scientifique et technologique de l'intelligence artificielle (IA) est de reproduire, à l'aide d'ordinateurs ou de logiciels, des processus cognitifs qui sont normalement exclusifs au cerveau humain. Elle englobe des capacités telles que la compréhension, la communication (avec les machines et les humains), l'adaptation et l'apprentissage. Ces systèmes informatiques sont conçus pour simuler le raisonnement humain, apprendre de l'expérience, s'adapter à de nouvelles données et prendre des décisions de manière autonome. Parfois discrète dans notre vie quotidienne, l'IA se manifeste à travers des technologies telles que la reconnaissance vocale, les assistants intelligents, les jeux vidéo et les systèmes domotiques. Ce chapitre explore les fondements conceptuels de l'intelligence artificielle, son fonctionnement général et ses principaux domaines d'application.

2.2 L'utilisation de l'intelligence artificielle

L'intelligence artificielle, qui a fait son apparition dans notre vie quotidienne depuis plusieurs années, continue de progresser de manière significative. Voici quelques exemples d'applications quotidiennes de l'intelligence artificielle :

- Les réseaux sociaux : L'intelligence artificielle joue un rôle central dans le tri et l'affichage des contenus en fonction de leur pertinence sur des plateformes comme Facebook. Elle recommande également des contenus ou produits adaptés aux préférences des utilisateurs.



FIG. 2.1 : L'intelligence artificielle

- Les applications de navigation : L'IA aide à recommander des produits aux utilisateurs, facilite les opérations d'achat et de vente en ligne, et optimise la gestion des stocks à l'aide de robots intelligents.
- Les sites marchands : L'intelligence artificielle est utilisée dans des applications comme Google Maps et Waze pour identifier les itinéraires les plus rapides et éviter les embouteillages grâce à des mises à jour en temps réel du trafic.

2.3 Les domaines d'applications de L'IA

L'intelligence artificielle, décrite comme l'intelligence démontrée par les machines, possède de multiples applications dans la société contemporaine. Elle a servi à élaborer et faire évoluer divers secteurs et industries, tels que la santé, le secteur financier et bancaire, les transports, le commerce, l'industrie, l'agriculture intelligente, les médias, le droit... et bien d'autres.

- La santé : Le domaine médical est l'un des plus dynamiques en IA. Aujourd'hui, des systèmes intelligents peuvent diagnostiquer certaines maladies, notamment les cancers, avec une grande précision. L'IA permet une détection précoce, accélère le développement de médicaments et réduit le délai de mise sur le marché.
- Le transport : Les voitures modernes utilisent l'intelligence artificielle pour des fonctions comme le stationnement automatique et la gestion du trafic, réduisant ainsi les délais, la consommation et les émissions. À l'avenir, on prévoit de développer des véhicules entièrement autonomes pour un transport plus sûr et écologique.
- Le commerce : L'intelligence artificielle transforme le commerce en automatisant diverses tâches : gestion des stocks, nettoyage, suivi des prix, encaissement, etc. Ces robots aident à pallier le manque de main-d'œuvre dans les supermarchés.

2.4 Machine Learning (ML)

Le machine learning est un domaine de l'intelligence artificielle qui se concentre sur le développement d'algorithmes et modèles basés sur des statistiques qui permettent aux ordinateurs d'identifier les schémas de l'expérience et faire des prédictions ou prendre des décisions avec une intervention humaine minimale et sans avoir besoin de la programmation explicite (GÉRON, 2019). Chaque modèle d'apprentissage automatique présente un schéma d'apprentissage particulier lorsque nous avons expliqué ci-dessous certains modèles d'apprentissage existants :

- Apprentissage supervisé : Dans l'apprentissage supervisé, les algorithmes sont formés en utilisant des données étiquetées sous forme d'entrée paires de sortie (x,y) pour déterminer la meilleure fonction de mappage f qui met en correspondance les entrées sorties $y = f(x)$ dans lesquelles les entrées sont des attributs ou des caractéristiques qui sont liées à la sortie et les résultats sont la cible ou le label d'intérêt que nous voulons que l'algorithme apprenne à prédire (LODI & ZARPELLON, 2017). La régression et la classification sont les deux principales sous-catégories de l'apprentissage supervisé
- Apprentissage non supervisé : Dans l'apprentissage non supervisé, les données de formation ne sont pas étiquetées (nous n'avons que les caractéristiques d'entrée), et les méthodes dans ces domaines tentent d'explorer les données et de trouver une structure cachée et des relations à l'intérieur, sans avoir besoin de supervision ou de connaissances préalables sur le résultat (étiquettes) (HEATON, 2018). Utiliser couramment cette technique d'apprentissage pour le clustering comme K-moyen (basé sur centroïde) et DBSCAN (basé sur la densité), réduction de dimensionnalité comme l'analyse des composants principaux (PCA), et la décomposition de valeur singulière (SVD).

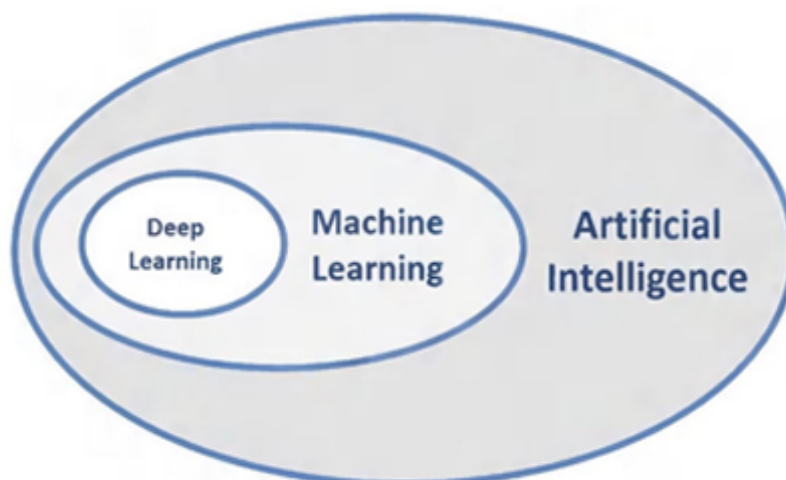


FIG. 2.2 : IA, Machine Learning et Deep Learning (CANONAR, 2023)

2.5 Apprentissage Profond (Deep Learning)

Le Deep Learning ou apprentissage profond est un type d'intelligence artificielle, dérivé du machine Learning qui a été développé dans le but de créer des algorithmes capables d'apprendre et de s'améliorer de manière autonome, contrairement à la programmation où la machine se contente d'exécuter à la lettre des règles prédéterminées. L'apprentissage en profondeur utilise une succession de couches d'unités de traitement non linéaire pour pouvoir extraire ou transformer les caractéristiques des données. La sortie d'une couche sert d'entrée de la couche suivante (REESE, 2017). Les algorithmes de l'apprentissage profond peuvent être supervisés et servir à classer les données, ou non supervisés et aider à effectuer une analyse de modèle. L'algorithme de Deep Learning absorbe des quantités de données énormes par rapport aux autres algorithmes d'apprentissages machine utilisés et développés actuellement, et il a été capable de battre les humains dans certaines tâches cognitives. Par exemple, la reconnaissance faciale par ordinateur et la reconnaissance vocale ont connu des Progrès significatifs et cela grâce aux approches d'apprentissage approfondies (LECUN et al., 2015).

2.6 Les réseaux de neurones

2.6.1 Réseau de neurones récurrents (RNN)

Réseau de neurones récurrents Le premier réseau RNN, les réseaux de Hopfield mis au point par John Hopfield, a été proposé en 1982 (HOPFIELD, 1982). il s'agit d'une classe de réseaux neuronaux qui sont naturellement adaptés au traitement des données des séries temporelles et autres données séquentielles (CHACHOUA & BENSABA, 2021). Il est largement utilisé pour résoudre un large éventail de problèmes complexes, notamment la traduction automatique, la reconnaissance vocale, l'annotation automatique d'images et les prévisions d'inventaire. RNN se distingue par son architecture unique qui diffère du reste des réseaux, où sa particularité est que sa sortie dépend non seulement de l'entrée actuelle, mais aussi de la sortie précédente. En fait, les neurones RNN peuvent retenir des informations de mémoire sur des valeurs antérieures appartenant à une séquence, ce qui permet de prédire la prochaine valeur dans la séquence. Par exemple, une personne qui lit un texte, lit les mots un à un, mais considère le sens des mots précédents, sur lesquels la compréhension du mot présent est fondée, c'est cette séquence qui permet de donner un sens à l'ensemble de la phrase. C'est un cas typique des réseaux de neurones récurrents (RNNs). Les RNNs sont appelés récurrents parce qu'ils effectuent la même tâche pour chaque élément de la séquence, et que le résultat est basé sur des calculs précédents. la composante la plus fondamentale d'un RNN est le neurone récurrent. à chaque pas de temps, le neurone reçoit une entrée standard en plus de la sortie de l'étape de temps précédente.

Fonctionnement du RNN

RNN prend le vecteur X_t comme entrée et crée le vecteur Y_t comme sortie. Toutefois, ce vecteur produit en sortie n'est pas seulement affecté par l'entrée que nous présentons en temps réel mais aussi sur l'historique d'entrée qui est alimenté avant. Chaque fois, lorsque l'étape est de connaître le RNN, la classe met à jour l'état interne. Le seul vecteur caché h présent peut être conclu comme le cas le plus simple (GUPTA, 2017). Le mécanisme illustré permet au réseau de fonctionner de différentes façons, selon la taille de la séquence d'entrée que nous présentons et la taille de la séquence de sortie que nous attendons.

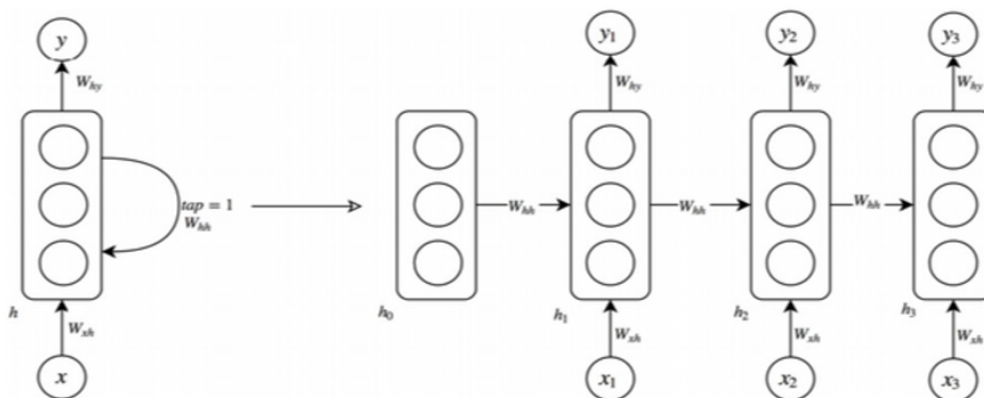


FIG. 2.3 : Réseau neuronal récurrent

2.6.2 Mémoire à long terme (LSTM)

Première proposition de Hochreiter et Schmidhuber (1997). Il s'agit d'une architecture commune pour fournir des RNNs à une unité de mémoire. LSTM est un type de RNN qui a été créé pour soulager certains des problèmes rencontrés par les RNNs traditionnels, en particulier le problème du gradient de disparition, où les gradients sont très petits ou même nuls lorsque les séquences sont grandes, bloquant ainsi l'apprentissage. Le problème est résolu grâce à l'organisation des données sauvegardées et oubliées, qui est mise en œuvre par le système de "portes" fourni par les LSTMs. Contrairement aux RNN classiques, le modèle LSTM est capable de mémoriser des informations pendant une longue période (MIAO et al., 2016). Le LSTM est constitué d'une cellule de mémoire et sa structure est illustrée. Un bloc de mémoire se compose de 3 composants principaux : la porte d'entrée, la porte d'oubli et la porte de sortie. Fondamentalement, chaque porte a sa responsabilité privée. La porte d'entrée définit quelles informations doivent être stockées dans la cellule, la porte d'oubli définit quelles informations de l'état caché précédent doivent être transmises au réseau, La porte de sortie contrôle l'information du nouvel état caché calculé qui est transmis au vecteur de sortie du réseau.

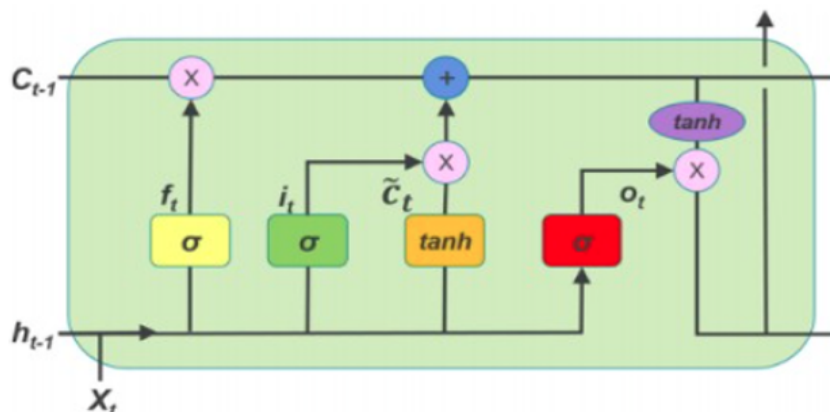


FIG. 2.4 : Structure d'une cellule LSTM (ABIRAMI & CHITRA, 2018)

2.7 conclusion

Ce chapitre a permis d'introduire les concepts fondamentaux de l'intelligence artificielle (IA), en présentant ses définitions, ses principes de base ainsi que les grandes lignes de son fonctionnement général. Nous avons exploré les usages quotidiens de l'IA et ses nombreux domaines d'application, tels que la santé, les transports.

Une attention particulière a été portée au Machine Learning (apprentissage automatique), une sous-discipline clé de l'IA, qui permet aux machines d'apprendre à partir de données. Nous avons également abordé le Deep Learning (apprentissage profond), une approche plus avancée reposant sur des réseaux de neurones profonds, qui a révolutionné plusieurs domaines comme la reconnaissance vocale et visuelle.

Par ailleurs, nous avons détaillé des architectures importantes de réseaux de neurones, notamment les Réseaux de Neurones Récurrents (RNN) et les LSTM, qui sont particulièrement adaptés au traitement des données séquentielles.

Pour le chapitre à suivre, nous allons nous intéresser aux différentes méthodes de classification, qui constituent l'un des aspects fondamentaux de la conception des systèmes intelligents.

Chapitre 3

La classification

3.1 Introduction

Les méthodes de l'apprentissage automatique sont en constante évolution. Nous allons mentionner quelques-unes des techniques couramment employées dans ce domaine. Dans le domaine de l'informatique, la classification est mise en œuvre et employée sur les données.

données numériques comme (points, tableaux, images, etc.) Dans cette section, nous allons suivre ce principe et commencer par expliquer ce qu'est la classification, ses méthodes et techniques, ainsi que son champ d'application, etc.

3.2 Définition

La classification est une spécialité appliquée dans de nombreux domaines et elle est fréquemment désignée par divers termes (classification, segmentation, clustering). Pour comprendre la définition du terme classification, il est essentiel d'examiner ses origines, qui proviennent d'un verbe. « classer », qui fait référence à plus d'une action qu'à un domaine spécifique.

Dans le domaine des mathématiques, la classification se réfère à la catégorisation d'objets. Elle implique l'attribution d'une classe spécifique à chaque objet ou individu à classer, basant cette attribution sur les données d'apprentissage préexistantes. Cette procédure fait généralement appel aux méthodes d'apprentissage. .

3.3 Les étapes d'une classification

La classification est un processus central dans l'analyse des données, visant à regrouper des objets ou des individus en catégories homogènes selon certains critères. Afin d'obtenir une classification pertinente et exploitable, plusieurs étapes clés doivent être respectées. Ces étapes permettent non seulement de structurer les données, mais aussi de garantir la fiabilité des résultats obtenus :

- Sélection des données
- Calcul de la similarité entre n individus basé sur les données de départ
- Sélection d'un algorithme de classification et sa mise en œuvre.
- L'interprétation des résultats.

3.4 Mesure de la similarité

Pour évaluer la similarité ou la correspondance entre deux entités (points, images, classes, phonème...), il est nécessaire de pouvoir quantifier leur ressemblance (ou leur différence).

Tout dispositif destiné à examiner ou structurer un ensemble de données ou d'informations doit, d'une manière ou d'une autre, faire appel à un opérateur de similarité visant à déterminer les similitudes ou les liens existants entre les informations traitées. Il convient de mentionner qu'il existe deux idées distinctes pour définir le principe de proximité entre les éléments à classer.

3.5 Les méthodes de la classification :

On distingue plusieurs types de techniques de classification basées sur les approches supervisée (classement) et non supervisée (regroupement).

3.5.1 Méthodes non supervisée

- Définition : Le clustering, également appelé segmentation, est une méthode de classification homogène qui vise à organiser un ensemble de points dans un espace donné en une série de groupes appelés « clusters ». Le regroupement consiste à établir une collection . Un ensemble d'objets semblables au sein d'un même groupe et différents lorsque ceux-ci appartiennent à des groupes distincts.
- Principe : À l'opposé de la classification (techniques supervisées), nous ne disposons pas de connaissances préexistantes concernant les catégories prédéterminées des éléments. La répartition des éléments en divers groupes (clusters) s'effectue sur la base du calcul de la similarité entre les éléments.

Le but des techniques de regroupement est de rassembler des éléments similaires dans un même ensemble, afin que deux données au sein du même groupe présentent une similarité maximale, tandis que deux éléments provenant de groupes différents affichent une dissimilarité maximale. Dans une perspective mathématique, nous avons un ensemble X constitué de N données, chaque donnée étant caractérisée par P attributs. Ainsi, le clustering vise à subdiviser cet ensemble en segments (appelés clusters)

3.5.2 Méthodes supervisée

la classification des données est l'une des tâches fondamentales qui consiste à attribuer une signification ou une catégorie à chaque information analysée. Parmi les approches utilisées pour atteindre cet objectif, on trouve ce qu'on appelle les méthodes supervisées.

- Définition : La classification est une technique supervisée qui vise à établir une fonction permettant de catégoriser les objets en différentes classes basées sur les données quantitatives et qualitatives qui caractérisent ces objets.
- Principe : Le classificateur supervisé repose sur une idée fondamentale : il a pour objectif de classer un ensemble d'éléments dont les catégories sont connues à l'avance. Chaque élément de la base d'apprentissage est donc associé à une étiquette représentant sa classe.

À partir de ces exemples annotés, le système va apprendre à reconnaître des modèles ou des régularités permettant de distinguer les différentes classes. Ce processus d'apprentissage permet de construire un modèle capable de prédire la classe d'un nouvel élément inconnu, en se basant sur les connaissances extraites des données d'apprentissage.

