RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Yahia Fares de Medea
Faculté des Sciences
Département Mathématiques et Informatique



Mémoire

En vue de l'obtention du diplome de Master
en Informatique
Option : Ingénierie des systèmes et technologies du web

Classification des tumeurs cérébrales

Mémoire rédirigé par :
Terkmane Youcef

Encadrant :
Dr.Mazari Redha

Année universitaire : **2023 – 2024**

Remerciement et Dédicace

La réalisation de ce travail a´été possible grâce à dieu le tout puissant, grâce à plusieurs personnes à qui nous voudrions témoigner toute notre reconnaissance.

Nous tenons à remercier, dans un premier temps, toute l'équipe pédagogique de l'université pour avoir assuré la partie théorique de celle-ci.

Nous remercions profondément notre encadreur monsieur **Dr. Mazari Redha** pour ses suivis et ses précieuses orientations dans notre travail et Nous voudraient vous remercier pour tous vos conseils et vos remarques intéressantes.

Nous remercions également l'enseignant examinateur d'avoir accepté de juger notre travail.

Pour finir , nous adressons toute notre gratitude 'a nos chers parents, fr'eres et sœurs pour leurs patiences et leurs soutiens.

Résumé

Ce projet présente le développement d'un modèle de classification des tumeurs cérébrales utilisant des réseaux de neurones convolutionnels (CNN). L'objectif principal était de créer un modèle capable de distinguer efficacement entre différents types de tumeurs cérébrales à partir d'images IRM. Nous avons commencé par préparer un ensemble de données complet en appliquant des techniques de redimensionnement, de normalisation. Ensuite, un modèle CNN performant a été conçu, comprenant plusieurs couches de convolution, de pooling et des couches entièrement connectées pour la classification finale. Le modèle a été entraîné et validé en utilisant des métriques telles que l'accuracy, la précision, le rappel et le score F1, obtenant des résultats prometteurs avec une accuracy de 97% sur l'ensemble de test. L'analyse de la matrice de confusion a révélé une bonne capacité de classification du modèle, bien que certaines erreurs de différenciation entre classes spécifiques subsistent. En comparaison avec des travaux antérieurs, notre modèle a démontré des performances supérieures, soulignant l'efficacité de notre approche.

Les perspectives futures incluent l'exploration d'architectures CNN plus complexes, l'enrichissement de l'ensemble de données, la validation clinique et le développement d'interfaces utilisateur intuitives. Ces travaux contribuent à l'amélioration du diagnostic des tumeurs cérébrales et ouvrent la voie à de futures recherches dans ce domaine crucial.

Abstract

This project presents the development of a brain tumor classification model using convolutional neural networks (CNN). The main objective was to create a model capable of effectively distinguishing between different types of brain tumors from MRI images. We began by preparing a comprehensive dataset, applying techniques such as resizing and normalization. Next, a high-performance CNN model was designed, comprising several convolutional layers, pooling layers, and fully connected layers for final classification. The model was trained and validated using metrics such as accuracy, precision, recall, and F1 score, achieving promising results with an accuracy of 97% on the test set. The confusion matrix analysis revealed a good classification capability of the model, although some differentiation errors between specific classes remain. Compared to previous works, our model demonstrated superior performance, highlighting the effectiveness of our approach.

Future perspectives include exploring more complex CNN architectures, enriching the dataset, clinical validation, and developing intuitive user interfaces. This work contributes to improving the diagnosis of brain tumors and paves the way for future research in this critical field.

ملخص

يهدف هذا العمل الى تطوير نموذج لتصنيف أورام الدماغ باستخدام الشبكات العصبية الالتفافية .(CNN) كان الهدف الرئيسي هو إنشاء نموذج قادر على التمييز بفعالية بين أنواع مختلفة من أورام الدماغ من صور الرنين المغناطيسي. بدأنا بإعداد مجموعة بيانات شاملة بتطبيق تقنيات مثل إعادة التشكيل والتطبيع. بعد ذلك، تم تصميم نموذج CNN عالي الأداء، يتضمن عدة طبقات التفاف، طبقات تجميع وطبقات متصلة بالكامل للتصنيف النهائي. تم تدريب النموذج والتحقق من صحته باستخدام مقاييس مثل الدقة، الاستدعاء، الدقة والدرجة ،F1 محققًا نتائج واعدة بدقة 97% على مجموعة الاختبار. كشف تحليل مصفوفة الارتباك عن قدرة جيدة على تصنيف النموذج، رغم بقاء بعض الأخطاء في التمييز بين فئات محددة. بالمقارنة مع الأعمال السابقة، أظهر نموذجنا أداءً متفوقًا، مما يبرز فعالية نهجنا.

تشمل الآفاق المستقبلية استكشاف هياكل CNN أكثر تعقيدًا، إثراء مجموعة البيانات، التحقق السريري وتطوير واجهات مستخدم بديهية. تساهم هذه الأعمال في تحسين تشخيص أورام الدماغ وتفتح الطريق لأبحاث مستقبلية في هذا المجال الحيوي.

Table des matières

Remerciement et Dédicace

Resumé

Abstract

1 Introduction générale **1**

2 État de l'art **3**
2.1 Introduction . . . 3
2.2 Les soins de santé . . . 3
2.2.1 Définitions . . . 3
2.2.2 Étapes de soins . . . 4
2.3 Intelligence artificielle et santé . . . 4
2.4 Apprentissage automatique . . . 6
2.4.1 Définition . . . 6
2.4.2 Types d'apprentissage automatique . . . 6
2.5 L'apprentissage en profondeur . . . 10
2.5.1 Définition . . . 10
2.5.2 Réseaux de neurones artificiels . . . 11
2.6 Réseaux de neurones convolutifs . . . 16
2.6.1 Définition de CNN . . . 17
2.6.2 Blocs CNN . . . 17
2.6.3 Types de couches CNN selon les tâches . . . 18
2.6.4 Détection d'objets . . . 21
2.7 Tumeurs cérébrales . . . 22
2.7.1 L'importance du sujet : . . . 23
2.7.2 Classes des tumeurs cérébrales . . . 23
2.8 Travaux connexes . . . 26
2.9 Conclusion . . . 27

3 Conception et mise en œuvre : **28**
3.1 Introduction . . . 28
3.2 Conception du système . . . 28
3.2.1 Jeu de données . . . 29
3.2.2 Prétraitement . . . 30
3.2.3 Division de l'ensemble de données . . . 31

3.2.4 Apprentissage CNN 32
3.2.5 Structure du Modèle 32
3.2.6 Processus d'Apprentissage 33
3.2.7 Prédiction 33
3.2.8 Évaluation du Modèle CNN 33
3.3 Mise en œuvre d'une architecture d'apprentissage profond 35
3.3.1 Cadres, outils et bibliothèques 35
3.3.2 Préparation et prétraitement des données 38
3.3.3 Construction de notre modèle CNN 39
3.4 Test de notre modèle CNN 42
3.5 Conclusion 42

4 Expérimentation et résultats **43**
4.1 Introduction 43
4.2 Ensemble de données 43
4.3 Initialisation des paramètres 44
4.4 Expérimentation 44
4.5 Tester le modèle 45
4.6 Comparaison de notre travail avec les travaux précédents 47
4.7 Conclusion 47

5 Conclusion et perspectives **49**

Table des figures

2.1 AI vs ML vs DL . . . 6
2.2 Apprentissage supervisé. . . . 7
2.3 Classification et Regression. . . . 7
2.4 Apprentissage non supervisé. . . . 8
2.5 Apprentissage semi-supervisé. . . . 8
2.6 Apprentissage par renforcement. . . . 9
2.7 Apprentissage par transfert. . . . 10
2.8 Différence entre apprentissage automatique et DL. . . . 11
2.9 Neurone biologique . . . 12
2.10 Neurone artificiel . . . 12
2.11 Réseaux de neurones artificiels . . . 13
2.12 Courbe de fonction ReLU . . . 14
2.13 Courbe de fonction Leaky ReLU . . . 15
2.14 Courbe de fonction Sigmoid . . . 15
2.15 Softmax function curve . . . 16
2.16 Cortex visuel humain . . . 17
2.17 Composition de CNN . . . 18
2.18 Analyser l'image à l'aide du filtre . . . 19
2.19 Stride en CNN. . . . 19
2.20 Padding en CNN. . . . 20
2.21 Pooling layer . . . 20
2.22 Fully connected layer . . . 21
2.23 CNN learning process . . . 21
2.24 Détection d'objets . . . 22
2.25 Gliome . . . 24
2.26 Méningiome . . . 24
2.27 Tumeur hypophysaire . . . 25
2.28 Absence de tumeur . . . 25

3.1 Schéma de notre système . . . 29
3.2 Structure de l'ensemble de donnée . . . 30
3.3 Python . . . 35
3.4 OpenCV . . . 36
3.5 VSCode . . . 36
3.6 Colab . . . 36
3.7 Matplotlib . . . 37
3.8 Keras . . . 37
3.9 TensorFlow . . . 37
3.10 NumPy . . . 38

3.11 Kaggle 38
3.12 Résumé du modèle 41

4.1 L'accuracy du modèle 44
4.2 La perte du modèle 45
4.3 Test du modèle 45
4.4 Matrice de confusion 46

Chapitre 1

Introduction générale

De nos jours, avec le développement des ordinateurs et leur capacité à effectuer des calculs complexes, de nombreux algorithmes et techniques d'apprentissage ont été mis en œuvre. Ces techniques ont pour but d'automatiser des tâches et d'aider les humains dans leurs processus de prise de décision. Divers secteurs tels que les achats en ligne, la recherche sur le web, la cybersécurité et les soins de santé ont bénéficié de l'application de ces techniques. En particulier, les soins de santé ont connu des progrès significatifs grâce à l'automatisation des tâches effectuées par les machines [1]. De nombreux systèmes ont été développés pour aider les cliniciens dans leur prise de décision en matière de diagnostic des maladies [2]. Ces systèmes utilisent principalement l'apprentissage automatique et l'apprentissage profond pour extraire des informations cliniques à partir d'images.
Les tumeurs cérébrales représentent un problème de santé complexe et difficile qui représente des menaces importantes pour les individus à travers le monde, avec environ 24 810 adultes aux États-Unis ayant reçu un diagnostic de tumeurs cancéreuses primaires du cerveau et du SNC en 2023 [3].
Ces croissances anormales dans le cerveau peuvent se manifester de diverses manières. formes, allant de bénignes à malignes, et nécessitent souvent un diagnostic et un traitement précis. Les progrès de l'imagerie médicale et des technologies informatiques ont ouvert la voie à une meilleure compréhension et classification des tumeurs cérébrales. La capacité d'identifier et de catégoriser avec précision ces tumeurs est cruciale pour développer des stratégies de traitement efficaces et améliorer les résultats pour les patients.

Problématique

Bien que les techniques d'imagerie médicale aient considérablement progressé, la classification précise des tumeurs cérébrales reste un défi de taille. Les approches diagnostiques traditionnelles reposent souvent sur une interprétation subjective, ce qui peut entraîner des erreurs de diagnostic et des retards dans le traitement. De plus, la quantité croissante de données médicales, notamment dans le domaine de la neuroimagerie, représente un obstacle majeur pour l'analyse manuelle. Pour surmonter ces défis, il est nécessaire d'intégrer des technologies de pointe telles que l'intelligence artificielle et l'apprentissage automatique afin d'améliorer la précision et l'efficacité de la classification des tumeurs cérébrales.

Objectif de la thèse

L'objectif principal de cette thèse est de créer un modèle d'apprentissage automatique robuste capable de classer automatiquement les tumeurs cérébrales. La recherche vise à améliorer la précision des diagnostics, évaluer la praticabilité dans les contextes cliniques et apporter des contributions précieuses aux progrès en cours dans ce domaine.

Organisation de la thèse

Après cette introduction générale, la thèse est structurée comme suit :

— **Chapitre 2: État de l'art.** fournit une revue approfondie de la littérature, mettant en évidence les techniques et méthodologies de pointe en matière de classification des tumeurs cérébrales.
— **Chapitre 3: Conception et mise en œuvre.** détaille les matériaux et les méthodes utilisés dans cette recherche, y compris la collecte de données, le prétraitement et le développement du modèle d'apprentissage automatique.
— **Chapitre 4: Expérimentation et résultats.** présente les résultats expérimentaux, les évaluations des performances et les comparaisons avec les approches existantes.
— **Conclusion et perspectives :** résume les principales contributions de la thèse, tire des conclusions et propose des recommandations pour l'avancement de la recherche sur la classification des tumeurs cérébrales.

Chapitre 2

État de l'art

2.1 Introduction

Ces dernières années, l'intégration de l'intelligence artificielle (IA) a révolutionné plusieurs aspects des soins de santé, ouvrant la voie à une nouvelle ère de médecine personnalisée. Dans le domaine de la neurologie en particulier, l'IA joue un rôle central dans l'avancement de notre compréhension et de notre gestion de maladies complexes telles que les tumeurs cérébrales. La combinaison des technologies d'IA avec les pratiques médicales offre la promesse d'améliorer considérablement la précision du diagnostic, la planification du traitement et les résultats pour les patients.
Le potentiel transformateur de l'IA dans le domaine de la santé réside dans sa capacité à analyser rapidement et avec précision de grandes quantités de données médicales. Dans le contexte des tumeurs cérébrales, cela implique l'utilisation de techniques d'imagerie avancées comme l'imagerie par résonance magnétique (IRM) et la tomodensitométrie (TDM) pour extraire des détails complexes qui pourraient échapper à la perception humaine. En appliquant des algorithmes d'apprentissage automatique, il devient possible d'identifier des modèles et des anomalies subtiles, contribuant ainsi à la détection précoce et à la classification des tumeurs cérébrales.
Dans ce chapitre, nous définirons les soins de santé, nous donnerons un aperçu de l'apprentissage automatique et de l'apprentissage profond, nous décrirons les tumeurs cérébrales et les différents types de tumeurs cérébrales, et enfin nous mentionnerons quelques travaux antérieurs connexes.

2.2 Les soins de santé

2.2.1 Définitions

Les soins de santé, également connus sous le nom de santé, englobent différentes définitions selon les sources :

Selon un dictionnaire médical, les soins de santé englobent "*la prévention, le traitement et la gestion des maladies, ainsi que la préservation du bien-être mental et physique grâce aux services fournis par les professionnels de la santé et des professions connexes*" [4].

Dans le dictionnaire Cambridge, les soins de santé sont définis comme "*l'ensemble des services fournis par un pays ou une organisation pour le traitement des personnes physiquement et mentalement malades*" [5].

Market Business News (MBN) définit les soins de santé comme "*les efforts déployés par les professionnels de la santé pour restaurer notre bien-être physique et mental, et cela inclut également la fourniture de services pour maintenir le bien-être émotionnel*" [6].

2.2.2 Étapes de soins

Les soins de santé sont divisés en trois étapes en fonction du problème de santé traité.

1. **Soins de santé primaires :**

   Les soins de santé primaires désignent les services de santé essentiels accessibles à tous dans la communauté, qu'il s'agisse d'individus ou de familles. Ils visent à répondre aux besoins de santé d'une personne tout au long de sa vie, en englobant le bien-être physique, mental et social. Plutôt que de se concentrer sur les maladies, les soins de santé primaires adoptent une approche centrée sur les personnes. Ils incluent divers aspects tels que la promotion de la santé, la prévention des maladies, le traitement, la réadaptation et les soins palliatifs [7].
2. **Soins de santé secondaires :**

   Lorsqu'un prestataire de soins primaires réfère un patient à un spécialiste, celui-ci passe aux soins de santé secondaires. Les soins de santé secondaires impliquent de recevoir une attention médicale spécialisée de la part de professionnels possédant une expertise spécifique dans le traitement de la pathologie particulière du patient. Les spécialistes se concentrent généralement sur un système spécifique du corps ou sur une maladie ou affection spécifique. Par exemple, les cardiologues se concentrent sur le cœur et son système de pompage, tandis que les endocrinologues se focalisent sur les systèmes hormonaux et peuvent se spécialiser dans des maladies telles que le diabète ou les troubles thyroïdiens. Les soins de santé secondaires sont nécessaires lorsque les individus ont besoin de soins médicaux spécialisés [8].
3. **Soins de santé tertiaires :**

   Les soins de santé tertiaires offrent des soins spécialisés aux patients référés par les soins de santé secondaires pour un diagnostic ou un traitement. Ils englobent des services qui ne sont pas disponibles dans les établissements de soins primaires ou secondaires. Les soins de santé tertiaires sont souvent associés à des affections rares ou à des situations où les installations de diagnostic ou de traitement sont limitées ou nécessitent des ressources rares ou des combinaisons de ressources rares. Ces installations sont couramment présentes dans les écoles de médecine et les hôpitaux universitaires. Elles peuvent également être impliquées dans des activités de recherche liées à des domaines médicaux spécifiques [9].