L'objectif principal est de minimiser les erreurs de classification en attribuant à chaque nouvel élément la catégorie la plus appropriée. Ainsi, la classification supervisée permet de généraliser l'information tirée d'un ensemble de données étiquetées afin de traiter efficacement de nouvelles données.

3.6 Quelques algorithmes de la classification supervisée

3.6.1 Arbre de décision

- Définition : Les arbres de décision sont utilisés pour catégoriser un groupe d'individus en fonction des valeurs de leurs attributs. Il s'agit d'une illustration graphique du processus de classification où : Une feuille désigne une classe, Un nœud détermine un test que doit passer un attribut spécifique, chaque branche représente une valeur possible pour cet attribut. Pour catégoriser un nouvel élément, on suit le parcours de la racine à une feuille en réalisant les diverses évaluations d'attributs à chaque nœud (HAWARAH et al., 2006) .
- Structure d'un arbre de décision : Chaque nœud interne d'un arbre de décision facilite la distribution homogène des éléments à classer entre ses différents descendants, en se basant sur une variable discriminante de ces éléments. Les liaisons entre un nœud et ses descendants, représentées par les branches, sont les valeurs distinctives de la variable du nœud. Et pour finir, les feuilles correspondent aux valeurs qui illustrent les résultats de la classification des données.

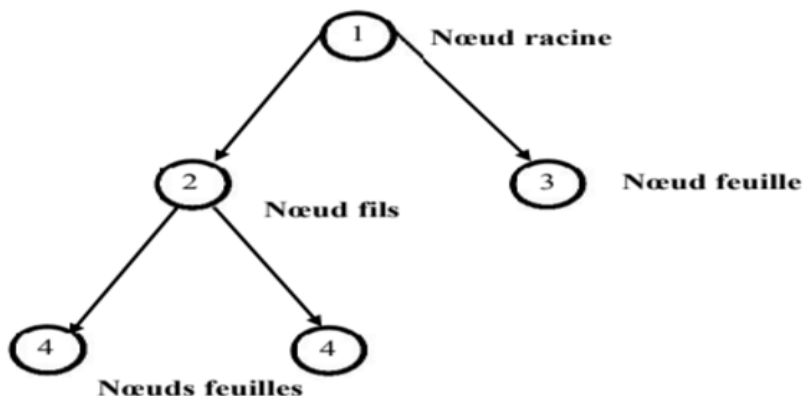


FIG. 3.1 : Structure d'un arbre de décision

3.6.2 Le classifieur naïf de Bayes

Le classificateur de Bayes naïf est un algorithme probabiliste-supervisé qui suppose que l'existence d'une caractéristique dans une classe est indépendante de l'existence d'autres caractéristiques, d'où l'utilisation de l'adjectif « naïf ». Une personne peut être considérée comme un homme si elle a un certain poids et une certaine taille. Bien que ces caractéristiques soient liées à la réalité, un classificateur bayésien conclura néanmoins qu'une personne est un homme sur la seule base de son poids et de sa taille. Malgré des hypothèses de base extrêmement simplistes, ce classificateur a donné de très bons résultats dans plusieurs cas vraiment compliqués. Une recherche publiée en 2004 a démontré qu'il existe des explications théoriques à cette efficacité inattendue (DOMINGOS & PAZZANI, 1997). Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats (RISH et al., 2001).

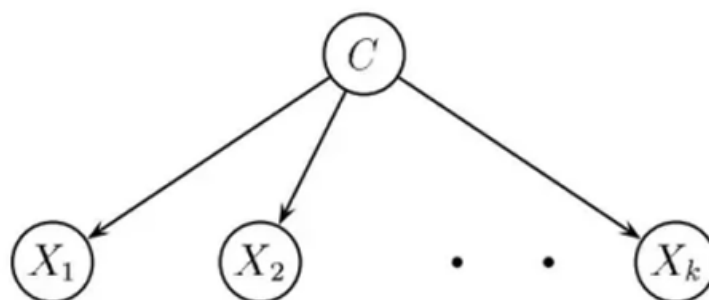


FIG. 3.2 : Structure graphique du classifieur naïf de Bayes

3.6.3 Les machines à vecteur de support (SVM)

Les SVM (Support Vector Machines), proposés initialement par Vladimir Vapnik en 1995 dans le cadre de la théorie de l'apprentissage statistique, sont des algorithmes

d'apprentissage supervisé largement utilisés pour des tâches de classification et de régression. Leur principe repose sur la recherche d'un hyperplan optimal (dans un espace de dimension n) qui sépare les données de manière maximale selon leur classe, en minimisant le risque de généralisation (principe de minimisation du risque structural) (CHEN et al., 2006).

Chaque point de données est représenté comme un vecteur dans un espace \mathbb{R}^n , et le classifieur tente de trouver un hyperplan de dimension $n - 1$ capable de séparer les classes. Les SVM ont été conçus à l'origine pour les problèmes de classification binaire, mais ont depuis été étendus à des cas multi-classes et à d'autres types de données. Ils sont notamment utilisés dans des domaines comme la reconnaissance de formes, la classification d'images et l'analyse de textures (AMARAPPA & SATHYANARAYANA, 2014).

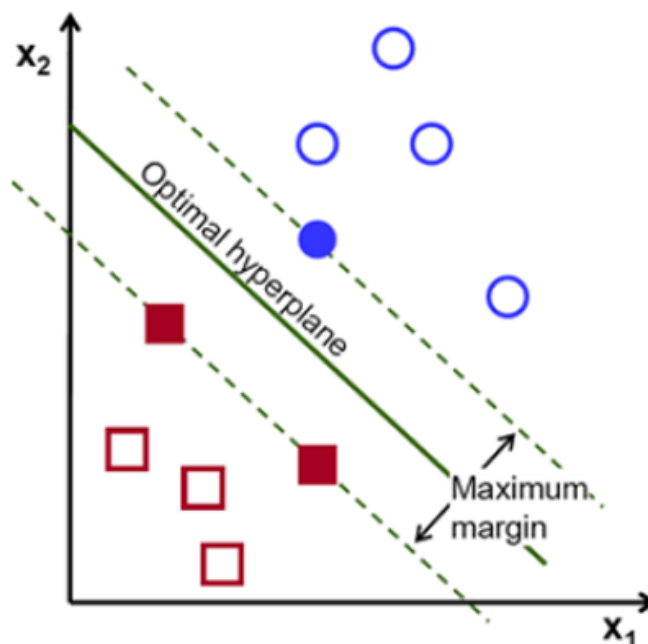


FIG. 3.3 : Principe de séparation maximale en SVM

Tous les points situés sur les bords de la marge sont appelés vecteurs de support. Lorsque les classes ne sont pas linéairement séparables, le SVM cherche à déterminer un hyperplan qui maximise la marge tout en minimisant une quantité proportionnelle au nombre d'erreurs de classification. Ce compromis entre marge maximale et erreurs de classification est contrôlé par une constante définie par l'utilisateur (VAPNIK, 2013).

Afin de traiter des jeux de données non linéairement séparables, les SVM utilisent une technique de projection dans un espace de caractéristiques de dimension plus élevée où la séparation linéaire devient possible.

3.6.4 Deep Rule-Based Classifier (DRB)

Le système DRB est une architecture de classification à plusieurs couches qui combine la puissance de l'apprentissage profond avec la transparence des systèmes de

règles floues. Il repose sur un ensemble massivement parallèle de règles floues d'ordre zéro (0-order fuzzy rules), et apprend de manière autonome à partir des données, sans nécessiter de paramètres prédéfinis ni de structure fixe. Son processus d'apprentissage est non itératif, non paramétrique, hautement parallélisable et commence « à partir de zéro », dès la première instance de chaque classe. Grâce à ses prototypes représentatifs des données, il construit une structure interprétable humainement et capable d'évoluer en continu (CIREAN et al., 2011). Ce système constitue une alternative efficace aux approches traditionnelles d'apprentissage profond, notamment dans les contextes exigeant une transparence et une adaptabilité en temps réel (SCHMIDHUBER, 2012).

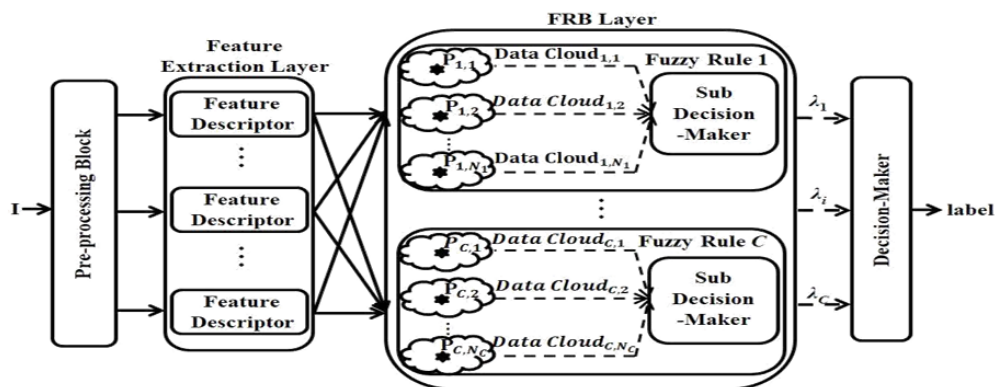


FIG. 3.4 : Architecture du classificateur DRB (P. P. ANGELOV & GU, 2018)

- Prétraitement
- Couche d'extraction des caractéristiques
- Ensemble massivement parallèle de règles SI...ALORS hautement interprétables
- Couche de décision

Le bloc de prétraitement du classificateur DRB n'implique que les techniques de prétraitement les plus fondamentales et les plus largement utilisées, à savoir : i) la normalisation, ii) la mise à l'échelle, iii) la rotation, et iv) la segmentation d'image.

Ainsi, il est, en fait, composé d'un certain nombre de sous-couches servant à des fins diverses. Le but principal du bloc de prétraitement dans le classificateur DRB est d'améliorer la capacité de généralisation du classificateur et d'augmenter l'efficacité des descripteurs de caractéristiques dans la collecte d'informations à partir de l'image. Une description plus détaillée des techniques de prétraitement que nous avons utilisées peut également être trouvée dans la section de processus de la classification.

Pour la couche d'extraction de caractéristiques, le classificateur DRB peut utiliser différents types de descripteurs de caractéristiques qui sont utilisés dans le domaine de la vision par ordinateur. Différents descripteurs de fonctionnalités présentent différents avantages. Nous avons parlé des descripteurs de caractéristiques et des différents niveaux, notamment deux types de descripteurs : l'un de bas niveau et l'autre de haut niveau, dans la section de processus de la classification.

La troisième couche du classifieur DRB est un ensemble massivement parallèle de règles SI...ALORS. C'est le « moteur » du classificateur DRB, basé sur les modèles à base de règles floues (FRB) à développement autonome du type AnYa (P. ANGELOV & YAGER, 2012). AnYa représente un ensemble de règles floues non paramétriques SI...ALORS qui ne nécessitent pas de fonction d'appartenance prédéfinie. Au lieu de cela, ces fonctions émergent automatiquement du modèle de données en suivant le concept d'analyse empirique des données (P. P. ANGELOV et al., 2017).

La structure d'une règle floue de type AnYa est représentée dans la figure (Architecture générale du classificateur DRB) . Comme on peut le voir, chaque règle floue est elle-même une disjonction (« OU ») d'un ensemble de sous-ensembles flous formés à partir de nuages de données (data clouds) associés à des prototypes respectifs. Ces prototypes sont identifiés à l'aide d'un processus d'apprentissage de type one-pass, qui peut être massivement parallélisé en considérant chaque nuage de données/prototype comme une règle floue distincte. Le décideur local fonctionne selon un opérateur « winner-takes-all ».

La dernière couche est le décideur, qui attribue l'étiquette de classe finale en se basant sur les suggestions partielles des décideurs locaux ou sous-décideurs parallèles, selon les règles SI...ALORS par classe. Cette couche n'est activée que durant la phase de validation, en appliquant également le principe du « winner-takes-all ». Par conséquent, on peut conclure que le classificateur DRB repose sur une structure de décision à deux étapes.

3.7 Présentation de notre méthodologie

Dans notre projet, nous avons adopté une approche basée exclusivement sur des techniques avancées de deep learning pour la classification des images IRM. Contrairement aux méthodes classiques fondées sur l'extraction manuelle de caractéristiques (comme HOG), notre travail repose sur des modèles capables d'apprendre automatiquement les représentations discriminantes à partir des données brutes.

Nous avons expérimenté trois architectures puissantes :

- CNN comme base de référence, connu pour sa capacité à capturer les motifs spatiaux locaux dans les images médicales.
- ResNet18, une architecture profonde avec connexions résiduelles, permettant d'atténuer le problème du gradient qui disparaît et d'améliorer l'apprentissage sur des ensembles de données complexes.
- Vision Transformer (ViT), un modèle plus récent basé sur des mécanismes d'attention, offrant une vision globale de l'image et souvent plus performant sur des tâches complexes de classification visuelle.
- Enfin, nous avons proposé une approche hybride combinant ResNet18 avec un classifieur DRB (Deep Rule-Based). Dans cette configuration, ResNet18 est utilisé comme extracteur de caractéristiques, et les représentations générées sont ensuite transmises au classifieur DRB pour effectuer la classification finale. Cette méthode vise à tirer parti de la puissance d'extraction de ResNet tout en exploitant la capacité explicative et structurée du DRB pour des décisions plus transparentes et interprétables.

Pour optimiser les performances de ces modèles, nous avons intégré des techniques d'optimisation adaptées telles que l'utilisation d'optimiseurs comme Adam, le réglage des hyperparamètres (learning rate, batch size), ainsi que des stratégies de régularisation comme le dropout et l'early stopping. Ces méthodes nous ont permis d'améliorer à la fois la précision et la généralisation des modèles tout en évitant le sur-apprentissage (overfitting).

3.8 Conclusion

Les fondements théoriques de la classification dans le contexte de l'apprentissage automatique ont été présentés dans ce chapitre. Nous avons distingué les approches supervisées et non supervisées en décrivant leurs fondements, leurs étapes essentielles et leurs algorithmes clés, tels que les arbres de décision, la classification de Bayes naïf et les machines à vecteurs de support.

Une attention particulière a été accordée au classificateur Deep Rule-Based (DRB), qui combine l'interprétabilité des systèmes à base de règles avec la force des approches d'apprentissage profond. Nous avons décrit son architecture en termes de couches, ses mécanismes de traitement et ses avantages, y compris dans le domaine de la catégorisation des images médicales.

Cette étude constitue une base solide pour comprendre les choix méthodologiques qui seront approfondis dans les chapitres suivants, notamment dans le cadre de l'application pratique sur des données médicales réelles.

Chapitre 4

Conception, Implémentation

4.1 Introduction

En raison des avancées dans les technologies d'intelligence artificielle, l'application transcendante de la classification des images médicales, en particulier des images d'imagerie par résonance magnétique (IRM), est de plus en plus réalisée avec des modèles d'apprentissage profond sophistiqués. Même avec l'abondance et la complexité de ces modèles, l'efficacité des modèles en termes de précision, de fiabilité et de dépendabilité reste un défi, notamment en ce qui concerne l'identification des algorithmes les plus adaptés pour des résultats précis et fiables.

Dans cet esprit, nous avons adopté trois modèles qui représentent différentes approches pour un but distinct concernant l'apprentissage algorithmiques : les réseaux de neurones convolutionnels (CNN), les réseaux résiduels profonds (ResNet) et le transformateur de vision (ViT).

Chaque modèle illustre une certaine approche dans le traitement d'image et dans ce chapitre, nous nous concentrons sur la fourniture des explications qui soutiennent les décisions avant de passer à l'architecture ainsi qu'aux fonctions principales pour établir les bases du travail pratique à entreprendre par la suite.

- CNN (Convolutional Neural Network)
- Res-Net18 (Residual Network)
- ViT (Vision Transformer)

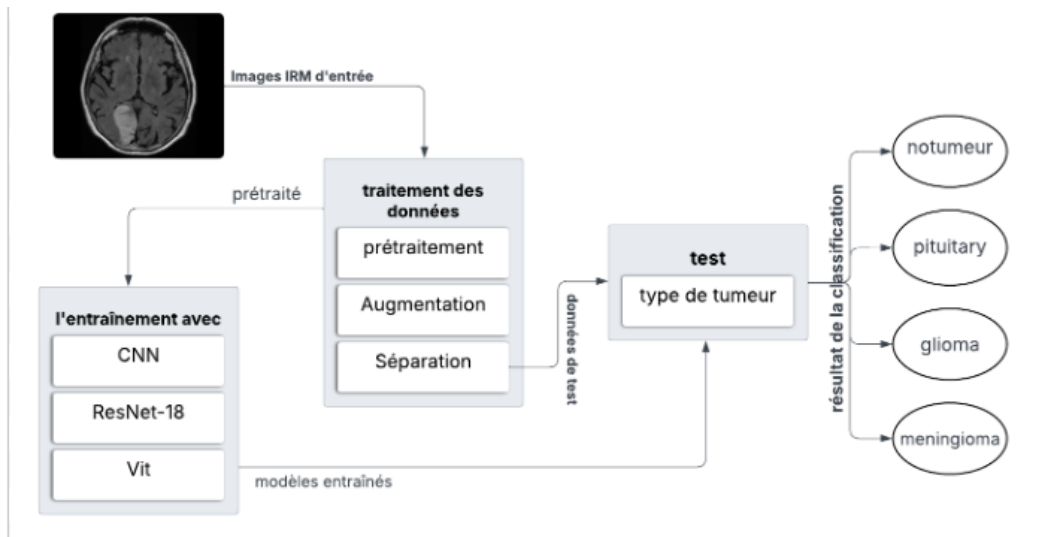


FIG. 4.1 : Architecture du système de détection et classification des tumeurs cérébrales

4.2 Explication du choix des modèles adoptés

Le choix des modèles utilisés dans cette étude est basé sur un ensemble d'approches scientifiques et pratiques destinées à capturer les progrès réalisés dans les techniques de classification d'images dans le domaine de l'intelligence artificielle.

Nous avons sélectionné le modèle de Réseau de Neurones Convolutif (CNN) comme point de départ puisqu'il est l'une des approches fondamentales dans le traitement des images médicales. Son design architectural permet des performances remarquables en extraction de caractéristiques grâce à l'application de couches convolutionnelles, et il est également relativement simple et facile à entraîner. Cela facilite la compréhension de la manière dont les architectures de réseaux de neurones les plus simples réagissent aux images IRM.