2.3 Intelligence artificielle et santé

L'intelligence artificielle (IA) se réfère à la création de systèmes informatiques capables de reproduire des fonctions intellectuelles humaines telles que la prise de décision, la détec-

tion d'objets et la résolution de problèmes complexes. L'IA présente plusieurs avantages significatifs, notamment :

- Capacité à prédire avec une grande précision.
- Assistance dans les processus de prise de décision.
- Résolution de problèmes complexes.
- Capacité à effectuer des calculs de haut niveau qui nécessiteraient des jours pour être résolus par un être humain.

L'IA a eu un impact significatif dans divers domaines tels que le marketing, la finance, les jeux et même les arts musicaux. Cependant, son impact le plus important s'est produit dans le domaine de la santé, et ce pour deux raisons principales.

1. **Disponibilité des données médicales :** Un grand nombre de données médicales, notamment les antécédents médicaux et les images médicales, sont maintenant facilement accessibles. L'IA repose sur des technologies telles que l'apprentissage profond et l'apprentissage automatique, qui nécessitent une quantité considérable de données. Grâce à la disponibilité de ces données, l'implémentation de l'IA dans le secteur de la santé est devenue beaucoup plus facile.
2. **Développement d'algorithmes complexes :** Auparavant, l'apprentissage automatique avait du mal à traiter des ensembles de données de grande dimension. Les données médicales, notamment les données de santé, sont caractérisées par leur grande dimension, comprenant des milliers d'attributs. Il était difficile d'analyser ces données à l'aide de l'apprentissage automatique traditionnel. Cependant, l'introduction de l'apprentissage profond a permis le développement d'algorithmes complexes qui ont joué un rôle majeur dans les soins de santé.

L'intelligence artificielle (IA) a le potentiel d'impact sur divers domaines dans le domaine de la santé, notamment les applications administratives, les soins aux patients et les processus médicaux. Voici quelques exemples de la manière dont l'IA pourrait avoir un impact sur différents domaines de la santé :

1. **Applications administratives :**
   - L'IA peut être utilisée pour le traitement des demandes, la documentation clinique, la gestion du cycle de revenus et la gestion des dossiers médicaux, ce qui permet d'obtenir des gains d'efficacité considérables dans le domaine de la santé[10].
   - Elle peut automatiser des tâches répétitives telles que la saisie de données, le traitement des demandes et la planification des rendez-vous, améliorant ainsi les performances opérationnelles et simplifiant les procédures complexes[11].
2. **Soins aux patients :**
   - L'IA peut soutenir le personnel de santé dans des tâches telles que l'automatisation des dispositifs médicaux, la surveillance des patients et l'aide à la décision clinique, ce qui se traduit finalement par de meilleurs résultats pour les patients et une efficacité accrue[12].
   - Elle peut révolutionner la médecine personnalisée, optimiser les dosages médicamenteux et améliorer la gestion de la santé de la population, offrant ainsi un meilleur accès aux traitements personnalisés et à des soins de qualité[13].

3. **Processus médicaux :**
   - L'IA peut être utilisée pour la détection précoce et le diagnostic des maladies, car les algorithmes surpassent déjà les radiologues dans la détection de tumeurs malignes et aident les chercheurs à constituer des cohortes pour les essais cliniques[10].
   - Elle est utilisée dans l'imagerie médicale, l'analyse des codes de diagnostic, les prédictions de flambées de maladies infectieuses et la modélisation des maladies, contribuant ainsi à améliorer le diagnostic et le traitement des maladies[14].

2.4 Apprentissage automatique

2.4.1 Définition

L'apprentissage automatique (ML) est un domaine relevant de l'intelligence artificielle (IA) et de l'informatique qui utilise des données et des algorithmes pour reproduire le processus d'apprentissage des êtres humains, améliorant continuellement sa précision [15]. Il s'agit d'une subdivision de l'IA qui permet aux machines d'acquérir indépendamment des connaissances à partir de données et d'expériences passées, d'identifier des motifs et de faire des prédictions avec une intervention humaine minimale. L'objectif principal de l'apprentissage automatique est de permettre aux ordinateurs d'apprendre et de manifester un comportement sans avoir besoin d'une programmation explicite.

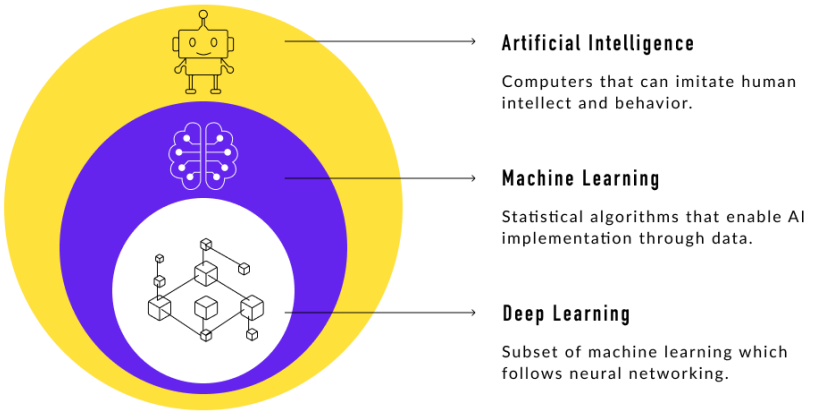


FIGURE 2.1 – AI vs ML vs DL

The relation between artificial intelligence (AI), machine learning (ML), and deep learning (DL) is shown in Figure 2.1.

2.4.2 Types d'apprentissage automatique

Selon la nature des données et le résultat souhaité de l'algorithme, il existe principalement cinq types d'apprentissage automatique :

1. **Supervised Learning ou Apprentissage supervisé :**
   L'apprentissage supervisé est un type d'apprentissage automatique où l'algorithme apprend à associer des données d'entrée à des données de sortie connues. En d'autres termes, l'algorithme est entraîné sur un ensemble de données étiquetées, où la sortie correcte est déjà connue, et il apprend à prédire la sortie correcte pour de nouvelles données d'entrée non vues auparavant. L'objectif de l'apprentissage supervisé est de minimiser la différence entre la sortie prédite et la sortie réelle.

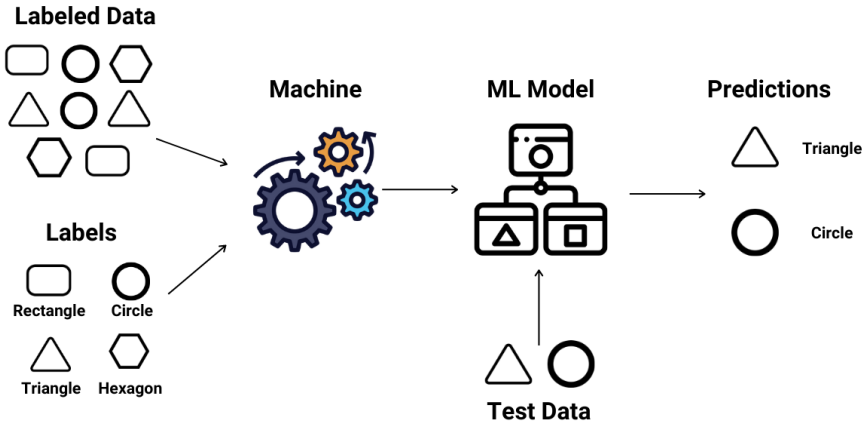


FIGURE 2.2 – Apprentissage supervisé.

L'apprentissage supervisé permet d'accomplir deux types de tâches : la classification et la régression. La figure 2.3 illustre la distinction entre la tâche de classification et la tâche de régression.

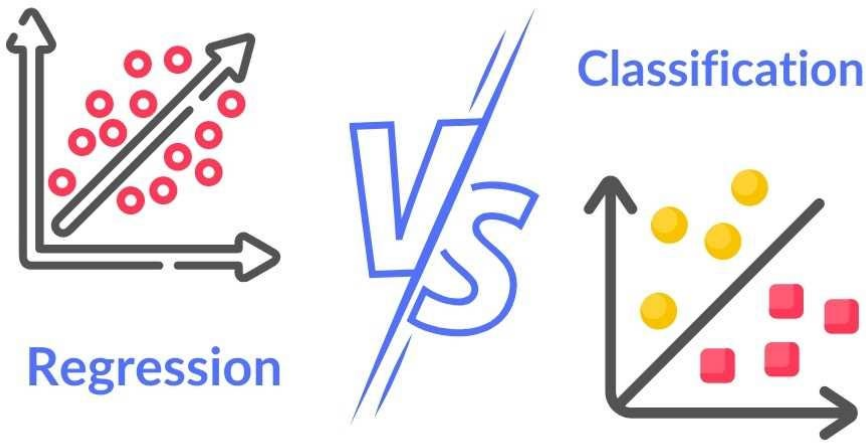


FIGURE 2.3 – Classification et Regression.

La régression consiste à prédire une valeur en tant que résultat. Tous les types de sorties sont possibles, tels qu'une chaîne de caractères, une valeur numérique, une date, etc. En revanche, la classification vise à diviser les données d'entrée en au moins deux catégories. Il n'est pas nécessaire de faire des prédictions, car toutes les données d'entrée sont classées en fonction de leur catégorie respective.

2. **Unsupervised Learning ou Apprentissage non supervisé :**
   L'apprentissage non supervisé est un type d'apprentissage automatique où l'algorithme apprend à trouver des modèles dans des données non étiquetées. En d'autres termes, l'algorithme ne reçoit aucune donnée étiquetée et doit trouver lui-même les modèles et la structure dans les données. L'objectif de l'apprentissage non supervisé est de découvrir des modèles cachés et des relations dans les données.

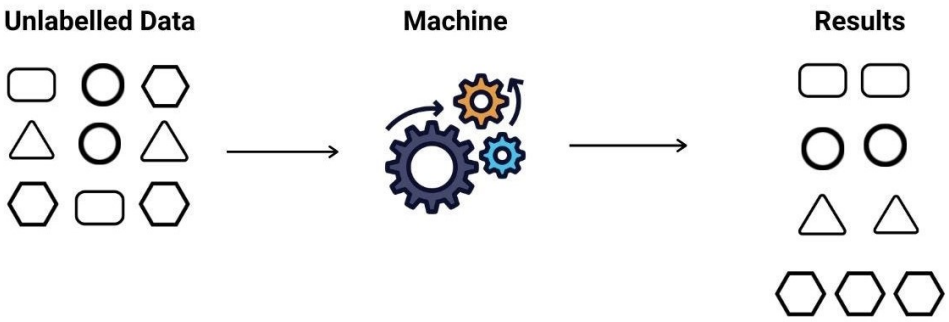


FIGURE 2.4 – Apprentissage non supervisé.

3. **Semi-Supervised Learning ou Apprentissage semi-supervisé :**
   L'apprentissage semi-supervisé est un type d'apprentissage automatique qui combine à la fois l'apprentissage supervisé et non supervisé. Dans l'apprentissage semi-supervisé, l'algorithme est entraîné sur un ensemble de données qui contient à la fois des données étiquetées et non étiquetées. L'objectif de l'apprentissage semi-supervisé est d'utiliser les données étiquetées pour guider le processus d'apprentissage et améliorer la précision du modèle sur les données non étiquetées.

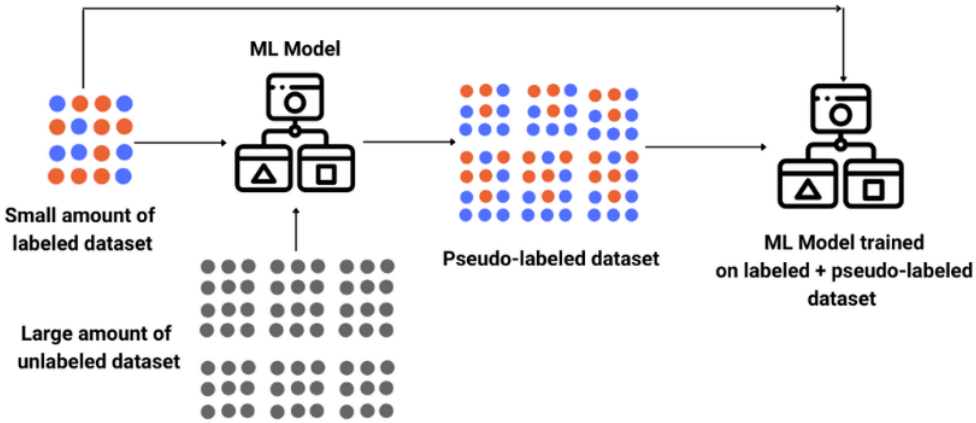


FIGURE 2.5 – Apprentissage semi-supervisé.

4. **Reinforcement Learning ou Apprentissage par renforcement :**

L'apprentissage par renforcement est une forme d'apprentissage automatique où l'algorithme apprend à prendre des décisions en fonction des retours de l'environnement. En d'autres termes, l'algorithme ne reçoit pas de données étiquetées, mais apprend plutôt par essais et erreurs. L'objectif de l'apprentissage par renforcement est de maximiser un signal de récompense en prenant des actions qui conduisent à la récompense la plus élevée possible. Des exemples courants d'apprentissage par renforcement incluent les jeux et la robotique.

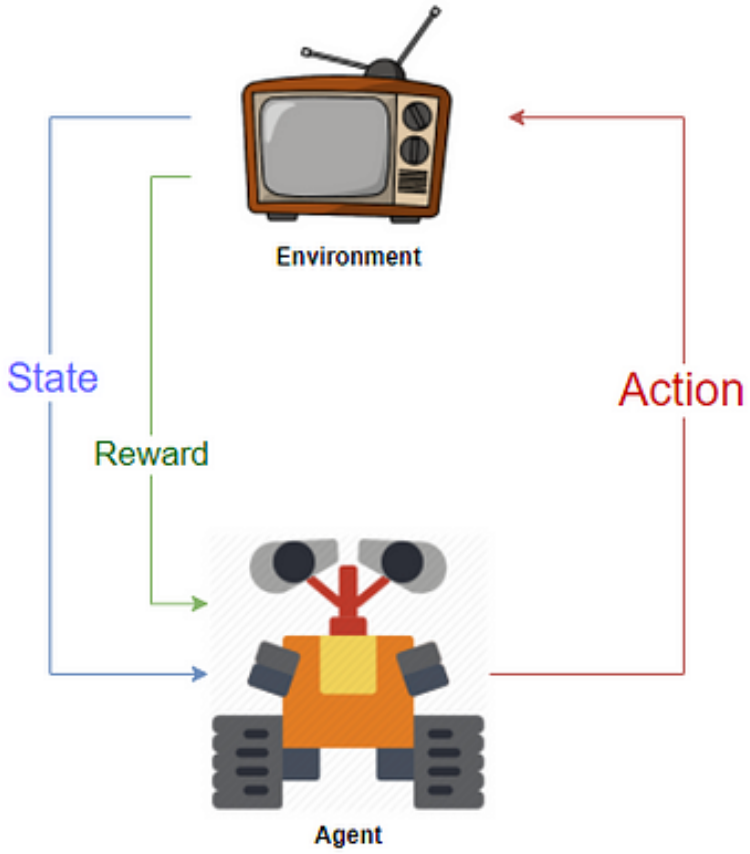


FIGURE 2.6 – Apprentissage par renforcement.

5. **Transfer Learning ou Apprentissage par transfert :**
L'apprentissage par transfert est un type d'apprentissage automatique où un modèle formé sur une tâche est réutilisé comme point de départ pour un modèle sur une tâche différente. En d'autres termes, les connaissances acquises en résolvant un problème sont transférées à un problème différent mais connexe. L'objectif du transfert d'apprentissage est de réduire la quantité de données annotées nécessaires pour former un nouveau modèle et améliorer les performances du modèle sur la nouvelle tâche. Des exemples courants de transfert d'apprentissage comprennent la classification d'images et le traitement du langage naturel.