De plus, nous avons considéré que le modèle Residual Network (ResNet) constitue une version plus avancée de réseaux profonds car il a été conçu pour minimiser les effets néfastes liés à l'augmentation de la profondeur, notamment la détérioration des performances à cause de la prolifération des gradients. En permettant des connexions résiduelles, ResNet rend possible l'apprentissage dans des réseaux très profonds sans perte d'information, ce qui permet une augmentation considérable de la précision, en particulier dans les tâches difficiles telles que la classification d'images médicales.

Pour témoigner d'un changement dans les stratégies d'analyse d'image, il a été opté pour le troisième modèle, Vision Transformer (ViT). Différemment des réseaux convolutifs, ViT utilise le mécanisme d'attention auto-dirigée, qui a déjà démontré son efficacité dans le traitement des langues. Son émergence en vision par ordinateur constitue une avancée majeure, permettant d'avoir non seulement une meilleure lecture d'une image, mais également une compréhension plus interrompue et dynamique de l'image en s'adaptant sur de plus loin les différentes zones qui la composent.

Il en découle que le choix de ces trois modèles n'est pas fait de manière aléatoire. Ils

ont été choisis afin d’englober trois paradigmes différents dans le domaine de l’intelligence artificielle : le paradigme classique (CNN), le paradigme profond et optimisé (ResNet) qui est le troisième paradigme, et le paradigme moderne basé sur l’attention (ViT).

Cette variété permet de faire une comparaison juste et approfondie concernant l’exactitude, la capacité de généralisation, la vitesse d’apprentissage et, finalement, de trouver le mieux adapté pour les IRM du cervelet.

Dans cette étude, nous avons également combiné le modèle ResNet18 avec un classifieur à base de règles, le Deep Rule-Based Classifier (DRB). Cette approche vise à exploiter la capacité de ResNet18 à extraire des caractéristiques complexes tout en bénéficiant de l’interprétabilité du DRB.

ResNet18, avec ses connexions résiduelles, permet une extraction fiable des informations discriminantes. En l’associant au DRB, les décisions deviennent plus compréhensibles et transparentes, ce qui est un atout important dans le domaine médical.

Ainsi, cette combinaison permet d’obtenir de bonnes performances tout en offrant une meilleure lisibilité des résultats.

4.2.1 CNN (Le Réseau de Neurones Convolutifs)

- **Définition :**

Les réseaux de neurones convolutifs, ou CNN (Convolutional Neural Networks), représentent une classe puissante de réseaux de neurones profonds largement utilisée dans le domaine de l’analyse d’images et de la vision par ordinateur. Inspirés du fonctionnement du cortex visuel des animaux, les CNN sont spécialement conçus pour traiter des données structurées en grille, comme les images, où des dépendances spatiales locales jouent un rôle essentiel (BEYSOLOW II, 2017).

Contrairement aux réseaux neuronaux classiques, les CNN exploitent une opération mathématique appelée convolution, qui permet d’extraire automatiquement des caractéristiques discriminantes à partir des régions locales d’une image, à l’aide de filtres (ou noyaux) entraînaables. Cette capacité à apprendre les bonnes représentations sans ingénierie manuelle des caractéristiques constitue l’un de leurs plus grands atouts.

La structure typique d’un CNN comporte :

- des couches convolutionnelles pour l’extraction de caractéristiques.
- des couches de sous-échantillonnage (ou pooling) pour la réduction dimensionnelle.
- suivies de couches entièrement connectées pour la classification.

Un mécanisme clé souvent utilisé dans les architectures avancées comme ResNet est la connexion résiduelle (skip connection ou bypass), qui permet de transmettre l’information directement entre différentes couches, favorisant un apprentissage plus stable et profond.

Enfin, les CNN peuvent être appliqués non seulement à des images, mais aussi à des vidéos, des signaux audio, et d'autres types de données structurées (GU et al., 2018).

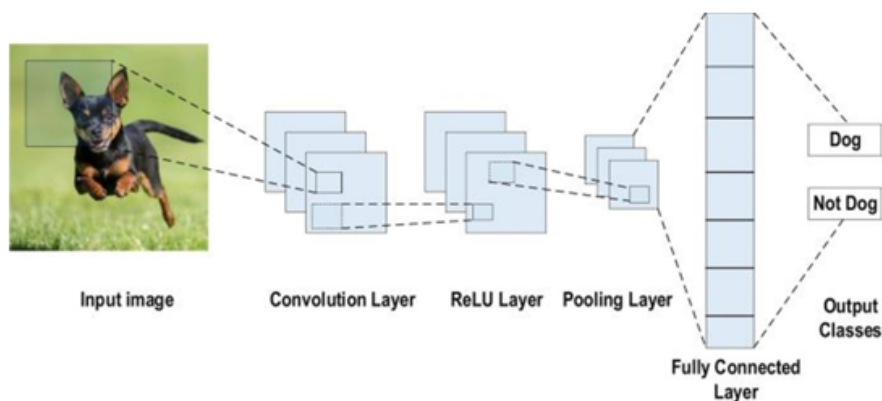


FIG. 4.2 : Architecture du Réseau de Neurons Convolutionnel (CNN) (ALZUBAIDI et al., 2021)

Principe et architecture d'un Réseau de Neurons Convolutif

Un réseau de neurones convolutif n'est pas seulement un réseau neuronal profond avec de nombreuses couches cachées. Il s'agit plutôt d'un réseau profond qui simule le fonctionnement du cortex visuel du cerveau pour reconnaître et classifier des images ou des vidéos, et pour découvrir un objet ou même une partie dans une image. Le concept et le fonctionnement des réseaux de neurones convolutifs est différent des autres réseaux de neurones, en effet un réseau neuronal convolutif comporte deux parties distinctes avec une entrée dans laquelle une image en forme de matrice de pixels bidimensionnelle (avec 2 dimensions, noir et blanc), ou une image couleur avec 3 dimensions (couleurs : rouge, vert et bleu) ou une image multidimensionnelle (image satellitaire). La première partie d'un réseau de neurones convolutif est la partie convolutionnelle qui sert à extraire les caractéristiques de l'image. Ensuite, l'image passe par le fichier de séquence de filtre, ou le noyau d'enroulement, ce qui conduit à la création d'une nouvelle image appelée cartes de convolution. Généralement, les filtres intermédiaires réduisent la résolution de l'image. Ensuite, les cartes des caractéristiques sont aplaties dans un vecteur de caractéristiques pour former les données d'entrée de la partie de couche entièrement connectée. Le rôle principal de cette couche (complètement connectée) est de combiner les caractéristiques contenues dans le vecteur de son entrée pour la classification des images. Les réseaux de neurones convolutifs (CNN), également appelés Convolutional Neural Networks, sont constitués de trois types de couches principales : les couches convolutives, les couches de regroupement et les couches entièrement connectées. Lorsqu'elles sont empilées dans un ordre spécifique, ces couches forment une architecture CNN

Couches des Réseaux de Neurones Convolutionnels

- **La couche convolutive (CONV)** : La couche convolutive est crucial pour l'analyse d'images. Elle applique des filtres à l'entrée afin d'extraire des cartes caractéristiques qui représentent des éléments visuels significatifs. Cette étape est suivie d'une fonction d'activation ReLU qui introduit la non-linéarité et améliore la distinction des pixels (CHACHOUA & BENSAHA, 2021).

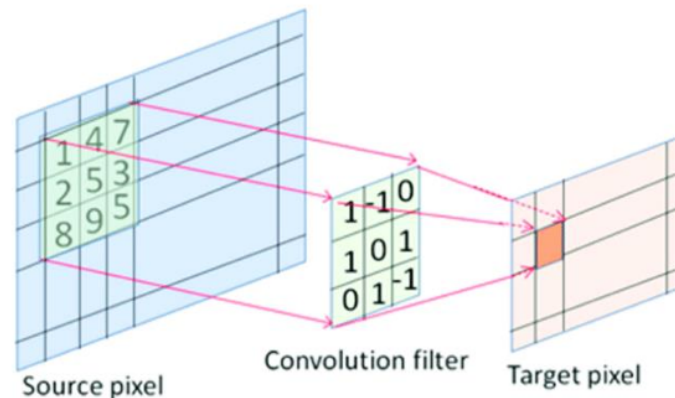


FIG. 4.3 : Opération de convolution (KIMURA et al., 2019)

- **La couche de regroupement (POOL)** : La couche de pooling est chargée de réduire la taille spatiale de la caractéristique convoluée. Cela permet de diminuer la puissance de calcul nécessaire au traitement des données en réduisant les dimensions. De plus, elle aide à extraire les caractéristiques les plus dominantes et les plus utiles. Différents types de pooling existent :

- Max pooling : Prend la valeur maximale parmi toutes les valeurs dans la fenêtre de collecte ; c'est le type le plus courant.
- Average pooling : Prend la moyenne des valeurs dans la fenêtre de collecte.

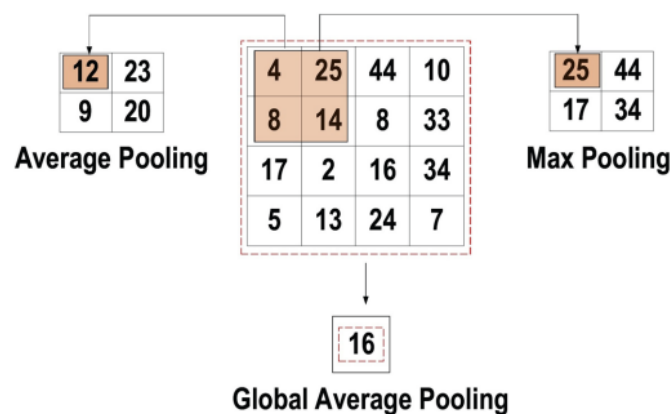


FIG. 4.4 : Types de Pooling (ALZUBAIDI et al., 2021)

- Couche Entièrement Connectée (FC)** : La couche entièrement connectée, comme son nom l'indique, est responsable de réaliser la classification en fonction des caractéristiques extraites par les couches précédentes et leurs filtres respectifs. Cette couche connecte chaque neurone de la couche précédente à chaque neurone de la couche actuelle, créant un réseau dense de connexions. En tirant parti de ces connexions, la couche entièrement connectée combine et traite les caractéristiques extraites pour faire des prédictions ou des classifications sur les données d'entrée. De cette manière, elle prend en compte l'information collective apprise des couches antérieures pour effectuer la tâche finale de classification (IBM, 2023).

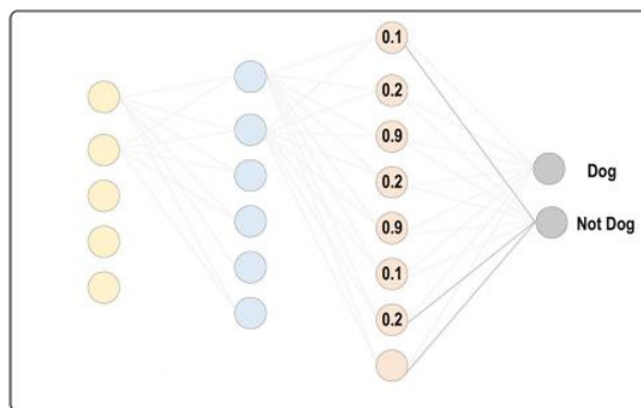


FIG. 4.5 : Couche entièrement connectée (ALZUBAIDI et al., 2021)

- Couche ReLU** : Dans cette couche, chaque valeur négative provenant de l'image filtrée est supprimée et remplacée par zéro.

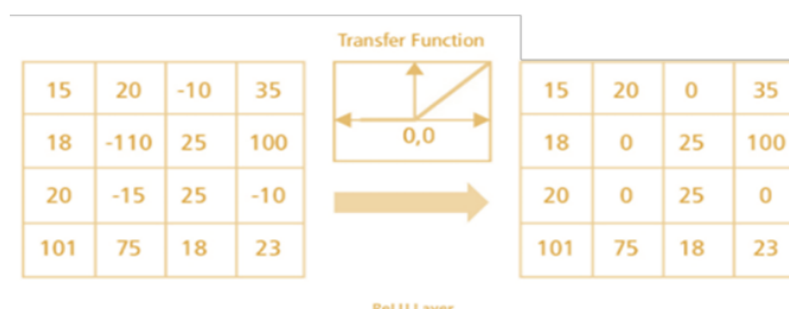


FIG. 4.6 : Opération ReLU (PRABHU, 2018)

4.2.2 ResNet18 (Residual Network)

Dans un paysage de plus en plus complexe de données visuelles, ResNet18 incarne une approche révolutionnaire qui réinvente la transmission de l'information à travers les couches neuronales. Plus qu'une simple architecture, elle introduit une philosophie inédite : 'l'apprentissage résiduel', où chaque couche apprend non pas une transformation

complète, mais seulement l'écart optimal entre entrée et sortie. Ce paradigme contre-intuitif - apprendre les différences plutôt que les absolus - permet à ResNet18 d'atteindre des profondeurs record (18 couches) tout en conservant une stabilité remarquable durant l'entraînement, marquant ainsi un tournant décisif dans l'histoire des CNN.

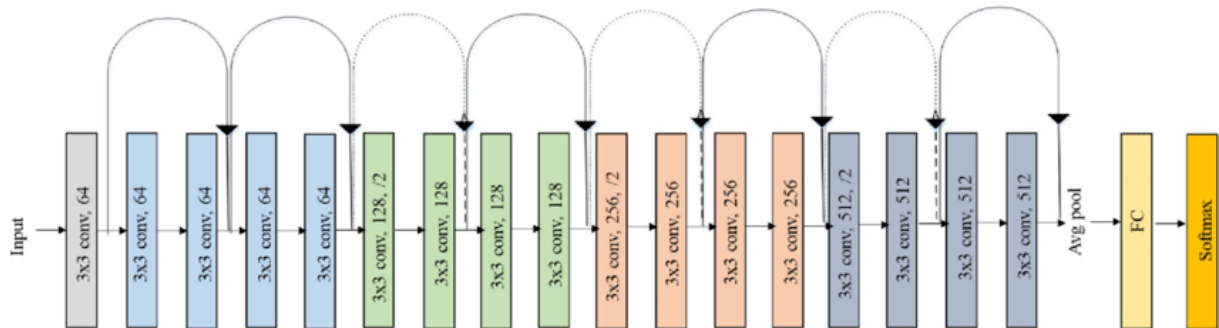


FIG. 4.7 : Original ResNet-18 Architecture (RAMZAN et al., 2019)

Introduction et Innovation

Le modèle ResNet18, développé par Microsoft Research en 2015, a révolutionné l'apprentissage profond grâce à son architecture résiduelle. Sa particularité principale réside dans l'utilisation de connexions raccourcies (skip connections) qui permettent :

- De simplifier considérablement l'entraînement des réseaux profonds
- D'atteindre des profondeurs inédites (18 couches dans ce cas)
- D'améliorer les performances de classification tout en maintenant une complexité modérée
- **Batch Normalization (Normalisation par lot)** : En normalisant les entrées de chaque couche, la technique de normalisation par lots stabilise et accélère l'apprentissage des réseaux neuronaux. Elle améliore la convergence, diminue la décroissance covariante interne et agit comme un type léger de régularisation. Elle est appliquée après chaque couche convolutive et avant l'activation ReLU dans l'architecture ResNet18, ce qui contribue à créer un environnement d'apprentissage plus stable et plus efficace.
- **ReLU (Rectified Linear Unit)** : ReLU est une fonction d'activation non linéaire largement utilisée dans les réseaux de neurones profonds. Elle est définie mathématiquement par :

$$f(x) = \max(0, x)$$

Cette fonction se distingue par sa simplicité, sa rapidité de calcul et sa capacité à atténuer le problème du gradient qui disparaît. De plus, elle induit une sparsité dans les activations, ce qui améliore l'efficacité du modèle et réduit les risques de surapprentissage.

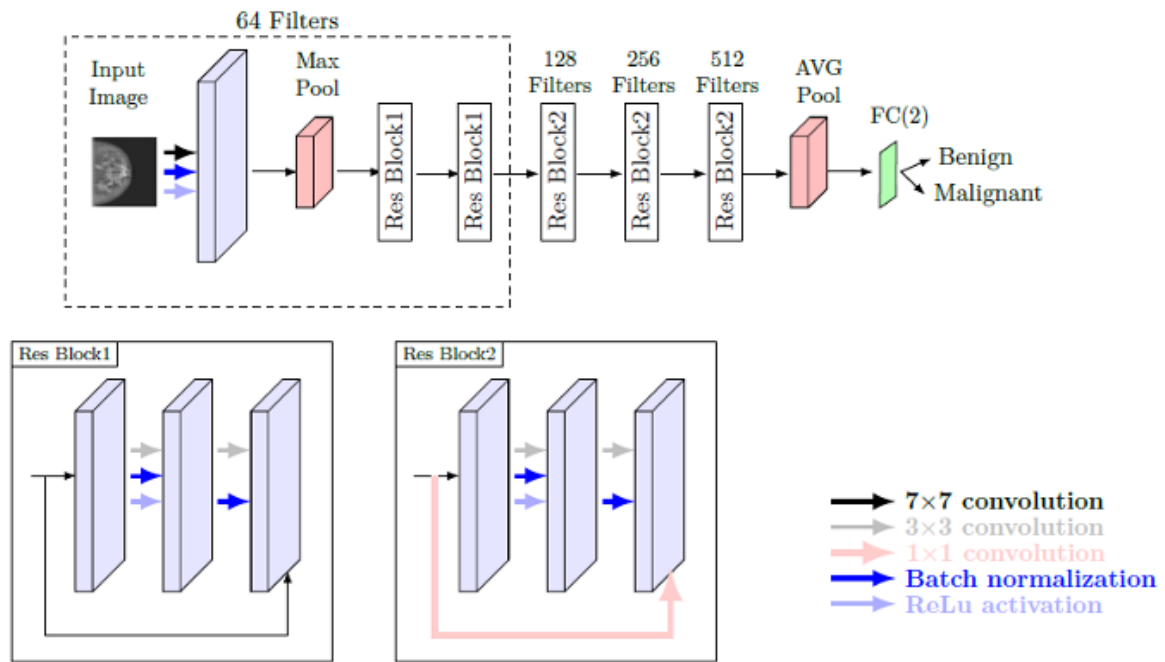


FIG. 4.8 : Architecture ResNet18 (RAMZAN et al., 2019)

- **Max Pooling (Regroupement par maximum)** : Le Max Pooling est une opération de sous-échantillonnage utilisée dans les réseaux de neurones convolutifs pour réduire les dimensions spatiales des cartes de caractéristiques, tout en conservant les informations les plus pertinentes. Elle consiste à extraire la valeur maximale dans une fenêtre locale (souvent de taille 22 ou 33).