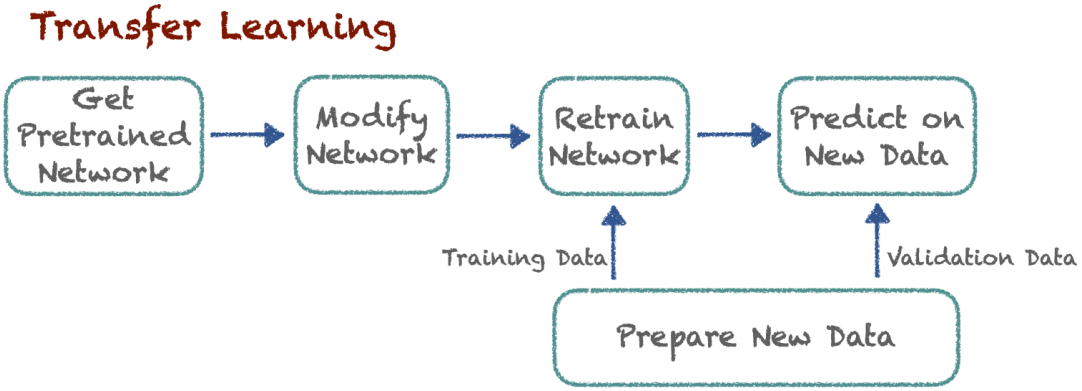


FIGURE 2.7 – Apprentissage par transfert.

2.5 L'apprentissage en profondeur

2.5.1 Définition

L'apprentissage en profondeur (ou deep learning en anglais) est une branche de l'intelligence artificielle qui utilise des réseaux de neurones artificiels pour apprendre à partir de données. Les réseaux de neurones sont des modèles mathématiques qui simulent le fonctionnement du cerveau humain en utilisant des couches de neurones interconnectés pour traiter des informations. L'apprentissage en profondeur permet de résoudre des problèmes complexes tels que la reconnaissance d'images, la traduction automatique, la reconnaissance vocale, etc. en utilisant des algorithmes d'apprentissage automatique pour entraîner les réseaux de neurones à partir de grandes quantités de données.

La principale différence entre les algorithmes de deep learning et d'apprentissage automatique (voir figure 2.8) réside dans le fait que les techniques de deep learning utilisent des réseaux neuronaux à couches multiples pour apprendre et extraire automatiquement des caractéristiques à partir de données brutes, contrairement aux algorithmes d'apprentissage automatique qui reposent sur des caractéristiques créées manuellement et sur l'ingénierie des caractéristiques.

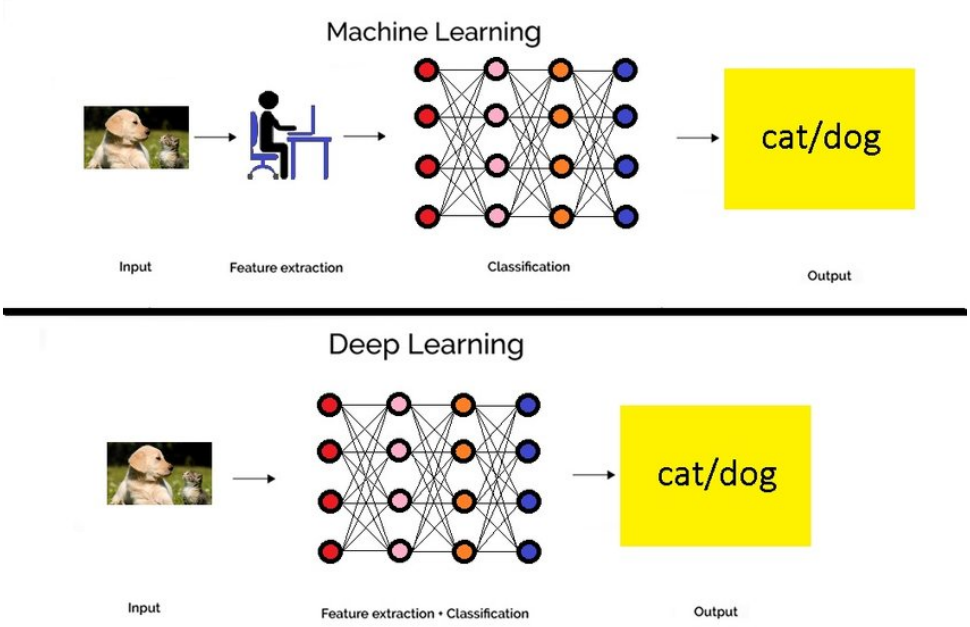


FIGURE 2.8 – Différence entre apprentissage automatique et DL.

2.5.2 Réseaux de neurones artificiels

Avant de comprendre les réseaux neuronaux artificiels, il est essentiel de saisir certains concepts fondamentaux liés aux réseaux neuronaux biologiques et aux modifications apportées dans le modèle de réseau neuronal artificiel.

— **- Neurone biologique :**
Le système nerveux repose sur une cellule essentielle appelée neurone biologique, comme le montre la Figure 2.9. Sa fonction principale est de collecter, analyser et transmettre des informations à l'aide de signaux électriques et chimiques.
Un neurone biologique est composé de quatre éléments structurels clés : les dendrites, responsables de la réception des signaux d'entrée, la zone du cône d'implantation, qui intègre ces signaux et génère le potentiel de membrane, l'axone, responsable de la transmission du potentiel d'action, et la synapse, qui convertit le potentiel d'action en énergie électrique ou chimique et le transmet au neurone suivant.

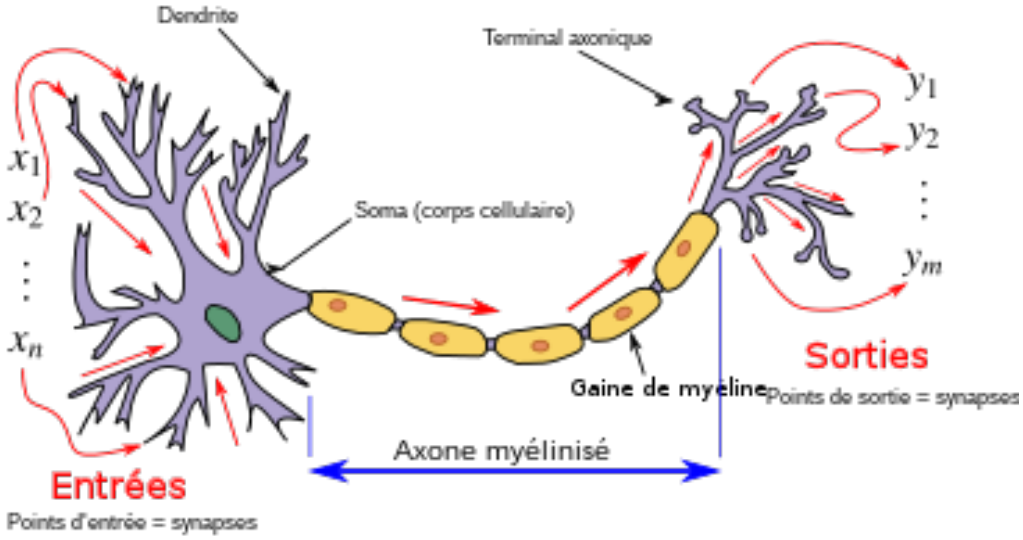


FIGURE 2.9 – Neurone biologique

— **- Neurone artificiel :**
Un modèle computationnel qui imite la structure et le fonctionnement d'un neurone biologique est appelé un neurone artificiel, comme le montre la Figure 2.10. Il constitue une partie essentielle des réseaux neuronaux artificiels, qui sont utilisés dans diverses tâches telles que l'apprentissage automatique, la vision par ordinateur et le traitement du langage naturel. Un neurone artificiel est composé d'entrées X $= \{x1, x2, x3, ...xn\}$ appelées couche d'entrée, de poids W $= \{w1, w2, w3, ...wn\}$, d'un biais (b) et d'une fonction d'activation.
Une fonction d'activation $_f$ convertit la somme pondérée des variables d'entrée, de leurs poids et du biais (b) $\sum_{i=1}^{n}(wi \times xi) + biais$ en une valeur qui est ensuite transférée à la couche de sortie pour être comparée à une valeur seuil et fournir une réponse en sortie.