Cette technique permet non seulement de diminuer le coût computationnel, mais aussi d'introduire une certaine invariance aux petites translations ou déformations de l'image. Dans l'architecture ResNet18, le Max Pooling est appliqué dès les premières couches afin de réduire efficacement la taille de l'image traitée.

- **Fully Connected Layer (Couche entièrement connectée)** : C'est la dernière étape du réseau, dédiée à la classification. Dans ResNet18, les caractéristiques extraites sont d'abord aplaties, puis transmises à cette couche où chaque neurone est connecté à tous ceux de la couche précédente. Elle génère les probabilités de sortie finales, généralement à l'aide d'une fonction Softmax, comme c'est le cas pour les 1000 classes d'ImageNet.
- **Interaction des composants dans ResNet18** : Le réseau commence par une couche convolutionnelle initiale, suivie d'une série de blocs résiduels. Chaque bloc applique la séquence suivante :

Convolution → Batch Normalization → ReLU → Convolution → Batch Normalization

Ensuite, l'entrée du bloc est ajoutée à sa sortie via une connexion résiduelle, suivie d'une activation ReLU.

En fin de réseau, les cartes de caractéristiques sont compressées à l'aide d'un **Average Pooling**, puis transmises à la **couche entièrement connectée** pour produire

la prédiction finale.

4.2.3 Vision Transformer (ViT)

est une architecture innovante d'apprentissage profond conçue pour traiter des données visuelles en utilisant la même architecture de transformateur qui a révolutionné le traitement du langage naturel (PNL). À la différence des réseaux de neurones convolutionnaires (CNN), qui s'appuient sur les convolutions pour capturer les caractéristiques spatiales locales, les transformateurs de vision adoptent le mécanisme d'auto attention pour modéliser les relations globales entre les zones d'image. Cette architecture a démontré des performances de pointe dans de nombreuses tâches de vision par ordinateur telles que la classification d'images, la détection d'objets et la segmentation. Le transformateur de vision s'appuie sur l'architecture du transformateur initialement introduite par Vaswani et al. en 2017 pour les tâches PNL. Les transformateurs sont très efficaces pour traiter des données séquentielles, en utilisant l'auto-attention aux dépendances de modèle entre différentes parties de l'entrée. Les transformateurs de vision appliquent cette architecture aux données d'image, traitant l'image comme une séquence de patches au lieu d'une grille de pixels (DOSOVITSKIY et al., 2020).

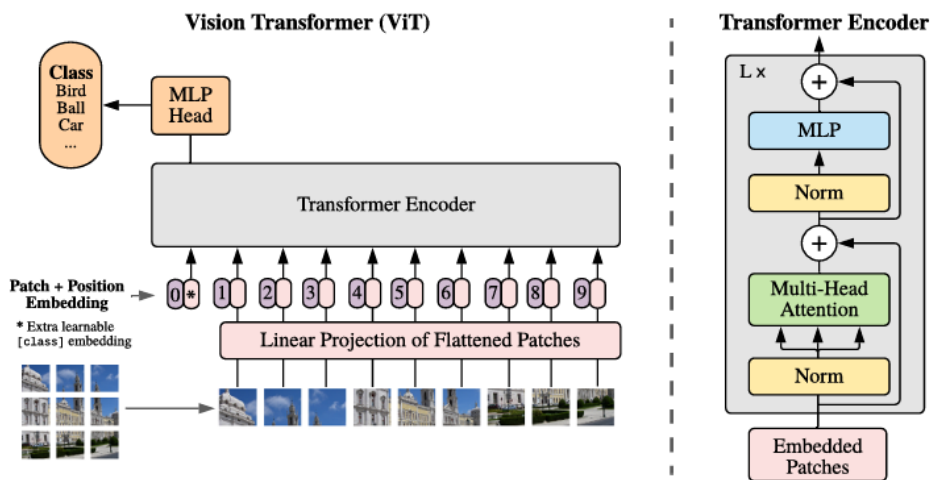


FIG. 4.9 : Architecture du transformateur de vision (ViT)(DOSOVITSKIY et al., 2020)

- Tokenisation** La tokenisation est une étape importante dans la préparation de l'image d'entrée pour le ViT. conventionnellement une image d'entrée $X \in \mathbb{R}^{H \times W \times C}$ et ses étiquettes suivantes Y , X est remodelée en une séquence de f d'images 2D lattées $X_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ où (H, W) est la hauteur et la largeur de l'image d'entrée, C est le nombre de canaux de l'image d'entrée. l'image d'entrée, C est le nombre de canaux, P représente la taille du patch et N le nombre de patches. Chaque patch est aplati puis transmis à une couche linéaire, pour être mis en correspondance avec les dimensions D en multipliant une matrice de poids pouvant être apprise. en multipliant une matrice de poids pouvant être apprise par chaque encastrement de patch

aplatis, le résultat est un encastrement dans des dimensions inférieures (XU et al., 2022). Le résultat est un encastrement dans un espace de caractéristiques de dimension inférieure, car le transformateur a une taille de vecteur latent D constante à travers toutes ses couches. Le jeton de classe est pré-apposé à la séquence de jetons, il agit comme un paramètre pouvant être appris par le modèle ViT qui s'occupe de l'apprentissage et de l'analyse de l'information. par le modèle ViT qui s'intéresse aux caractéristiques ou aux régions les plus importantes de l'image. après l'inférence des blocs d'encodage à transformateur multiple, il est transmis au MLP pour la classification finale (DOSOVITSKIY et al., 2020).

- **Position Embedding** Pour conserver l'information positionnelle relative ou absolue des embeddings des patches (tokens de la séquence), des embeddings positionnels apprenables ou pré-définis de même dimension d que le modèle sont ajoutés aux embeddings des patches VASWANI et al., 2017. Pour encoder cette information positionnelle, on adopte des fonctions sinusoidales de fréquences différentes :

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

Où pos est la position du jeton de la séquence et i représente l'indice de la dimension dans le vecteur d'intégration (VASWANI et al., 2017). Les encastresments combinés de patch et de position sont formulés de la manière suivante :

$$z_0 = [x_{cls}; x_p^1 \cdot E; \dots; x_p^N \cdot E] + [E_{pos}^{cls}; E_{pos}^1; \dots; E_{pos}^N]$$

où $x_{cls} \in \mathbb{R}^D$ est le token de classe, $E \in \mathbb{R}^{N(P^2)D}$ est la projection linéaire de chaque patch X_p , et $E_{pos}^i \in \mathbb{R}^D$ est l'embedding positionnel du i -ème token. Les entrées nouvellement formulées sont ensuite transmises aux blocs du transformeur (DOSOVITSKIY et al., 2020).

- **Attention à soi (Attention)** Le mécanisme fondamental de l'architecture Transformer est connu sous le nom de « Scaled Dot Product Attention », un concept introduit par Vaswani et al. (2017) (VASWANI et al., 2017). Les entrées consistent en un vecteur de requête et d'un ensemble de paires de vecteurs clé-valeur, il mesure le score de similarité entre chaque requête et les vecteurs clé à l'aide de la méthode du produit de points. e similarité entre chaque requête et les vecteurs clés en utilisant l'opération de produit de point. L'équation est exprimée comme suit :

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Où Q , K et V , avec d_k , désignent respectivement la dimension des vecteurs de requête et de clé, tandis que les valeurs ont une dimension d_v . Les vecteurs de clé, requête et valeur sont obtenus par transformation des embeddings initiaux des

patches à l'aide de projections linéaires avec des matrices de poids apprenables W_K , W_Q et W_V , respectivement. Ces matrices de poids sont initialisées aléatoirement au début de l'entraînement.

Le produit scalaire entre une paire requête-clé donne un ensemble de scores de similarité, ensuite mis à l'échelle par un facteur $\frac{1}{\sqrt{d_k}}$ pour éviter la saturation de la fonction Softmax (DEBENEDETTI & TRONCOSO—EPFL, 2022). Les scores ainsi normalisés sont ensuite passés à la fonction Softmax pour obtenir les scores d'attention. La somme pondérée des vecteurs valeur, avec les scores d'attention comme poids, constitue le résultat final de l'attention automatique (self-attention).

- **Attention Multi-Tête** Comme mentionné précédemment dans le mécanisme d'attention, les vecteurs de requête, de clé et de valeur sont obtenus par projection linéaire à l'aide de matrices de poids apprenables. De plus, cette projection linéaire est effectuée h fois, où h représente le nombre de têtes dans la fonction d'attention multi-tête (Multi-Head Attention). Chaque tête possède son propre ensemble de matrices de poids apprenables W_K , W_Q , et W_V , et elle effectue l'attention sur les nouvelles représentations transformées des embeddings des patches d'entrée. Toutes les têtes d'attention fonctionnent en parallèle afin de traiter l'information de manière efficace et de capturer différents aspects des données d'entrée (DEBENEDETTI & TRONCOSO—EPFL, 2022). Les matrices résultantes de ce traitement parallèle (la sortie MHSA) sont ensuite concaténées et transformées linéairement à l'aide d'une autre matrice de poids apprenable, notée W_O .

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O, \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

$$W_i^Q \in \mathbb{R}^{d_{\text{model}}d_k}, \quad W_i^K \in \mathbb{R}^{d_{\text{model}}d_k}, \quad W_i^V \in \mathbb{R}^{d_{\text{model}}d_k}, \quad W^O \in \mathbb{R}^{hd_v d_{\text{model}}}$$

Ce sont les matrices apprenables utilisées pour projeter respectivement les vecteurs d'entrée et de sortie dans l'opération d'attention.

- **Réseau Feed Forward (FFN)** Le réseau de neurones à propagation avant, connu sous le nom de Perceptron Multi-Couche (MLP), est utilisé pour traiter les résultats du calcul d'attention sur les données d'entrée. Ce réseau entièrement connecté comporte une seule couche cachée.

Formellement, étant donné une entrée x , des poids et biais apprenables W_1, b_1, W_2, b_2 , et une fonction d'activation ρ (DEBENEDETTI & TRONCOSO—EPFL, 2022), le MLP peut être exprimé comme suit :

$$\text{MLP}(x) = \rho(xW_1 + b_1)W_2 + b_2$$

4.3 Les algorithmes d'optimisation pour l'apprentissage profond

L'apprentissage profond nécessite souvent beaucoup de temps et de puissantes ressources informatiques pour effectuer l'entraînement, ce qui constitue également un obstacle majeur au développement des algorithmes d'apprentissage profond. Bien que l'on puisse recourir à l'apprentissage distribué sur plusieurs machines pour accélérer l'entraînement des modèles, cela ne permet pas pour autant de réduire les ressources de calcul requises. C'est dans ce contexte que sont apparus les algorithmes d'optimisation.

Les algorithmes d'optimisation nécessitent moins de ressources et permettent une convergence plus rapide du modèle. Ils peuvent avoir un impact fondamental sur l'apprentissage automatique en ajustant de manière optimale les hyperparamètres d'apprentissage, ce qui accélère le processus tout en réduisant la consommation de ressources.

C'est ainsi qu'est née l'algorithme Adam.

L'algorithme d'optimisation Adam est une extension de la descente de gradient stochastique (SGD = Stochastic Gradient Descent). Il est aujourd'hui largement utilisé dans les applications d'apprentissage profond, en particulier dans le domaine de la vision par ordinateur et des tâches de traitement du langage naturel.

4.3.1 L'algorithme d'optimisation Adam

L'algorithme Adam a été initialement proposé par le chercheur Diederik Kingma de OpenAI, et Jimmy Ba de l'Université de Toronto, dans un article intitulé :

Il s'agit d'un algorithme d'optimisation de premier ordre, qui peut remplacer la méthode classique de la descente de gradient stochastique (SGD). La descente de gradient stochastique démarre à partir d'un point aléatoire et progresse par pas fixes vers un minimum local, mais cela nécessite un grand nombre d'itérations en raison de son caractère aléatoire.

En revanche, Adam permet de mettre à jour les poids d'un réseau de neurones de manière répétée, en se basant sur les données d'entraînement (HOSPODARSKYY et al., 2024).

4.3.2 Avantages de l'algorithme d'optimisation Adam

L'algorithme **Adam** est l'un des algorithmes d'optimisation les plus utilisés dans le domaine de l'apprentissage profond, en raison des nombreux avantages qu'il offre. Il est notamment peu exigeant en mémoire, et ses opérations sont rapides et efficaces, ce qui le rend particulièrement adapté aux applications à grande échelle.

L'une des caractéristiques techniques majeures d'Adam réside dans sa capacité à assurer une **stabilité du gradient** grâce à la correction des variables m_t et v_t , permettant ainsi une convergence plus stable pendant l'entraînement.

Il est également bien adapté aux problèmes d'optimisation impliquant de grandes

quantités de données ou un nombre élevé de paramètres, et démontre de bonnes performances même dans des environnements bruyants ou présentant des gradients clairsemés.

De plus, Adam convient parfaitement aux tâches ayant des objectifs fixes, comme la classification d'images, où les données d'apprentissage ne changent pas au cours du processus.

Enfin, il est important de noter que cet algorithme nécessite peu d'ajustement des hyperparamètres de base, ce qui le rend pratique et efficace pour les utilisateurs (AHMED, 2020).

4.3.3 Le fonctionnement fondamental de l'algorithme d'optimisation Adam

L'algorithme **Adam** se distingue de la descente de gradient stochastique (SGD), dans laquelle un seul taux d'apprentissage (α , α) est utilisé pour mettre à jour tous les poids, sans variation durant l'entraînement.

En revanche, Adam calcule des taux d'apprentissage adaptatifs et indépendants pour chaque paramètre, en estimant la première et la seconde moments du gradient (c'est-à-dire l'espérance et la variance), selon une approche spécifique à cet algorithme.

Adam combine les avantages de plusieurs méthodes d'optimisation : AdaGrad, RMSProp et Momentum.

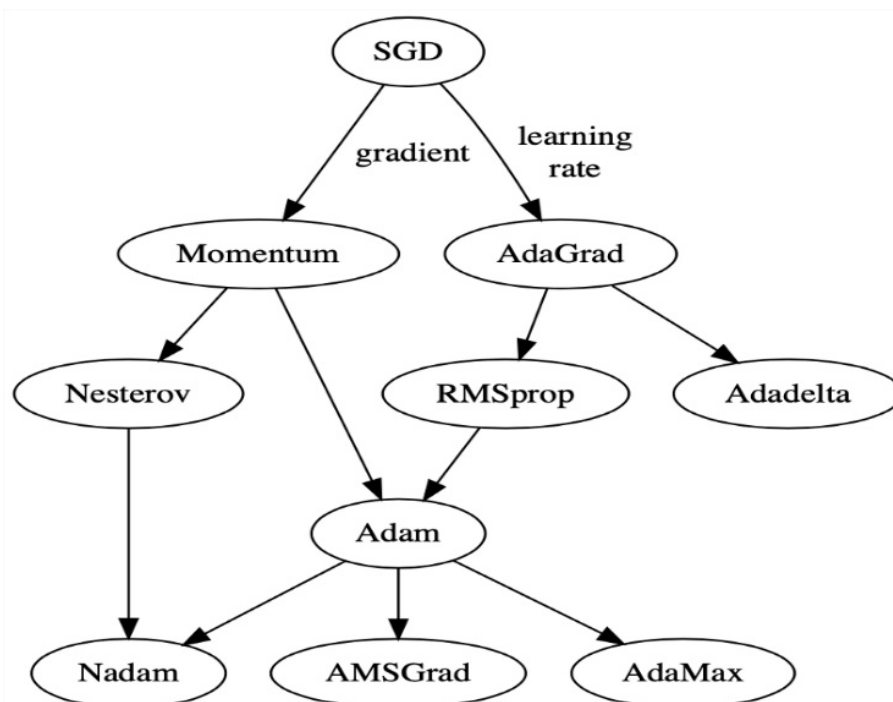


FIG. 4.10 : L'évolution des algorithmes d'optimisation (AHMED, 2020)

L'approche AdaGrad ajuste le taux d'apprentissage pour chaque paramètre θ , en permettant de grands ajustements pour les paramètres rares et de petits ajustements pour ceux fréquemment mis à jour. À chaque étape temporelle t , un taux d'apprentissage distinct est assigné à chaque paramètre.

Cependant, AdaGrad présente une limite : à mesure que les gradients s'accumulent dès le début de l'entraînement, le taux d'apprentissage peut devenir très petit, voire nul, ce qui empêche le modèle d'apprendre davantage certains paramètres.

Adam surmonte cette limitation en introduisant un mécanisme qui permet au taux d'apprentissage de se stabiliser au cours du temps, évitant ainsi une diminution excessive et assurant une convergence plus efficace (GYLBERTH, 2018).

4.3.4 Réglage des hyperparamètres

- **Alpha** α : le taux d'apprentissage ou le facteur de mise à jour, contrôle la vitesse de mise à jour des poids (par exemple $\alpha = 0,001$). Des valeurs plus grandes, comme 0,3, permettent un apprentissage initial plus rapide avant ajustement du taux, tandis que des valeurs plus petites, comme 10^{-5} , permettent une meilleure convergence mais plus lente.
- **Bêta₁** β_1 : le taux de décroissance exponentielle pour l'estimation du premier moment (la moyenne).
- **Bêta₂** β_2 : le taux de décroissance exponentielle pour l'estimation du second moment (la variance).
- **Epsilon** ϵ : un très petit nombre utilisé pour éviter la division par zéro.

Il est également possible d'appliquer une décroissance du taux d'apprentissage dans Adam. L'article original propose $\alpha = \frac{\alpha}{\sqrt{t}}$, où t représente l'itération d'entraînement (AHMED, 2020).

Valeurs par défaut recommandées pour les problèmes d'apprentissage automatique :

$$\alpha = 0,001, \quad \beta_1 = 0,9, \quad \beta_2 = 0,999, \quad \epsilon = 10^{-8}$$

4.4 Implémentation

4.4.1 Matériel Outils

Afin de mettre en œuvre ce projet, nous avons utilisé une gamme de matériaux et ses caractéristiques comme coulant :

- Nom de l'appareil : DESKTOP-BNGU3OC
- Processeur : Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 1.90 GHz
- Mémoire RAM installée : 8,00 Go (7,84 Go utilisable)

4.5 Technologies et outils utilisés

4.5.1 Google Colab

Google Colaboratory, ou « Colab », est un environnement Jupyter Notebook basé sur le cloud. Il s'exécute dans votre navigateur web. Vous pouvez écrire et exécuter du code Python, partager votre code en le modifiant avec d'autres membres de l'équipe et tout documenter en combinant du texte enrichi, des diagrammes, des images, du HTML et LaTeX dans un seul bloc-notes (ZEMAN, 2023).



FIG. 4.11 : Interface de Google Colab

Environnement de développement et ressources matérielles :

Parmi les GPU disponibles sur Colab, nous avons principalement utilisé le GPU NVIDIA Tesla T4. Il s'agit d'une carte graphique basée sur l'architecture Turing, dotée de 2560 cœurs CUDA et de 16 Go de mémoire GDDR6. Ce type de GPU est particulièrement adapté aux charges de travail liées à l'intelligence artificielle, notamment l'entraînement et l'inférence de modèles d'apprentissage profond. L'utilisation du GPU T4 a permis :

- Une réduction significative du temps d'entraînement, en comparaison avec une exécution sur CPU.

- Une meilleure gestion des grands volumes de données.
- Une exécution fluide des processus intensifs comme l'optimisation, la rétropropagation et la validation croisée.

Cette infrastructure a donc offert un cadre performant, reproductible et flexible pour le développement, l'entraînement et l'évaluation des différents modèles mis en œuvre dans ce projet.

4.5.2 Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source largement utilisé, développé par Microsoft. Connu pour sa conception légère et ses fonctionnalités puissantes, VS Code prend en charge le débogage, la coloration syntaxique, la complétion intelligente du code ainsi que le refactoring. Cet éditeur est hautement personnalisable, permettant aux utilisateurs d'installer des extensions qui améliorent ses fonctionnalités. Sa polyvalence le rend adapté à divers langages de programmation et tâches de développement. Dans l'enquête annuelle Stack Overflow des développeurs de 2024 (XU et al., 2022), VS Code a été reconnu comme l'outil de développement le plus populaire, avec 73,6



FIG. 4.12 : logo Visual Studio Code

4.5.3 Langage de programmation Python

Python a été développé en 1991 aux Pays-Bas, conçu par Guido van Rossum. Il s'inspire principalement du langage de programmation ABC et du langage C. Python est un langage dit de haut niveau, c'est-à-dire qu'il s'éloigne du langage machine plus facile à lire et à écrire, et plus proche du langage humain. Python est également un langage portable qui peut être exécuté de la même manière sur un ordinateur Windows, Mac OS ou GNU/Linux, ainsi que sur un système d'exploitation mobile comme Android ou IOS (GEEKSFORGEEKS, 2024).



FIG. 4.13 : Logo Python

- **Pandas** : Pandas est une bibliothèque de langage de programmation Python spécifiquement pour la science des données. Développé en 1991, c'est le langage de programmation le plus populaire pour l'analyse de données et l'apprentissage automatique. Pandas est conçu pour la manipulation et l'analyse des données (ROBERT, 2022).



FIG. 4.14 : Pandas logo

- **Matplotlib** : Matplotlib est une bibliothèque de traçage disponible pour le langage de programmation Python en tant que composant de NumPy, une ressource pour le traitement numérique des données volumineuses. Matplotlib utilise une API objectorientée pour intégrer des diagrammes dans des applications Python (ROUSE, 2019a).

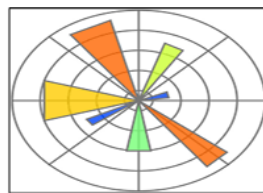


FIG. 4.15 : Matplotlib logo

- **PIL Python** : Pillow est une bibliothèque open-source, une bibliothèque d'imagerie Python qui améliore les capacités de traitement des images de notre interpréteur Python. La bibliothèque prend en charge plusieurs fonctions telles que l'ouverture, l'édition et l'enregistrement d'images dans différents formats (PILLOW CONTRIBUTORS, 2025).



FIG. 4.16 : PIL logo

- **NumPy** : NumPy est une bibliothèque Python pour manipuler des tableaux. Elle possède également des fonctions pour travailler avec l'algèbre linéaire, les transfor-

mations de Fourier et les matrices. Il s'agit d'un projet open-source et vous pouvez l'utiliser gratuitement. NumPy signifie Numerical Python (W3SCHOOLS, 2025).



FIG. 4.17 : NumPy logo

- **Scikit-learn** : Scikit-learn est une bibliothèque de clés pour le langage de programmation Python. Scikit-learn se concentre sur les outils d'apprentissage automatique, y compris les algorithmes mathématiques, statistiques et à usage général qui forment la base de nombreuses techniques d'apprentissage automatique (ROUSE, 2019b).



FIG. 4.18 : Scikit-learn logo

- **seaborn** : Une bibliothèque de visualisation statistique basée sur matplotlib. Cas d'utilisation :
 - Heatmaps, violons, pairplots (`sns.heatmap`)
 - Visualisations statistiques avancées



FIG. 4.19 : Seaborn logo

- **PyTorch** : Un framework open-source pour le deep learning (développé par Facebook). Cas d'utilisation :
 - Réseaux neuronaux (`torch.nn`)
 - Accélération GPU, différentiation automatique (`torch.Tensor`)
- **glob** : Un module intégré pour la recherche de fichiers avec des motifs. Cas d'utilisation :



FIG. 4.20 : PyTorch logo

- Recherche de fichiers avec des wildcards (*.txt)
- **transformers** : Une bibliothèque pour les modèles NLP de pointe (BERT, GPT, etc.). Cas d'utilisation :
 - Classification de texte, traduction, résumé
 - Modèles pré-entraînés (from transformers import BertModel)

Package	Utilisation Principale
os	Interactions avec le système d'exploitation
Pillow	Traitement d'images
pandas	Manipulation & analyse de données
matplotlib	Visualisations de base
seaborn	Visualisations statistiques
glob	Recherche de fichiers par motif
numpy	Calcul numérique
scikit-learn	Machine Learning
PyTorch	Deep Learning
transformers	NLP (BERT, GPT, etc.)

TAB. 4.1 : Description des principaux packages Python

4.5.4 Dataset

Le jeu de données utilisé dans notre projet est divisé en deux sous-ensembles principaux : **training** et **testing**, chacun contenant des images IRM classées en quatre catégories : *gliome*, *méningiome*, *pituitaire* et *sans tumeur (no tumor)*. Cette répartition permet de garantir une évaluation fiable des performances du modèle en séparant clairement les données d'apprentissage de celles de test.

- Le sous-ensemble **training** contient un total de 5 712 images, réparties comme suit
 - 1 321 images de gliome
 - 1 339 images de méningiome
 - 1 457 images de tumeurs pituitaires
 - 1 595 images sans tumeur
- Le sous-ensemble **testing** comprend un total de 1 311 images, réparties comme suit
 - 300 images de gliome
 - 306 images de méningiome
 - 300 images de tumeurs pituitaires
 - 405 images sans tumeur

Cette distribution équilibrée permet d’assurer une diversité suffisante dans les classes, ce qui est essentiel pour entraîner un modèle robuste et généraliser efficacement sur des données inconnues (NICKPARVAR, 2025).

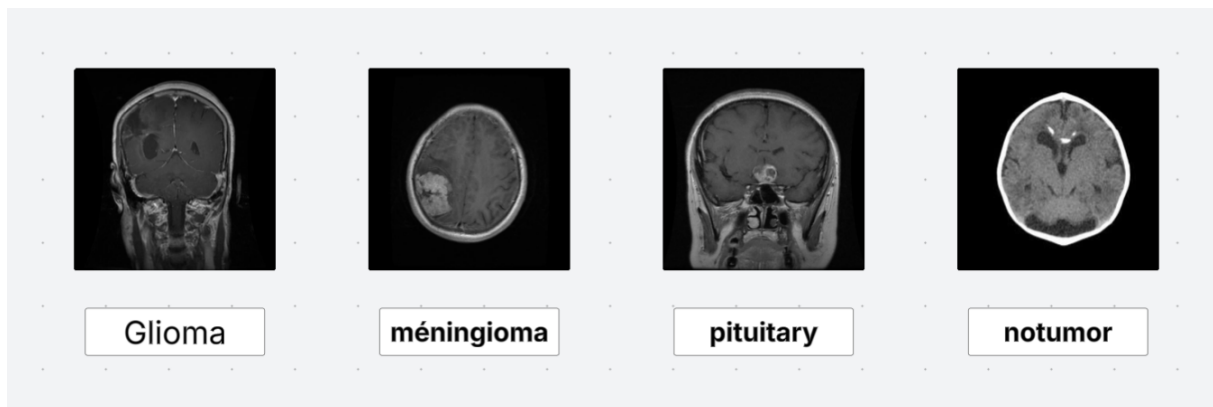


FIG. 4.21 : Exemples d’images IRM par classe dans le dataset

4.6 Mesures d’évaluation des algorithmes de classifications

Matrice de confusion multi-classe

La matrice de confusion indique le nombre de prédictions correctes pour chaque classe et le nombre de prédictions incorrectes pour chaque classe, organisée en fonction de la classe prédite. Chaque ligne du tableau correspond à une **classe prédite**, et chaque colonne correspond à une **classe réelle**.

Dans les lignes sous les classes réelles, les prédictions ou les résultats sont inscrits. Ces résultats peuvent être l’indication correcte d’une prédiction positive comme vraie positive (true positive) et d’une prédiction négative comme vraie négative (true negative), ou

		Classes prédictives			
		Classe 1	Classe 2	...	Classe n
Classes Effectives	Classe 1	X_{11}	X_{12}	...	X_{1n}
	Classe 2	X_{21}	X_{22}	...	X_{2n}
	⋮	⋮	⋮	⋮	⋮
	Classe n	X_{n1}	X_{n2}	...	X_{nn}

TAB. 4.2 : Matrice de confusion multi-classe

une prédiction positive incorrecte comme fausse positive (False positive) et une prédiction négative incorrecte comme fausse négative (false negative).

Le nombre total de faux négatifs (TFN), et le nombre total de faux positifs (TFP) et le nombre total de vrais négatifs (TTN) pour chaque classe sera calculé sur la base des équations généralisées 3, 4 et 5, respectivement. Le total des vrais positifs dans le système sera obtenu par l'équation 6.

Précision (P)

Cette mesure d'évaluation décrit le taux de réussite par rapport à une classe. Pour calculer la précision généralisée (P), on applique l'équation suivante :

$$P_i = \frac{TP_{tout}}{TP_{tout} + TFP_i} \tag{4.1}$$

Recall (True positive rate sensitivity)

Le principe de la mesure d'évaluation *Recall* (R) est très simple : de toutes les classes positives, combien le modèle a prédit correctement. Il doit être aussi élevé que possible. Il est exprimé par l'équation suivante :

$$R_i = \frac{TP_{tout}}{TP_{tout} + TFN_i} \tag{4.2}$$

F-Measure

Il est difficile de comparer deux modèles avec une faible précision et un *Recall* élevé ou vice versa. Donc, pour les rendre comparables, nous utilisons la **F-measure**. La **F-measure** permet de mesurer simultanément le *Recall* et la précision. Elle utilise la moyenne harmonique à la place de la moyenne arithmétique en punissant davantage les valeurs extrêmes. Elle est donnée par l'équation suivante :

$$F\text{-Measure} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4.3)$$

Accuracy

Le principe de cette mesure est de toutes les classes, combien le modèle a prédit correctement. Elle est définie par la formule suivante :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.4)$$

4.7 Implémentation de code et explication

```
def train_model(model, train_loader, val_loader, criterion, optimizer, num_epochs=10, name='model', patience=7):
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    model.to(device)

    best_val_loss = float("inf")
    tolerance = 0 # Counter for early stopping
    history = {'train_loss': [], 'train_acc': [], 'val_loss': [], 'val_acc': []}
```

Dans cette section du code, nous préparons l'environnement pour former le modèle. Tout d'abord, le lecteur de carte / ordinateur est automatiquement détecté : s'il y a un GPU disponible (carte graphique), il sera utilisé pour accélérer l'entraînement ; sinon, le CPU sera utilisé automatiquement. Ensuite, il est configuré que le modèle soit traduit sur cette machine (cuda ou cpu) afin que tous les calculs soient effectués correctement.

Nous allons ensuite initialiser une variable avec une valeur d'infini, qui sera utilisée pour garder la trace de la meilleure performance (la plus faible perte de validation) obtenue pendant l'entraînement, ce qui est important pour la stratégie d'arrêt précoce.

Une valeur de tolérance est également discutée. À chaque fois que les performances du modèle ne s'améliorent pas sur l'ensemble de la validation, une valeur de tolérance peut être spécifiée pour arrêter la formation après un certain nombre d'échecs non concluants afin d'améliorer les résultats.

Enfin, un dictionnaire historique est en place pour enregistrer l'évolution des métriques de performance au fil des époques : la perte (loss) et la précision (accuracy) pour les ensembles d'entraînement et de validation. Ces informations seront utiles pour visualiser et analyser la progression du modèle après l'entraînement.

Le modèle Brain Tumor CNN est un réseau de neurones convolutionnaires (CNN) conçu pour la classification d'images IRM de 4 classes du cerveau, constitué de 5 blocs convolutés soutenus par un clustering (pooling), la réglementation et enfin un contrôleur de classification final.

```
class BrainTumorCNN(nn.Module):
    def __init__(self, num_classes=4):
        super(BrainTumorCNN, self).__init__()

        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)
        self.dropout = nn.Dropout(0.5)

        # Conv Block 1
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, padding=1) # (B, 32, 224, 224)

        # Conv Block 2
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, padding=1) # (B, 64, 112, 112)

        # Conv Block 3
        self.conv3 = nn.Conv2d(64, 128, kernel_size=3, padding=1) # (B, 128, 56, 56)

        # Conv Block 4
        self.conv4 = nn.Conv2d(128, 256, kernel_size=3, padding=1) # (B, 256, 28, 28)

        # Conv Block 5
        self.conv5 = nn.Conv2d(256, 256, kernel_size=3, padding=1) # (B, 256, 14, 14)
```

Analyse détaillée du modèle

- **Blocs convolutifs (1 à 5)** : Chaque bloc est composé d'une couche de convolution 2D, suivie d'une fonction d'activation ReLU et d'un max pooling. Le nombre de filtres augmente progressivement de 32 à 256, permettant au modèle d'apprendre des caractéristiques de plus en plus complexes.
- **Activation ReLU** : Introduit de la non-linéarité pour permettre au réseau de modéliser des relations complexes entre les pixels.
- **MaxPooling (2x2, stride=2)** : Réduit la taille spatiale des cartes de caractéristiques, tout en conservant les informations essentielles. Cela aide également à réduire le coût computationnel.
- **Dropout (0.5)** : Une régularisation est appliquée en désactivant aléatoirement 50% des neurones pendant l'entraînement afin de limiter le surapprentissage.
- **Classificateur final** : Les couches entièrement connectées (linéaires) ne sont pas encore présentes dans le code. Elles seront ajoutées après l'aplatissement des cartes de caractéristiques pour effectuer la classification finale.

Dans cette partie du code, nous modifions la dernière couche entièrement connectée du modèle `model2`, qui est probablement un modèle pré-entraîné tel que ResNet18.

Tout d'abord, le nombre de caractéristiques en sortie de la couche précédente est extrait à l'aide de la variable `num_fts`. Ensuite, une nouvelle architecture est créée à l'aide de `nn.Sequential`, contenant les éléments suivants :

- Une couche linéaire (`Linear`) qui réduit le nombre de caractéristiques à 512 unités,
- Une fonction d'activation ReLU pour introduire de la non-linéarité,

```
▶ num_ftrs = model2.fc.in_features
  model2.fc = nn.Sequential(
    nn.Linear(num_ftrs, 512),
    nn.ReLU(),
    nn.Dropout(0.5),
    nn.Linear(512, 4)
  )
```

- Une couche de régularisation `Dropout` avec un taux de 0,5 pour limiter le surapprentissage,
- Une couche linéaire finale produisant 4 sorties correspondant aux 4 classes de notre tâche de classification.

Cette modification permet d'adapter le modèle pré-entraîné à notre propre jeu de données et à notre problème de classification spécifique.

```
[ ] from transformers import ViTForImageClassification, ViTFeatureExtractor

[ ] num_classes = NB_CLASSES

▶ model = ViTForImageClassification.from_pretrained(
  "google/vit-base-patch16-224",
  num_labels=num_classes,
  ignore_mismatched_sizes=True
)

🔄 config.json: 100% ██████████ 69.7k/69.7k [00:00<00:00, 4.54MB/s]
  model.safetensors: 100% ██████████ 346M/346M [00:03<00:00, 76.5MB/s]
```

Dans cette section du code, nous importons un modèle appelé **ViT (Vision Transformer)** pour la classification des images en utilisant la bibliothèque `transformers` de Hugging Face.

Le modèle utilisé est `google/vit-base-patch16-224`, un modèle pré-entraîné sur un large corpus d'images. Nous l'utilisons pour notre tâche de classification d'images IRM, en précisant le nombre de classes à prédire via le paramètre `num_labels`, qui correspond à `NB_CLASSES` dans notre code.

L'option `ignore_mismatched_sizes=True` permet de remplacer la couche de sortie du modèle afin qu'elle corresponde au nouveau nombre de classes, même si celui-ci diffère de celui utilisé lors du pré-entraînement. Cela évite ainsi toute erreur liée à une incompatibilité des dimensions.

4.8 Conclusion

Dans ce projet, nous avons conçu, implémenté et évalué plusieurs architectures de réseaux de neurones profonds afin de classifier efficacement les images de tumeurs cérébrales. En commençant par une explication du choix des modèles, nous avons exploré des architectures classiques et avancées telles que les **CNN**, **ResNet18** et **Vision Transformer (ViT)**, chacune apportant des avantages spécifiques pour la détection et l'analyse des caractéristiques visuelles dans les images médicales.

Le **CNN** a servi de base à notre étude, en exploitant les couches convolutives pour extraire les caractéristiques essentielles. Le **ResNet18**, avec ses connexions résiduelles, a permis de surmonter le problème de dégradation rencontré dans les réseaux profonds. Quant au **ViT**, il a mis en évidence les capacités puissantes des modèles à base de transformeurs pour capturer des relations complexes entre les pixels, grâce à la mécanique d'attention.

En parallèle, nous avons accordé une attention particulière au choix des **algorithmes d'optimisation**. L'optimiseur **Adam** s'est révélé être une solution efficace, combinant rapidité de convergence et stabilité, tout en nécessitant peu d'ajustements d'hyperparamètres. Ce choix a été essentiel pour renforcer la performance de nos modèles.

L'**implémentation** pratique s'est appuyée sur des outils modernes et accessibles tels que **Google Colab** et le langage **Python**, facilitant l'expérimentation rapide, le traitement des données et l'exploitation des puissantes bibliothèques de deep learning.