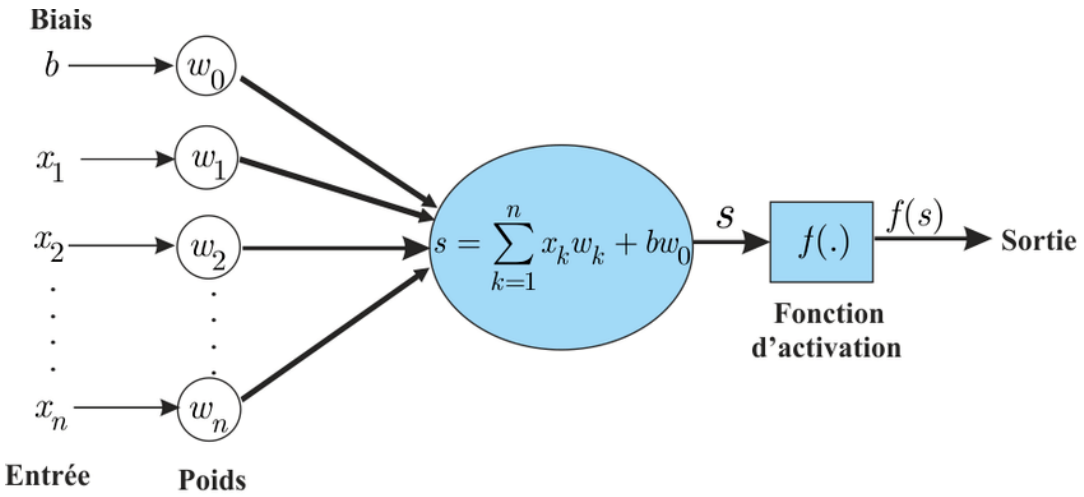


FIGURE 2.10 – Neurone artificiel

— **- Réseaux de neurones artificiels :**
Le réseau neuronal artificiel multicouche est composé de trois couches : la couche

d'entrée, une ou plusieurs couches cachées et la couche de sortie. Cette architecture permet de résoudre des problèmes plus complexes et représente une avancée par rapport aux réseaux précédents. Si l'objectif est de prédire la valeur des revenus, un neurone de sortie est nécessaire. En cas de classification, chaque classe sera représentée par un neurone dans la couche de sortie (voir figure 2.11).

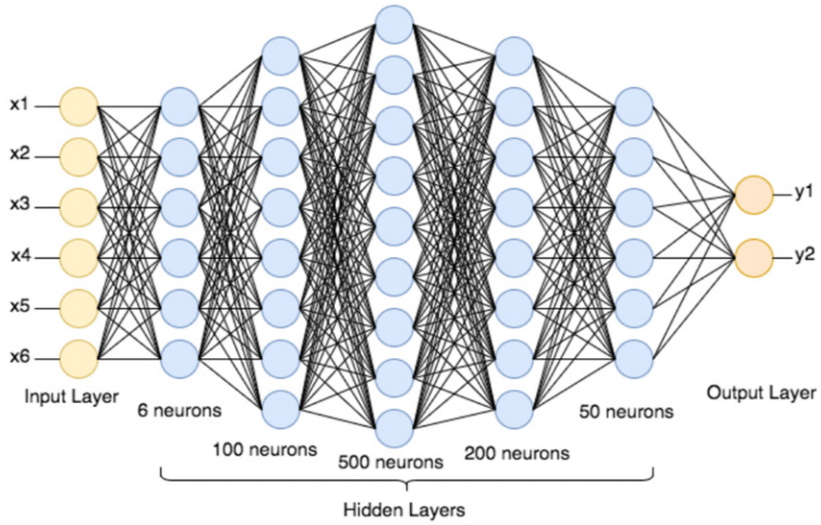


FIGURE 2.11 – Réseaux de neurones artificiels

— **- Types de couches :**
Le réseau de neurones artificiels contient 3 grands types de couches, expliquées ci-dessous :

1. **Couches d'Entrée :** Les couches d'entrée sont la première couche du réseau et servent à recevoir les données brutes en entrée. Chaque neurone dans cette couche correspond à une caractéristique ou une variable d'entrée. Ces neurones transmettent ensuite ces données aux couches suivantes pour le traitement.
2. **Couches Cachées :** Les couches cachées sont situées entre les couches d'entrée et de sortie. Elles effectuent des calculs complexes en utilisant des poids et des fonctions d'activation pour extraire des caractéristiques significatives des données en entrée. Plusieurs couches cachées peuvent être présentes dans un réseau de neurones profond, d'où le terme "profondeur" dans l'apprentissage en profondeur.
3. **Couches de Sortie :** Les couches de sortie produisent les résultats finaux du réseau après avoir traité les données à travers les couches précédentes. Le nombre de neurones dans cette couche dépend du type de tâche que le réseau doit accomplir, par exemple, un neurone pour chaque classe dans une tâche de classification.

— **- Fonction d'activation :**
La fonction d'activation est une fonction mathématique utilisée pour traiter la sortie d'un neurone dans un réseau neuronal, en transformant les valeurs obtenues dans une plage spécifique[16]. Elle joue un rôle essentiel dans l'introduction de la non-linéarité dans un réseau neuronal, ce qui lui permet d'apprendre et de représenter des relations complexes entre les entrées et les sorties. Chaque neurone d'un réseau

neuronal reçoit les signaux d'entrée provenant d'un ou plusieurs neurones et génère un signal de sortie qui est transmis à un ou plusieurs neurones de la couche suivante.

Il existe différents types de fonctions d'activation, chacune ayant un objectif spécifique. Nous allons maintenant mettre en évidence les fonctions d'activation les plus couramment utilisées :

— **Fonction ReLU**

La Rectified Linear Unit (ReLU) est une fonction d'activation fréquemment utilisée dans les réseaux neuronaux artificiels. Si la valeur d'entrée est positive, elle renverra la valeur d'entrée, mais si elle est négative, elle renverra zéro. ReLU est efficace en termes de calcul et est avantageuse pour résoudre le problème du gradient qui disparaît, ce qui la rend appropriée pour les réseaux neuronaux profonds[17].

La courbe de la fonction ReLU est représentée dans la figure 2.12.

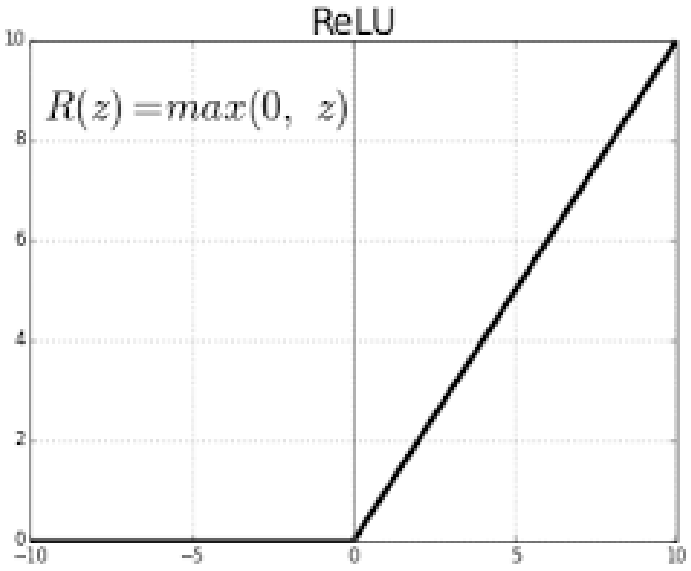


FIGURE 2.12 – Courbe de fonction ReLU

— **Fonction Leaky ReLU**

Définie par l'équation suivante :

$$LR(x) = \max(\alpha x, x)$$

Leaky ReLU est une version modifiée de la fonction d'activation Rectified Linear Unit (ReLU), créée pour résoudre le problème du "ReLU mort". Ce problème survient lorsque la sortie d'un neurone reste à zéro, entraînant le non-mise à jour de ses poids pendant l'entraînement. La fonction Leaky ReLU est définie comme $f(x) = \max(\alpha x, x)$, où α représente une constante positive minime (généralement environ 0,01)[18].

La courbe de la fonction Leaky ReLU est représentée dans la figure 2.13.