Enfin, une **évaluation rigoureuse** des performances a été réalisée à l'aide de métriques standards telles que la **précision**, le **rappel**, la **F-mesure** et l'**exactitude**, ce qui nous a permis d'analyser en détail la robustesse et la fiabilité de chaque modèle.

En conclusion, cette synergie entre modèles performants, optimisation intelligente, environnement de développement adapté et évaluation méthodique nous a permis de bâtir un système prometteur pour la classification des tumeurs cérébrales, ouvrant la voie à de futures améliorations dans le domaine du **diagnostic médical assisté par l'intelligence artificielle**.

partie II

Étude comparative

Chapitre 5

Évaluation

5.1 Introduction

Évaluer la performance des modèles est une étape essentielle dans tout processus d'apprentissage automatique. Elle permet de mesurer dans quelle mesure un modèle est capable de généraliser ses prédictions sur des données qu'il n'a jamais vues auparavant. Une bonne évaluation aide à comparer différents modèles, détecter les cas de surapprentissage (overfitting) ou de sous-apprentissage (underfitting), et guider le choix des paramètres ou des algorithmes à utiliser. Évaluer la performance des modèles est une étape essentielle dans tout processus d'apprentissage automatique. Elle permet de mesurer dans quelle mesure un modèle est capable de généraliser ses prédictions sur des données qu'il n'a jamais vues auparavant. Une bonne évaluation aide à comparer différents modèles, détecter les cas de surapprentissage (overfitting) ou de sous-apprentissage (underfitting), et guider le choix des paramètres ou des algorithmes à utiliser. Parmi les métriques les plus couramment utilisées, on trouve :

- **Accuracy.**
- **Précision.**
- **Recall.**
- **F-Measure.**
- **La matrice de confusion.**

5.2 CNN (Convolution Neural Network)

5.2.1 Avec optimisation

Afin d'évaluer la performance du modèle CNN optimisé, nous avons effectué un entraînement sur les données d'IRM cérébrales classées en quatre types de tumeurs : pituitary, glioma, no tumor et meningioma.

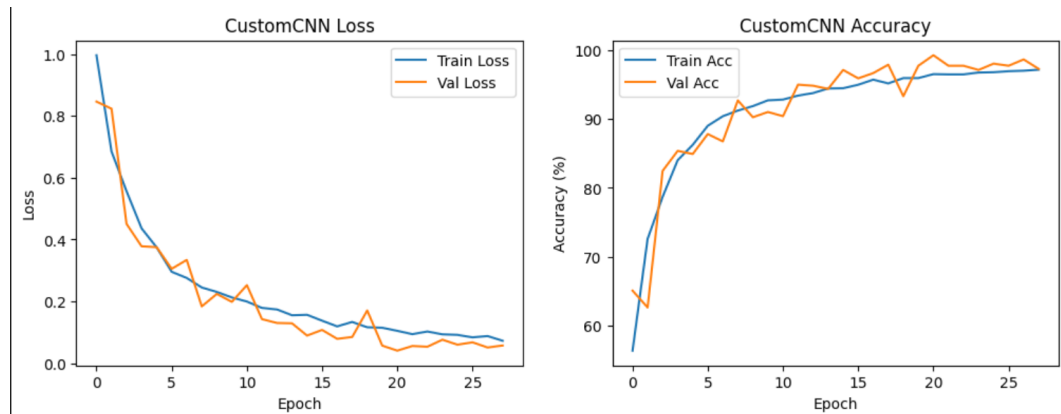


FIG. 5.1 : Courbes de performance du modèle CNN avec optimisation

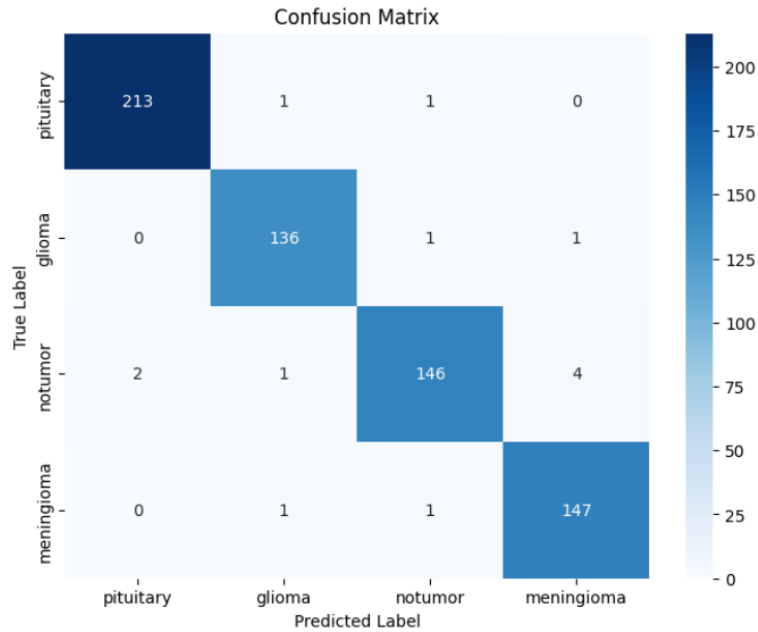
Analyse des courbes d'apprentissage : Les deux graphiques illustrent une convergence rapide du modèle, où la perte diminue de manière significative dès les premières époques, tandis que l'accuracy de validation atteint environ 98%. Cette performance suggère un bon apprentissage sans sur-apprentissage majeur.

- **Rapport de classification :** Le tableau ci-dessous montre les métriques principales pour chaque classe : précision, rappel, F1-score et support. On remarque une très bonne performance globale, en particulier pour la classe pituitary avec une précision de 99 pour cent.

	precision	recall	f1-score	support
pituitary	0.99	0.99	0.99	215
glioma	0.98	0.99	0.98	138
notumor	0.98	0.95	0.97	153
meningioma	0.97	0.99	0.98	149

TAB. 5.1 : Rapport de classification CNN-A

Analyse de la matrice de confusion : La matrice de confusion confirme que le modèle est capable de bien distinguer entre les différentes classes. On observe quelques confusions mineures entre les classes no tumor et glioma, ainsi qu'entre meningioma et no tumor, mais leur impact reste faible sur la performance globale.



Résumé des résultats : Exactitude globale (Accuracy) : 98.02% Meilleure classe prédite : pituitary Erreurs les plus fréquentes : confusions entre glioma/no tumor et no tumor/meningioma

5.2.2 Sans optimisation

Les courbes d'apprentissage révèlent une stagnation des performances du modèle. La courbe de perte reste élevée et irrégulière, indiquant une difficulté d'apprentissage. De plus, l'accuracy de validation stagne autour de 25%, ce qui correspond à une prédiction aléatoire pour un problème à quatre classes.

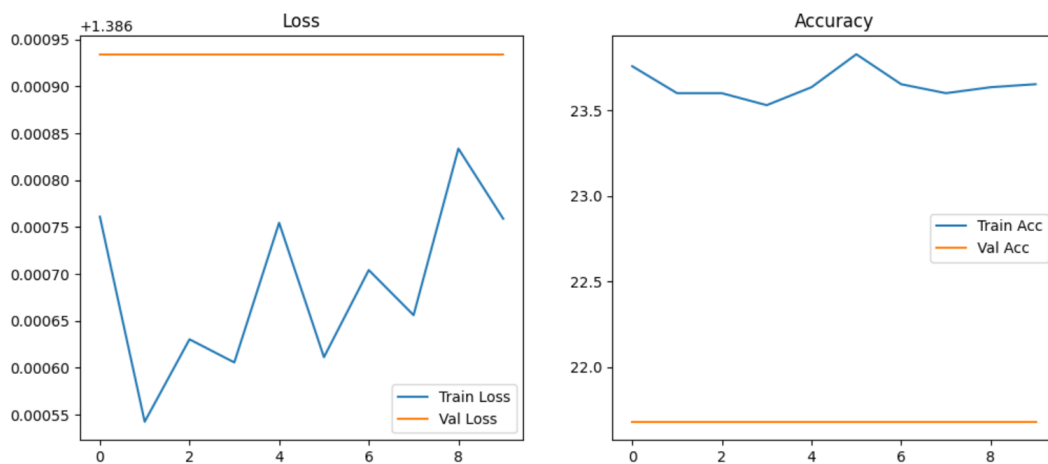


FIG. 5.2 : Courbes de performance du modèle CNN sans optimisation

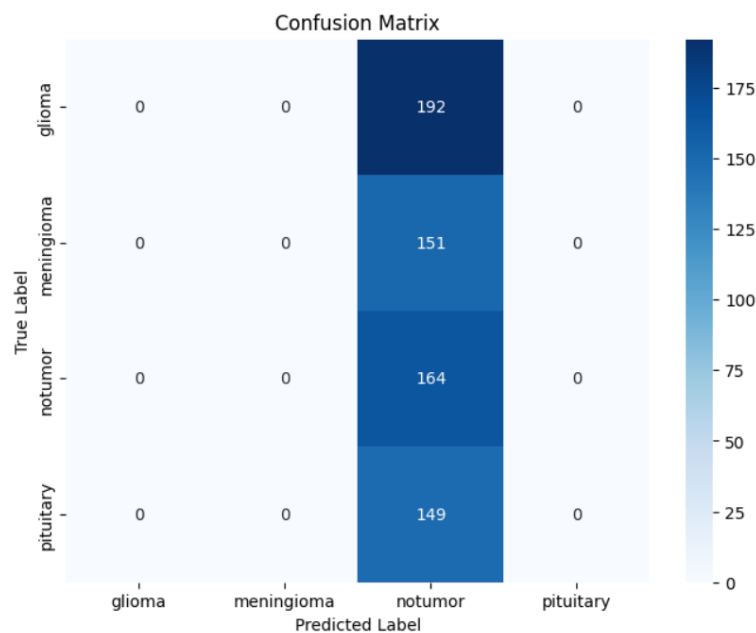
Rapport de classification : Le rapport de classification montre une performance très faible pour trois classes sur quatre. Seule la classe "notumor" est prédite correctement, mais même dans ce cas, la précision reste limitée (0.25), ce qui indique que le modèle

classe la majorité des échantillons à tort dans cette catégorie. Cela traduit une incapacité du modèle à différencier les classes en l'absence de techniques d'optimisation.

	precision	recall	f1-score	support
<i>pituitary</i>	0.00	0.00	0.00	192
<i>glioma</i>	0.00	0.00	0.00	151
<i>notumor</i>	0.25	1.00	0.40	164
<i>meningioma</i>	0.00	0.00	0.00	149

TAB. 5.2 : Rapport de classification CNN-S

Analyse de la matrice de confusion : La matrice de confusion confirme le constat précédent : le modèle classe tous les échantillons dans la catégorie "notumor", indépendamment de leur vraie classe. Cela indique un biais sévère et une généralisation inexistante, probablement dû à un manque de régularisation, de bon initialisation des poids ou de stratégie d'apprentissage efficace.



Résumé des résultats :

- **Exactitude globale (Accuracy) :** 25.00%
- **Problème identifié :** modèle biaisé vers une seule classe
- **Classes ignorées :** *pituitary*, *glioma*, *meningioma*
- **Conclusion :** le modèle CNN sans optimisation n'est pas capable d'apprendre efficacement à partir des données brutes. Une stratégie d'optimisation est indispensable.

5.3 Res-Net18 (Residual Network)

5.3.1 Avec optimisation

Les courbes de perte et de précision du modèle ResNet18 optimisé indiquent une convergence rapide avec une stabilité notable dès les premières époques. La perte diminue régulièrement pour l'entraînement et la validation, sans signes de surapprentissage. La précision de validation reste élevée et proche de celle de l'entraînement tout au long de l'apprentissage, dépassant 98%. Le modèle atteint une accuracy finale de 99.08%, confirmant une excellente généralisation et une performance globale très satisfaisante.

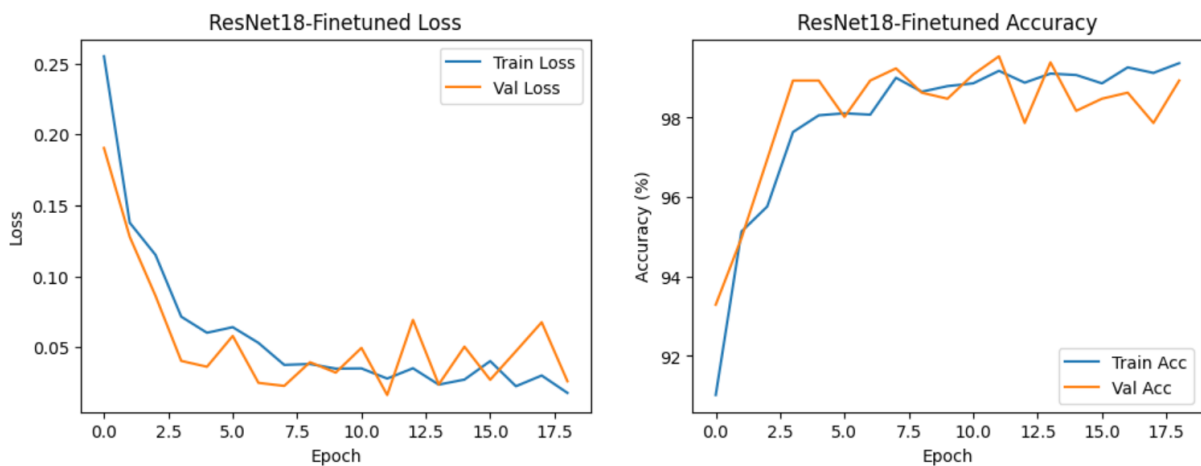


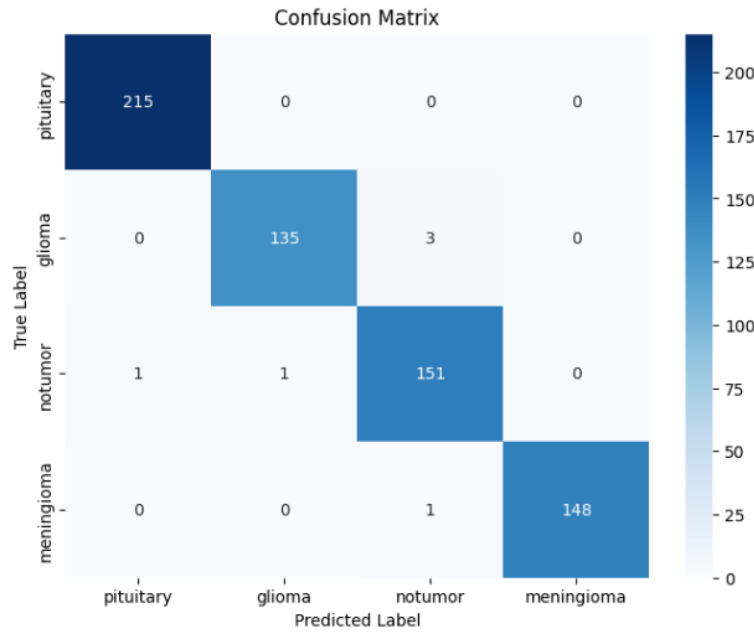
FIG. 5.3 : Courbes de performance du modèle ResNet18 Avec optimisation

Rapport de classification :

	Precision	Recall	F1-Score	Support
Pituitary	1.00	1.00	1.00	215
Glioma	0.99	0.98	0.99	138
Notumor	0.97	0.99	0.98	153
Meningioma	1.00	0.99	1.00	149

TAB. 5.3 : Rapport de classification Res-Net18|A

Le modèle présente de très bons résultats avec des f1-scores proches ou égaux à 1.00 pour toutes les classes. Les classes pituitary et meningioma sont parfaitement classées, tandis que glioma et notumor montrent une légère confusion, mais restent bien reconnues avec des f1-scores de 0.99 et 0.98 respectivement.



Analyse de la matrice de confusion : La matrice montre une très bonne performance globale, avec une prédominance des prédictions correctes sur la diagonale. Les erreurs sont rares, principalement entre glioma et notumor, ce qui reste attendu compte tenu de la similarité entre certaines images IRM.

5.3.2 Sans optimisation

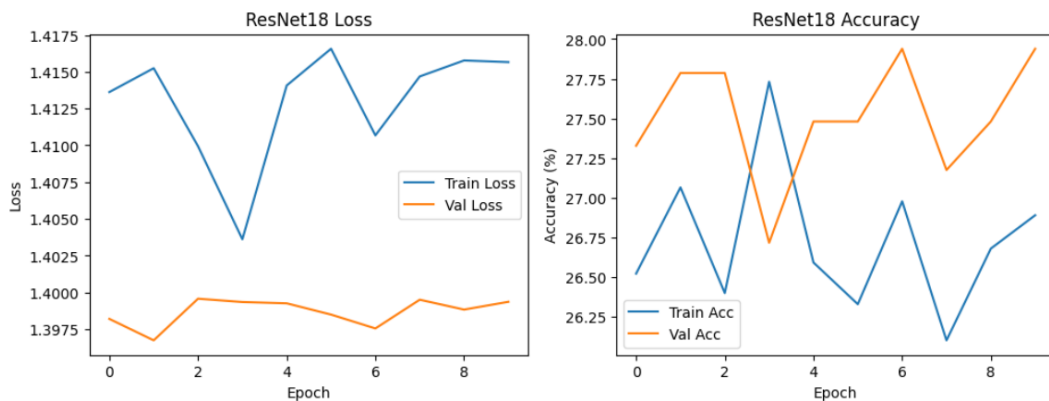


FIG. 5.4 : Courbes de performance du modèle ResNet18 sans optimisation

Analyse des courbes d'apprentissage : Les courbes de perte (loss) montrent une légère baisse, mais restent globalement stables autour de 1.4, ce qui indique une très faible progression de l'apprentissage. La courbe de précision (accuracy) reste également plate, avec des valeurs proches de 26-28%, ce qui est à peine mieux qu'une prédiction aléatoire dans un problème à 4 classes (25%). Cela indique que le modèle n'arrive pas à extraire des caractéristiques discriminantes efficaces, probablement à cause de l'absence d'optimisation.