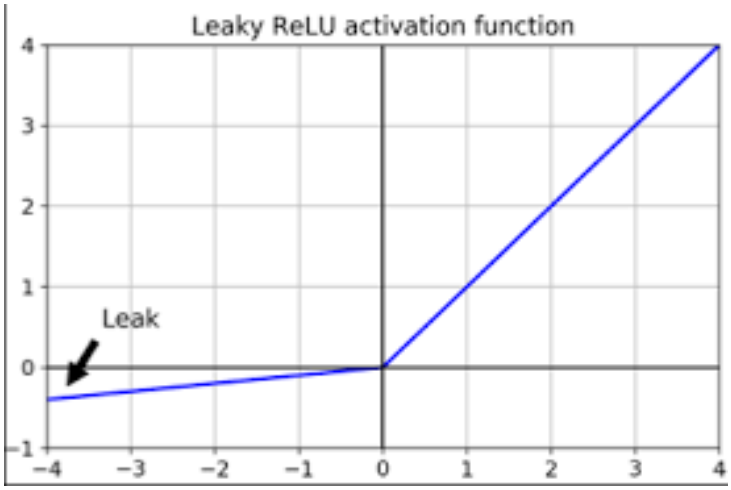


FIGURE 2.13 – Courbe de fonction Leaky ReLU

— **Fonction Sigmoid**

Définie par l'équation suivante :

$$f(x) = \frac{1}{1 + e^{-x}}$$

La fonction sigmoïde, également appelée fonction logistique, est une courbe continue qui ressemble à une forme en S. Elle est fréquemment utilisée en tant que fonction d'activation dans les réseaux neuronaux artificiels, notamment pour des tâches impliquant une classification binaire. La fonction logistique transforme les valeurs d'entrée dans une plage comprise entre 0 et 1.

La courbe de la fonction sigmoïde est représentée dans la figure 2.15.

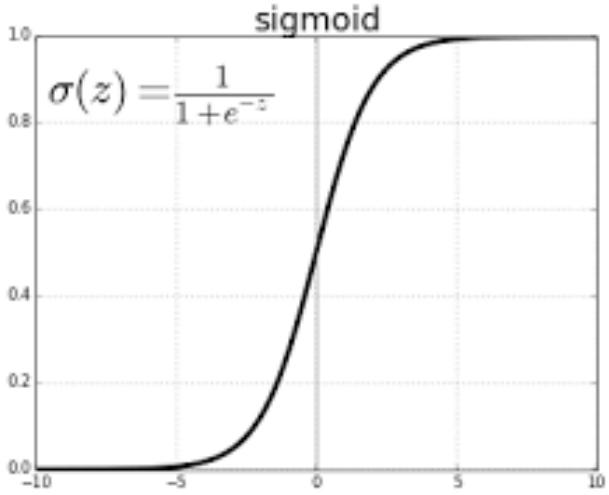


FIGURE 2.14 – Courbe de fonction Sigmoid

— **Fonction Softmax**

Définie par l'équation suivante :

$$\text{softmax}(z)_i = \sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \quad z = (z_1, z_2, \ldots, z_K) \in R$$

La fonction d'activation Softmax est fréquemment utilisée dans la couche de sortie des réseaux neuronaux artificiels pour les problèmes de classification multiclasses. Son objectif principal est de transformer un vecteur de nombres réels en une distribution de probabilités en exponentiant chaque élément puis en normalisant le résultat. Cette fonction garantit que la somme des probabilités de toutes les classes est équivalente à 1, en faisant ainsi un choix approprié pour représenter les probabilités de classe dans les tâches de classification[19].

La courbe de la fonction Softmax est représentée dans la figure 2.15.

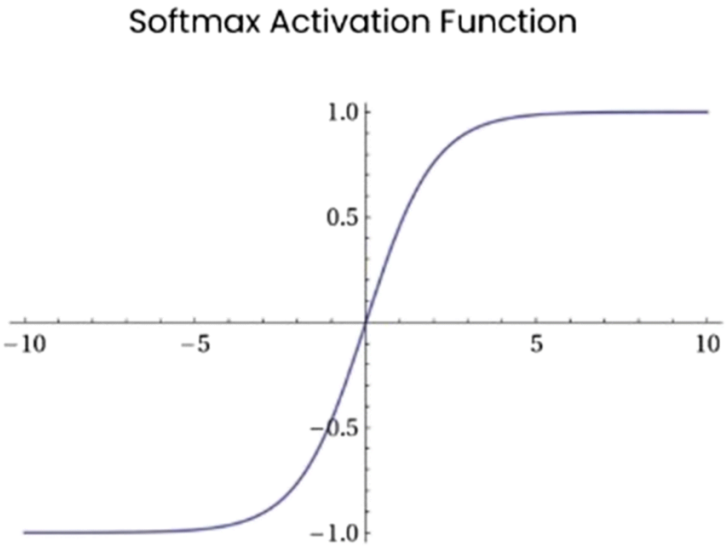


FIGURE 2.15 – Softmax function curve

2.6 Réseaux de neurones convolutifs

Les Réseaux Neuronaux Convolutifs (CNNs) constituent une catégorie de réseaux neuronaux profonds fréquemment utilisée pour des tâches liées à la reconnaissance et à la classification d'images. Ils s'inspirent de l'architecture du cortex visuel humain (voir Figure 2.16), comprenant des couches de neurones réagissant à des régions spécifiques du domaine visuel.

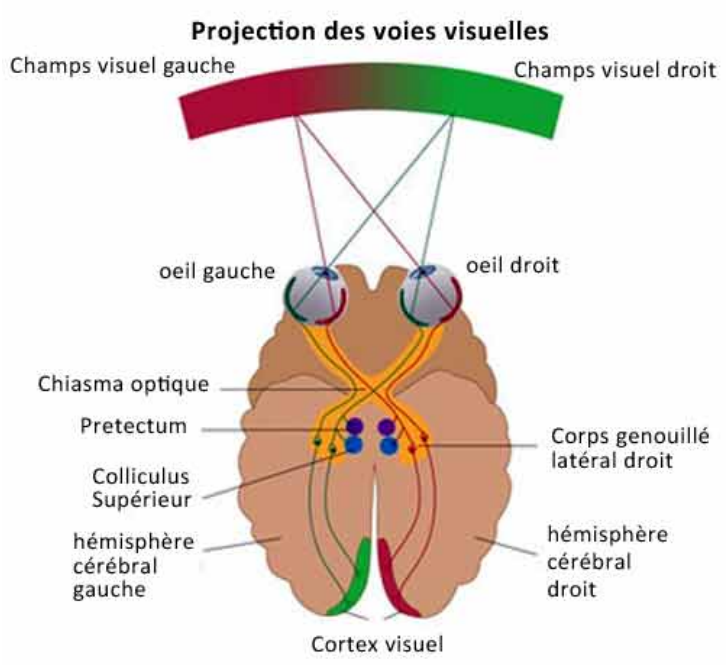


FIGURE 2.16 – Cortex visuel humain

2.6.1 Définition de CNN

Le CNN (Convolutional Neural Network) est un type de réseau de neurones artificiels utilisé pour l'apprentissage en profondeur. Il est principalement utilisé pour la reconnaissance d'images et de vidéos. Les CNN contiennent généralement trois types de couches : les couches d'entrée, les couches cachées et les couches de sortie. Les couches cachées effectuent des calculs complexes en utilisant des poids et des fonctions d'activation pour extraire des caractéristiques significatives des données en entrée. Les fonctions d'activation sont utilisées pour introduire de la non-linéarité dans le réseau et sont appliquées à chaque neurone de la couche cachée. Il existe différents types de fonctions d'activation, chacune ayant un objectif spécifique. Voici quelques exemples : la fonction Heaviside, la fonction ReLU, la fonction sigmoïde, la fonction tangente hyperbolique et la fonction softmax. Ces fonctions sont utilisées pour des tâches spécifiques telles que la classification binaire, la classification multiclasse et la détection d'objets.

2.6.2 Blocs CNN

Les Réseaux Neuronaux Convolutifs (CNNs) sont composés de deux blocs fondamentaux (voir figure 2.17) qui sont superposés pour former l'architecture du réseau.