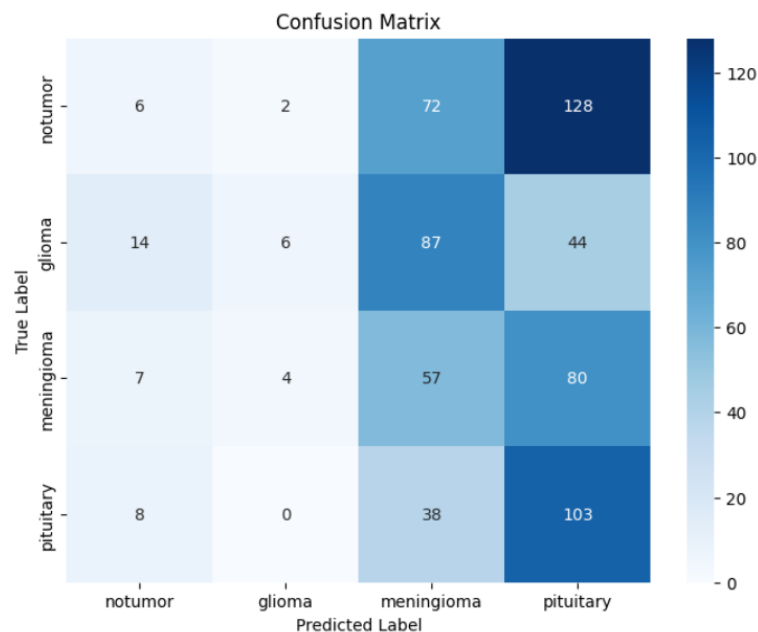
- **Rapport de classification** : Les résultats du rapport de classification montrent des performances très faibles dans la majorité des classes

	precision	recall	f1-score	support
pituitary	0.17	0.03	0.05	208
glioma	0.50	0.04	0.07	151
notumor	0.22	0.39	0.28	148
meningioma	0.29	0.69	0.41	149

TAB. 5.4 : Rapport de classification Res-Net18|S

Seule la classe meningioma atteint un rappel relativement élevé (0.69), mais la précision est faible, ce qui signifie que le modèle a tendance à surclasser dans cette catégorie. Les autres classes sont mal reconnues, avec des F1-scores très bas, notamment pour pituitary et glioma.

Analyse de la matrice de confusion : La matrice de confusion révèle un mélange significatif entre les classes : La majorité des images de notumor sont classées comme pituitary (128/148). Les images de pituitary sont fortement mal classées : 103 d'entre elles sont reconnues comme pituitary (sur 208), mais avec beaucoup d'erreurs. Le modèle confond glioma avec toutes les autres classes. Meningioma est la classe la mieux identifiée, mais contient également beaucoup de confusion (57 images classées come notumor).



Résumé des résultats :

- Exactitude globale (Accuracy) : 26.22%
- Problème principal : sous-apprentissage et absence d'optimisation
- Classe dominante (prédite) : meningioma, mais avec peu de fiabilité
- Classes négligées : glioma, pituitary, notumor

5.4 ViT (vision transformer)

5.4.1 Avec optimisation

Analyse des courbes d'apprentissage

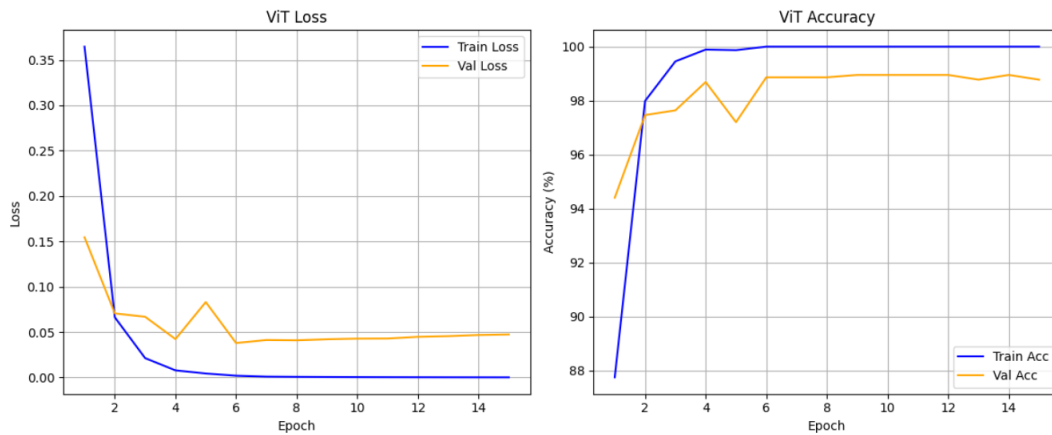


FIG. 5.5 : Courbes de performance du modèle ViT avec optimisation

Les courbes montrent une baisse rapide de la loss dès les premières époques, puis une stabilisation.

L'accuracy dépasse rapidement les 99%, ce qui indique un apprentissage rapide et efficace.

Rapport de classification

Les résultats sont excellents sur toutes les classes

:

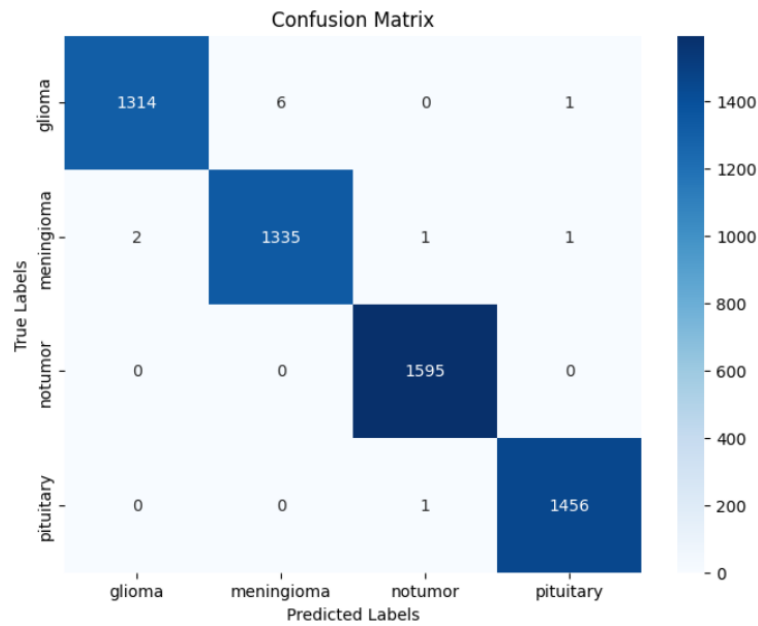
	precision	recall	f1-score	support
pituitary	1.00	0.99	1.00	1321
glioma	1.00	1.00	1.00	1339
notumor	1.00	1.00	1.00	1595
meningioma	1.00	1.00	1.00	1457

TAB. 5.5 : Rapport de classification ViT-A

Ces scores indiquent que le modèle reconnaît correctement presque tous les échantillons, avec très peu d'erreurs.

Analyse de la matrice de confusion

La matrice confirme la performance : Très peu de confusions entre les classes. Quelques erreurs mineures ex : 6 glioma prédits en pituitary), mais aucune classe n'est négligée.



Résumé : Accuracy globale : 99.79%

Résultat : Le modèle ViT, avec optimisation, offre une généralisation remarquable.

Conclusion : C'est un modèle très performant, bien adapté à la tâche de classification des IRM.

5.4.2 Sans optimisation

Analyse des courbes d'apprentissage

Les courbes sont plates : la loss ne diminue pas et l'accuracy reste constante autour de 25.30%, ce qui indique que le modèle n'apprend pratiquement rien.

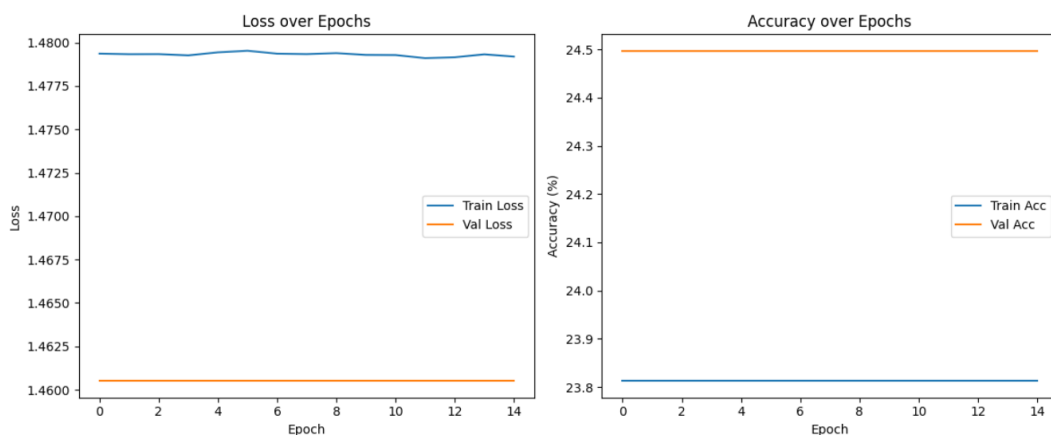


FIG. 5.6 : Courbes de performance du modèle ViT sans optimisation

Rapport de classification

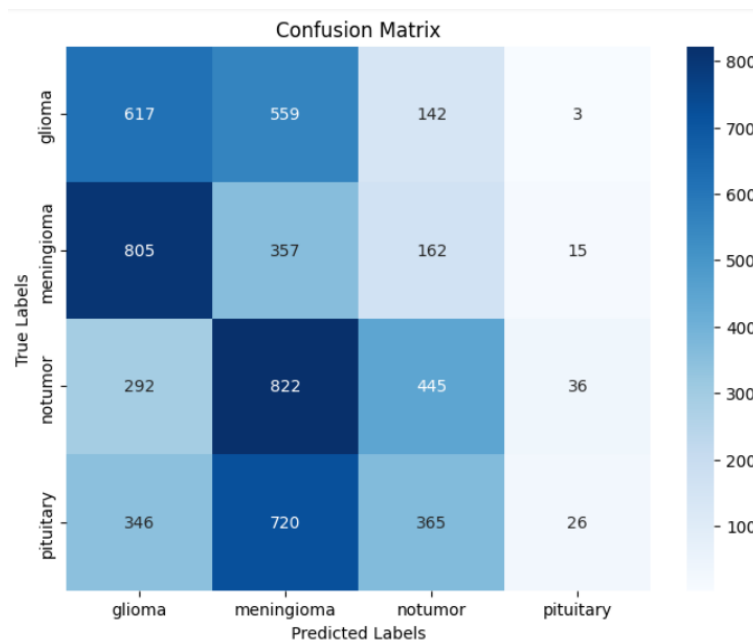
	precision	recall	f1-score	support
pituitary	0.35	0.49	0.40	304
glioma	0.13	0.26	0.18	252
notumor	0.44	0.32	0.37	299
meningioma	0.09	0.00	0.01	288

TAB. 5.6 : Rapport de classification ViT-S

Seule la classe pituitary montre un rappel acceptable. La classe meningioma est complètement ignorée par le modèle.

Matrice de confusion

On observe de grandes confusions entre toutes les classes. Le modèle a tendance à prédire les classes glioma et meningioma même lorsqu'elles sont incorrectes.



Résumé

- **Accuracy globale** : 25.30%
- **Problème principal** : Le modèle est non entraîné et ne parvient pas à apprendre les bonnes représentations.
- **Conclusion** : Sans optimisation, le ViT est inefficace, malgré son potentiel.

5.5 Res-net18 avec DRB (deep rule-based)

5.5.1 Avec optimization

Le graphique montre que le classificateur DRB reconnaît uniquement les tumeurs gliome avec une précision de 100%, tandis qu'il échoue complètement à identifier les autres types (méningiome, hypophyse, absence de tumeur).

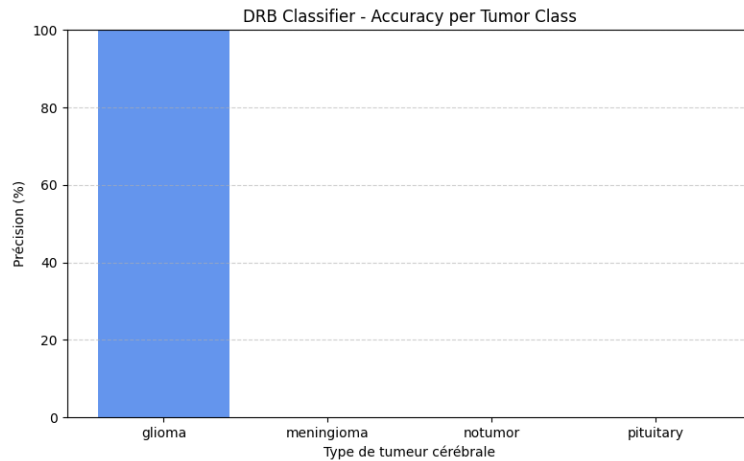


FIG. 5.7 : Performance du modèle DRB avec optimisation

Cela indique que le modèle est biaisé vers une seule classe, ce qui le rend inefficace pour une classification multi-classes. Cette limitation peut être due à un déséquilibre des données ou à une faible capacité de généralisation du classificateur DRB.

Rapport de classification

Le tableau ci-dessous montre les métriques principales pour chaque classe : précision, rappel, F1-score et support.

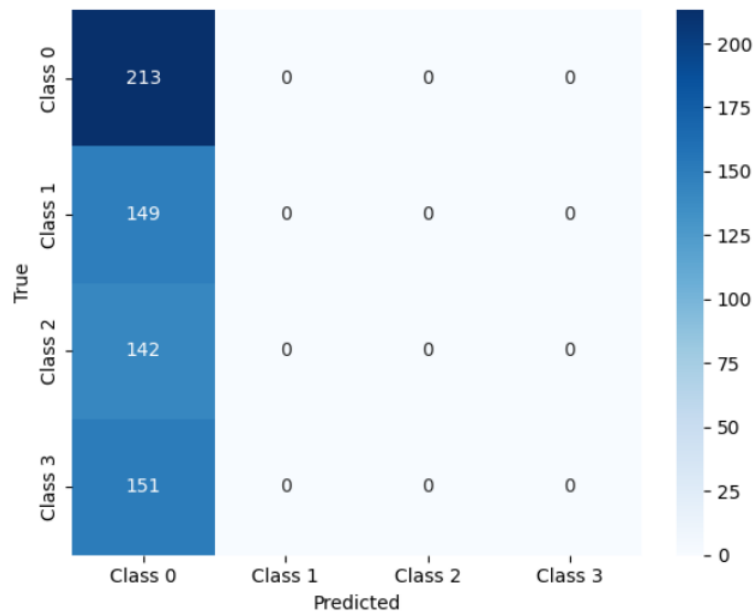
	precision	recall	f1-score	support
pituitary	0.33	1.00	0.49	213
glioma	0.00	0.00	0.00	149
notumor	0.00	0.00	0.00	142
meningioma	0.00	0.00	0.00	151

TAB. 5.7 : Rapport de classification ResNet-DRB|A

Le modèle montre une très faible performance globale. Seule la classe pituitary est partiellement reconnue (rappel élevé mais précision faible), tandis que les autres classes ne sont pas du tout correctement prédites.

l'analyse de la matrice de confusion

Le modèle prédit presque tout comme 'pituitary', avec aucune prédiction correcte pour les autres classes. Il est donc incapable de différencier les types de tumeurs.



Le modèle atteint une accuracy de 32.52%, mais cela est dû à une prédiction dominante de la classe 'pituitary'. Les autres classes sont mal classées, ce qui reflète un surapprentissage sévère et une mauvaise généralisation. Le modèle échoue à distinguer les tumeurs.

5.5.2 Sans optimisation

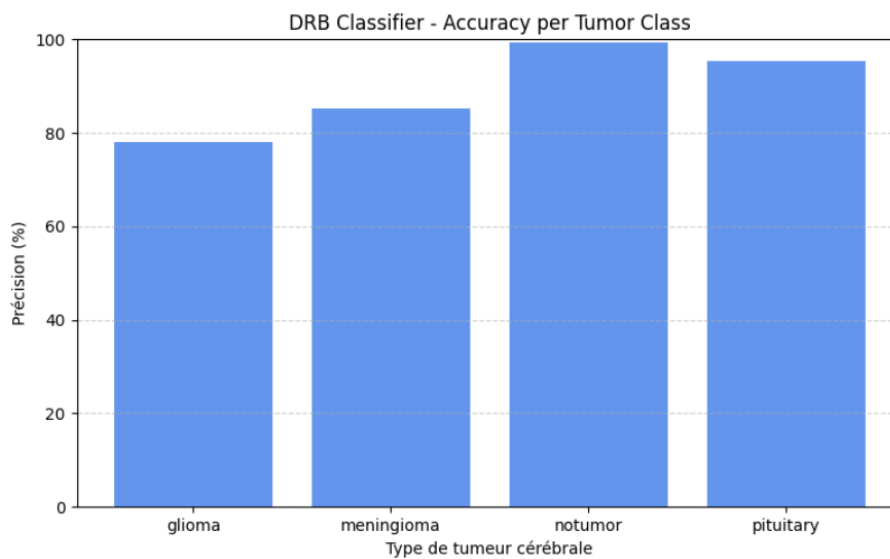


FIG. 5.8 : Performance du modèle DRB sans optimisation

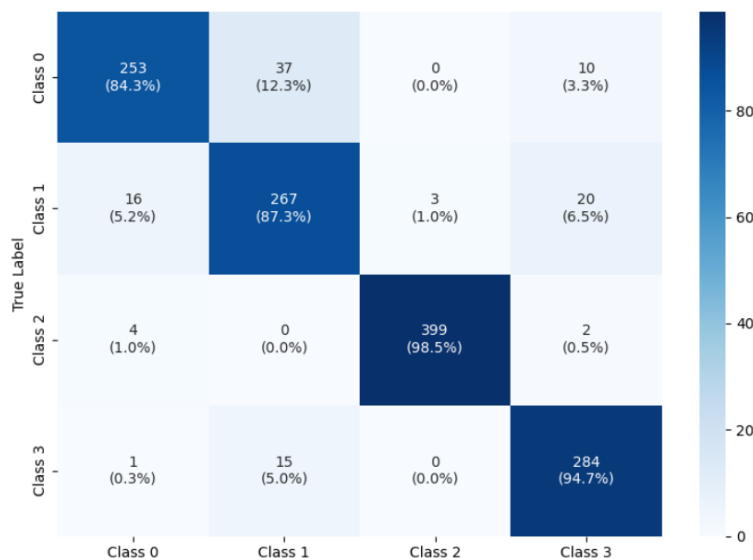
Le graphique montre que le modèle ResNet18 avec DRB sans optimisation identifie très bien la classe "notumor" (100%) et "pituitary" (95%). La précision diminue pour

”meningioma” (86%) et surtout ”glioma” (78%), ce qui reflète une difficulté à distinguer certaines tumeurs sans phase d’optimisation.

	precision	recall	f1-score	support
pituitary	0.93	0.81	0.86	300
glioma	0.83	0.89	0.85	306
notumor	0.99	0.99	0.99	405
meningioma	0.91	0.96	0.93	300

TAB. 5.8 : Rapport de classification ResNet-DRB|S

Malgré l’absence d’étape d’optimisation, le modèle ResNet18 combiné avec le classificateur symbolique DRB affiche de bonnes performances dans la classification des images IRM. Il a atteint une précision de 100% pour la classe notumor, et des résultats remarquables pour la classe pituitary. Même pour les classes meningioma et glioma, les taux de précision restent respectables, ce qui démontre la capacité de ce système à apprendre et à extraire des règles de classification de manière efficace sans réglage approfondi. Cela met en évidence la robustesse initiale de l’approche symbolique à base de règles dans le traitement des images médicales.



Le graphique montre que le modèle ResNet18 avec DRB, malgré l’absence d’optimisation, offre de bonnes performances, notamment pour la classe saine (Class 2) avec une précision de 98,5% et la classe 3 avec 94,7%. Cependant, certaines confusions apparaissent entre les classes 0 et 1, indiquant des difficultés à distinguer certains types de tumeurs.

5.6 Analyse Comparative de la Performance des Modèles avec et sans Optimisation

Afin d’évaluer de manière synthétique les performances des différents modèles de classification des IRM cérébrales, nous présentons ci-dessous un tableau comparatif ras-

semblant les principales métriques : accuracy, précision et F1-score, pour chaque modèle testé, avec et sans optimisation.

Ce tableau permet de visualiser rapidement l'impact des techniques d'optimisation sur la qualité des prédictions, ainsi que les différences de performance entre les architectures (CNN, ResNet18, DRB, ViT).

Cette analyse comparative est essentielle pour identifier les modèles les plus efficaces et les mieux adaptés à la tâche de classification multicatégorie dans un contexte médical sensible.

Modèle	Accuracy	Recall	Precision	Specificity	F1-score
CNN	98.02%	0.98	0.98	0.99	0.98
ViT	99.79%	1.00	1.00	0.99	1.00
ResNet-18	99.08%	0.99	0.99	0.99	0.99
ResNet-18 avec DRB	32.52%	0.33	0.11	0.57	0.16

Tableau : Résultats des modèles **avec optimisation**

Modèle	Accuracy	Recall	Precision	Specificity	F1-score
CNN	25.00%	0.25	0.06	0.74	0.10
ViT	25.30%	0.26	0.30	0.78	0.23
ResNet-18	29.73%	0.30	0.35	0.88	0.20
ResNet-18 avec DRB	91.76%	0.92	0.92	0.97	0.92

Tableau : Résultats des modèles **sans optimisation**

Le tableau présenté permet une comparaison approfondie des performances de plusieurs modèles de classification des IRM cérébrales, en considérant deux contextes distincts : avec et sans optimisation. Il met en évidence l'impact considérable de l'optimisation – réalisée ici à l'aide de l'optimiseur ADAM – sur l'efficacité des modèles.

Tout d'abord, le Vision Transformer (ViT) optimisé se démarque nettement avec une accuracy de 99.79% et des scores parfaits (1.00) pour toutes les métriques (rappel, précision, spécificité et F1-score). Ce résultat illustre la puissance de ce modèle à condition qu'il soit correctement ajusté, car sans optimisation, ses performances chutent brutalement à 25.30%, ce qui reflète une incapacité à apprendre efficacement dans son état brut.

Le ResNet-18 optimisé montre également d'excellentes performances avec une accuracy de 99.08% et des métriques équilibrées (0.99). Ce modèle surpasse clairement le CNN (Convolutional Neural Network) optimisé, qui atteint une accuracy de 98.02%. Bien que ce dernier soit plus simple en termes d'architecture, il parvient tout de même à offrir de très bons résultats après optimisation.

En revanche, sans optimisation, les modèles CNN, ViT et ResNet-18 affichent des résultats très faibles, avec des accuracies respectives de 25.00%, 25.30% et 29.73%. Cela souligne l'importance capitale de l'optimisation, sans laquelle ces modèles sont pratiquement inefficaces.

Un cas particulier concerne l'association ResNet-18 avec le classificateur DRB. Sans optimisation, ce modèle a obtenu une accuracy de 91.76%, ce qui est étonnamment élevé par rapport à sa version optimisée (32.52%). Cela suggère que l'intégration du DRB n'a pas bien interagi avec la stratégie d'optimisation appliquée. Il est possible que DRB introduise des biais ou perturbe la dynamique d'apprentissage lorsqu'il est utilisé avec un modèle déjà finement ajusté.

En résumé, ce tableau met clairement en évidence plusieurs points clés :

- L'optimisation est cruciale pour tirer pleinement parti des architectures avancées, notamment ViT et ResNet.
- Le modèle ViT se révèle être le plus performant, à condition d'être optimisé.
- Le ResNet-18 constitue un excellent compromis entre performance et complexité.
- L'association avec un classificateur externe comme DRB peut être bénéfique uniquement dans des cas spécifiques, et nécessite une intégration prudente.

Ces résultats démontrent la nécessité d'un choix judicieux des architectures, combiné à une stratégie d'optimisation adaptée, pour atteindre une classification fiable et robuste des IRM cérébrales.

5.7 Conclusion

Ce chapitre a permis de mettre en évidence l'influence de l'optimisation sur les modèles de classification de l'IRM cérébrale. Les résultats montrent clairement que les techniques d'optimisation, en particulier l'optimiseur ADAM, sont nécessaires pour utiliser pleinement le potentiel des architectures avancées comme ViT et ResNet-18. Après optimisation, le modèle ViT s'est avéré le plus efficace, tandis que ResNet-18 offre un bon équilibre entre simplicité et efficacité. Enfin, l'analyse a également mis en évidence la complexité des relations entre certains composants, tels que le DRB et les stratégies d'optimisation, ce qui nécessite une intégration plus réfléchie. Ces résultats fournissent donc une base pour guider les décisions futures sur la conception et l'optimisation des modèles dans un cadre médical raisonnable.

Conclusion générale et perspectives

L'Intelligence Artificielle (IA) est une technique appliquée dans divers domaines, entre autre le domaine médical, notamment dans l'aide au diagnostic. L'IA peut s'exercer dans un environnement qui contient beaucoup de données et sur des plateformes matérielles performantes. En effet, nous vivons dans une aire où les hôpitaux se numérisent, des quantités de données médicales sont générées quotidiennement. En plus que ces données sont en quantité énorme, elles souffrent aussi de plusieurs problèmes, ce qui rend difficile leur analyse et leur interprétation

Dans ce travail, nous avons abordé la problématique de la classification automatique des images IRM cérébrales, qui représente un enjeu majeur dans le domaine de l'intelligence artificielle appliquée à la santé. Notre objectif principal était de développer et d'évaluer différents modèles d'apprentissage profond capables d'identifier de manière précise les types de tumeurs cérébrales à partir d'images médicales.

Nous avons exploré plusieurs architectures, allant du CNN classique à des modèles plus avancés tels que ResNet-18 et Vision Transformer (ViT), avec et sans optimisation. Une attention particulière a été portée sur l'impact de l'optimisation notamment à l'aide de l'optimiseur ADAM ainsi que sur l'intégration éventuelle de classificateurs symboliques comme DRB.

Les résultats obtenus ont mis en lumière plusieurs constats clés :

- Le modèle ViT optimisé s'est distingué comme le plus performant, atteignant une précision globale de 99.79%, démontrant ainsi le potentiel remarquable des architectures basées sur les transformateurs dans le domaine de la vision médicale.
- Le ResNet-18 optimisé a également montré une excellente robustesse avec une précision de 99.08%, surpassant le CNN tout en gardant une architecture plus légère que le ViT.

L'étude comparative a souligné l'importance cruciale de l'optimisation pour améliorer significativement les performances des modèles, en particulier pour les architectures complexes. L'association entre ResNet-18 et le classificateur DRB a donné des résultats contrastés, suggérant que les approches hybrides nécessitent un ajustement fin pour être réellement efficaces.

Ainsi, notre étude confirme que le choix de l'architecture, l'optimisation des paramètres, et l'évaluation rigoureuse des résultats sont des éléments fondamentaux pour le développement de systèmes intelligents fiables dans le diagnostic assisté par ordinateur.

Ce travail ouvre également la voie à de futures perspectives telles que l'intégration de données cliniques supplémentaires, l'extension à d'autres pathologies cérébrales, ou encore le déploiement des modèles dans un environnement clinique réel pour valider leur utilité pratique.

Comme perspectives envisagées à la suite de cette étude comparative, nous proposons :

- L'exploitation de bases de données plus étendues et diversifiées, permettant une meilleure généralisation des modèles grâce à un apprentissage sur un volume de données significatif.
- L'utilisation de ressources matérielles plus puissantes, notamment des clusters de GPU, pour accélérer le processus d'entraînement et améliorer l'efficacité computationnelle.
- L'extension du système actuel par l'ajout de modules spécialisés dans la détection d'autres types de pathologies médicales, afin d'élargir le champ d'application du modèle au-delà des tumeurs cérébrales.
- La mise en œuvre du système proposé dans un contexte réel, via un déploiement sous forme de service web interopérable avec les systèmes d'information hospitaliers, afin d'évaluer sa faisabilité clinique et son impact opérationnel.

Bibliography

- ABIRAMI, S., & CHITRA, P. (2018). Energy-efficient edge node placement algorithm for smart and sustainable environment in IoT. *Procedia Computer Science*, 143, 387-394. <https://doi.org/10.1016/j.procs.2018.04.250>
- AHMAD, H. A., YU, H. J., & MILLER, C. G. (2013). Medical imaging modalities. In *Medical imaging in clinical trials* (p. 3-26). Springer.
- AHMED, M. (2020). Adam Optimization. <https://aiinarabic.com/adam-optimization/>
- ALZUBAIDI, L., ZHANG, J., HUMAIDI, A. J., AL-DUJAILI, A., DUAN, Y., AL-SHAMMA, O., SANTAMARÍA, J., FADHEL, M. A., AL-AMIDIE, M., & FARHAN, L. (2021). Review of deep learning : concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8, 1-74.
- AMARAPPA, S., & SATHYANARAYANA, S. (2014). Data classification using Support vector Machine (SVM), a simplified approach. *Int. J. Electron. Comput. Sci. Eng*, 3, 435-445.
- ANGELOV, P., & YAGER, R. (2012). A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, 41(2), 163-185.
- ANGELOV, P. P., & GU, X. (2018). Deep rule-based classifier with human-level performance and characteristics. *Information Sciences*, 463, 196-213.
- ANGELOV, P. P., GU, X., & PRÍNCIPE, J. C. (2017). A generalized methodology for data analysis. *IEEE transactions on cybernetics*, 48(10), 2981-2993.
- BEYSOLOW II, T. (2017). *Introduction to deep learning using R : A step-by-step guide to learning and implementing deep learning models using R*. Apress.
- BUERKI, R. A., HORBINSKI, C. M., KRUSER, T., HOROWITZ, P. M., JAMES, C. D., & LUKAS, R. V. (2018). An overview of meningiomas. *Future oncology*, 14(21), 2161-2177.
- CANONAR, A. T. (2023). *Derin Öğrenme Çalışmaları : Geçmişten Günümüze* [Consulté le 30 mai 2025]. <https://medium.com/@alitunacanonar/derin-%C3%B6%C4%9Frenme-%C3%A7al%C4%B1%C5%9Fmalar%C4%B1-ge%C3%A7mi%C5%9Ffa8d2188dcfb>
- CHACHOUA, R., & BENSABAHA, I. (2021). *Radio signal classification using deep learning* [thèse de doct., université Ghardaia].
- CHEN, C.-M., CHEN, C.-C., & CHEN, C.-C. (2006). A comparison of texture features based on SVM and SOM. *18th International Conference on Pattern Recognition (ICPR'06)*, 2, 630-633.

- CIRESAN, D. C., MEIER, U., GAMBARDILLA, L. M., & SCHMIDHUBER, J. (2011). Convolutional neural network committees for handwritten character classification. *2011 International conference on document analysis and recognition*, 1135-1139.
- CROISÉ-LAURENT, V. (2008). *Imagerie de Diffusion par RMN : optimisation des Paramètres et Applications Cliniques pour les Explorations Hépatiques* [thèse de doct., Université Henri Poincaré-Nancy 1].
- DEBENEDETTI, E., & TRONCOSO—EPFL, C. (2022). Adversarially robust vision transformers. *Master's thesis, Swiss Federal Institute of Technology, Lausanne (EPFL)*.
- DOMINGOS, P., & PAZZANI, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 29, 103-130.
- DOSOVITSKIY, A., BEYER, L., KOLESNIKOV, A., WEISSENBORN, D., ZHAI, X., UNTERTHINER, T., DEGHANI, M., MINDERER, M., HEIGOLD, G., GELLY, S., et al. (2020). An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*.
- EL-HACHEMI, M. G. (2018). Performances dans la Segmentation d'images médicales. *École nationale Supérieure en Informatique*.
- GEEKSFORGEEKS. (2024). *Python Features* [Consulté le 5 mai 2023]. <https://www.geeksforgeeks.org/python-features/>
- GÉRON, A. (2019). Hands-on machine learning with scikit-learn, keras, and tensorflow : concepts. *Aurélien Géron-Google Kitaplar, yy https://books.google.com.tr/books*.
- GU, J., WANG, Z., KUEN, J., MA, L., SHAHROUDY, A., SHUAI, B., LIU, T., WANG, X., WANG, G., CAI, J., et al. (2018). Recent advances in convolutional neural networks. *Pattern recognition*, 77, 354-377.
- GUPTA, D. (2017). Fundamentals of deep learning—introduction to recurrent neural networks. *Analytics Vidhya (2107)*.
- GYLBERTH, R. (2018). An Introduction to AdaGrad. *Medium*. <https://medium.com/konvergen/an-introduction-to-adagrad-f130ae871827>
- HAWARAH, L., SIMONET, A., & SIMONET, M. (2006). Evaluation d'une approche probabiliste pour le classement d'objets incomplètement connus dans un arbre de décision. *Avant-propos : Troisième atelier " Fouille de Données Complexes dans un processus d'extraction des connaissances*, 83.
- HEATON, J. (2018). Ian goodfellow, yoshua bengio, and aaron courville : Deep learning : The mit press, 2016, 800 pp, isbn : 0262035618. *Genetic programming and evolvable machines*, 19(1), 305-307.
- HOPFIELD, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
- HORRI TAQYEDDINE, M. M. (2021). Detection of Diabetic Retinopathy in fundus images using deep learning.
- HOSPODARSKYY, O., MARTSENYUK, V., KUKHARSKA, N., HOSPODARSKYY, A., & SVERSTIUK, S. (2024). Understanding the Adam Optimization Algorithm in Machine Learning.
- IBM. (2023). *What are Convolutional Neural Networks ?* [Consulté le 5 juin 2023]. <https://www.ibm.com/topics/convolutional-neural-networks>
- KIMURA, N., YOSHINAGA, I., SEKIJIMA, K., AZECHI, I., & BABA, D. (2019). Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. *Water*, 12(1), 96.

- KOZÁK, S. (2024). MRI Radiomics for IDH Genotype Prediction in Glioblastoma Diagnosis. *arXiv preprint arXiv :2409.16329*.
- LECUN, Y., BENGIO, Y., & HINTON, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- LODI, A., & ZARPELLON, G. (2017). On learning and branching : a survey. *Top*, 25(2), 207-236.
- MERCK MANUALS. (2023). *Brain - Biology of the Nervous System* [Consulté le 30 mai 2025]. <https://www.merckmanuals.com/home/brain-spinal-cord-and-nerve-disorders/biology-of-the-nervous-system/brain>
- MIAO, Y., LI, J., WANG, Y., ZHANG, S.-X., & GONG, Y. (2016). Simplifying long short-term memory acoustic models for fast training and decoding. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2284-2288.
- NICKPARVAR, M. (2025). *Brain Tumor MRI Dataset* [Consulté le 5 juin 2023]. <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>
- OBERHEIM BUSH, N. A., & CHANG, S. (2016). Treatment strategies for low-grade glioma in adults. *Journal of oncology practice*, 12(12), 1235-1241.
- PILLOW CONTRIBUTORS. (2025). *Pillow (PIL Fork) Documentation* [Consulté le 9 mai 2023]. <https://pillow.readthedocs.io/en/stable/>
- PRABHU, R. (2018). Understanding of Convolutional Neural Network (CNN) — Deep Learning [Accessed : 2025-05-31]. *Medium*. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>
- RAMZAN, F., KHAN, M. U., REHMAT, A., IQBAL, S., SABA, T., REHMAN, A., & MEHMOOD, Z. (2019). A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. *Journal of Medical Systems*, 44. <https://doi.org/10.1007/s10916-019-1475-2>
- REESE, H. (2017). Understanding the differences between AI, machine learning, and deep learning. *URL : https://www.techrepublic.com/article/understandingthedifferencesbetweenaimachinelearninganddeeplearning*, 11(02), 2023.
- RISH, I., et al. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 3(22), 41-46.
- ROBERT, J. (2022). *Pandas : la bibliothèque Python dédiée à la Data Science* [Consulté le 5 juin 2023]. <https://datascientest.com/pandas-python-data-science>
- ROUSE, M. (2019a). *What is Matplotlib ?* [Consulté le 5 juin 2023]. <https://www.techopedia.com/definition/33861/matplotlib>
- ROUSE, M. (2019b). *What is Scikit-Learn ?* [Consulté le 5 juin 2023]. <https://www.techopedia.com/definition/33860/scikit-learn>
- SCHMIDHUBER, J. (2012). Multi-column deep neural networks for image classification. *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3642-3649.
- SWANBERG, K. M., KURADA, A. V., PRINSEN, H., & JUCHEM, C. (2022). Multiple sclerosis diagnosis and phenotype identification by multivariate classification of in vivo frontal cortex metabolite profiles. *Scientific Reports*, 12(1), 13888.
- TAQYEDDINE, H., MOHAMED, M., & OMAR, B. (2020). Detection of Diabetic Retinopathy in fundus images using deep learning.

Bibliography

- THAPAR, K., KOVACS, K., SCHEITHAUER, B., & LLOYD, R. V. (2000). *Diagnosis and management of pituitary tumors*. Springer Science & Business Media.
- VAPNIK, V. (2013). *The nature of statistical learning theory*. Springer science & business media.
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., & POLOSUKHIN, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- W3SCHOOLS. (2025). *Introduction to NumPy* [Consulté le 5 juin 2023]. https://www.w3schools.com/python/numpy/numpy_intro.asp
- XU, Y., WEI, H., LIN, M., DENG, Y., SHENG, K., ZHANG, M., TANG, F., DONG, W., HUANG, F., & XU, C. (2022). Transformers in computational visual media : A survey. *Computational Visual Media*, 8, 33-62.
- ZEMAN, B. (2023). *What is Google Colab?* [Consulté le 5 juin 2023]. <https://www.androidpolice.com/google-colab-explainer/>