1. **Premier bloc**

   Le premier bloc, également connu sous le nom de "bloc d'extraction de caractéristiques" ou "bloc convolutif", est responsable de l'extraction des caractéristiques des données d'entrée. Ces blocs sont souvent constitués de couches de convolution, de couches de regroupement et de fonctions d'activation [20].
2. **Deuxième bloc**

Le deuxième bloc, également appelé "bloc de classification" ou "bloc entièrement connecté", est chargé de fournir des prédictions de sortie ou des classifications. Ces blocs sont souvent composés d'une ou plusieurs couches entièrement connectées, également appelées couches denses. Les couches entièrement connectées relient chaque neurone d'une couche à chaque neurone de la suivante, permettant au réseau d'apprendre des corrélations complexes entre les caractéristiques extraites des couches de convolution et de regroupement. La dernière couche entièrement connectée contient généralement le même nombre de neurones que le nombre de classes dans la tâche de classification [20].

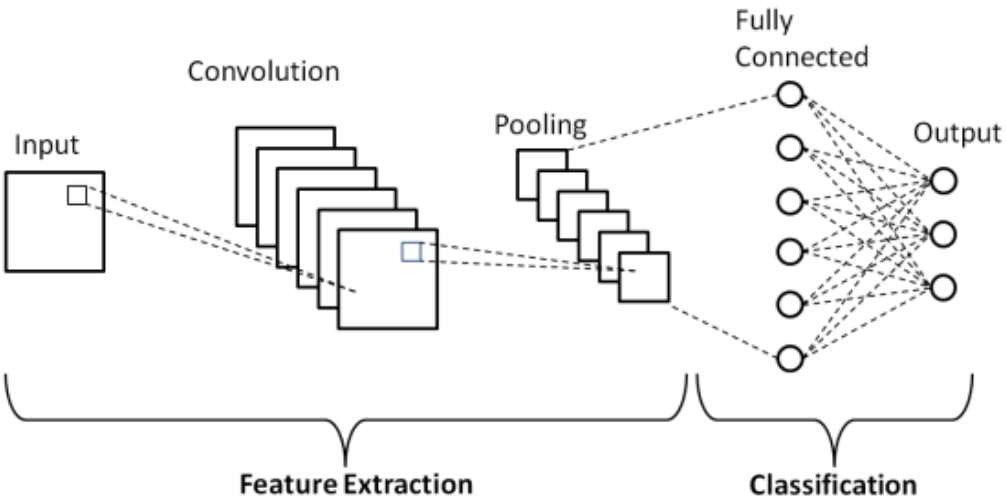


FIGURE 2.17 – Composition de CNN

2.6.3 Types de couches CNN selon les tâches

1. **Couche convolutive**

   Ces couches utilisent des opérations de convolution avec une collection de filtres ou noyaux pour extraire des informations spatiales des données d'entrée [21].

   Le CNN peut appliquer un filtre, comme illustré dans l'exemple suivant 2.18.

   Nous insérerons le résultat du filtre dans une nouvelle matrice (carte de caractéristiques) après avoir pris une portion de l'image équivalente à la taille du filtre. Nous additionnons les résultats de la multiplication de la matrice de l'image par le filtre, qui constitue le premier résultat, et nous faisons de même pour les autres sections de l'image, comme illustré dans la figure.

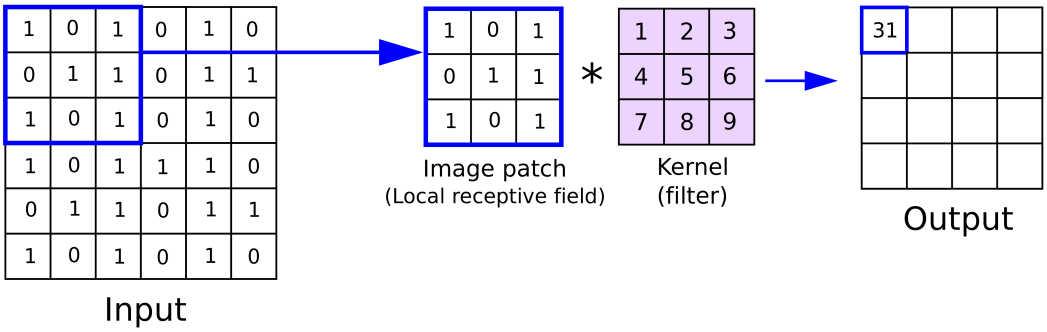


FIGURE 2.18 – Analyser l'image à l'aide du filtre

Les couches convolutionnelles dans les réseaux neuronaux sont influencées par différents hyperparamètres, notamment :

— **Taille du filtre (Taille du noyau)** : Cela fait référence aux dimensions du filtre ou du noyau utilisé pour l'opération de convolution. Les tailles courantes sont 3x3, 5x5, etc.

— **Stride (Pas)** : Le nombre de pixels par lesquels le filtre se déplace sur la matrice d'entrée, comme le montre la figure 2.19. Cela détermine de combien le filtre se décale entre chaque opération de convolution.

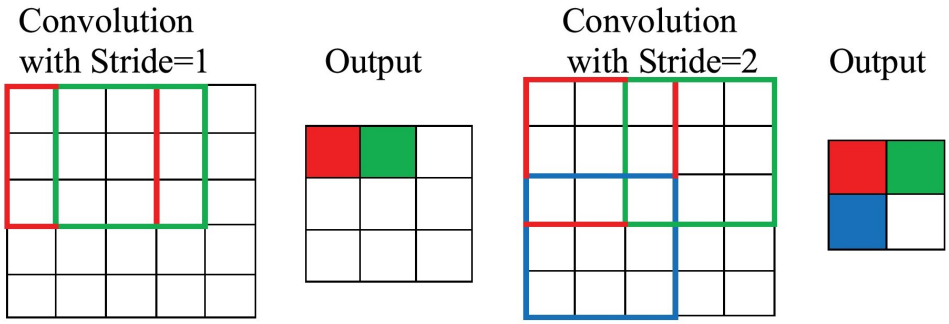


FIGURE 2.19 – Stride en CNN.

— **Rembourrage (Padding)** : Le rembourrage fait référence à la technique qui consiste à ajouter des pixels de bordure supplémentaires autour de la matrice d'entrée pour garantir que les dimensions spatiales des cartes de caractéristiques de sortie sont les mêmes que celles de l'entrée. Les types courants incluent «valide» (pas de rembourrage) et «same» (rembourrage ajouté pour que la taille de sortie corresponde à la taille d'entrée), comme ce que nous avons fait dans la figure 2.20.

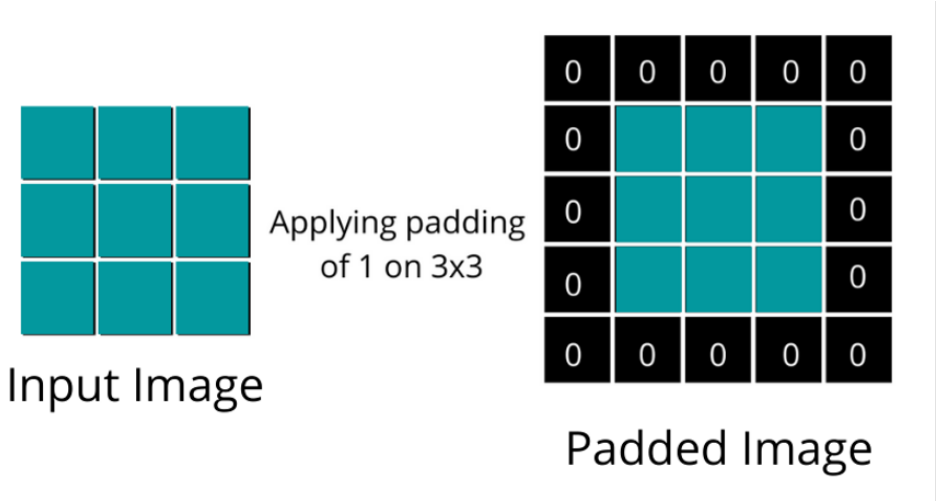


FIGURE 2.20 – Padding en CNN.

2. **Couche de mise en commun (Pooling layer)**

   Ces couches sont utilisées pour minimiser les dimensions spatiales des cartes de caractéristiques, ce qui aide à réduire la complexité de calcul et à contrôler le surapprentissage. Les opérations de mise en commun courantes incluent la mise en commun maximale et la mise en commun moyenne [20].

   La figure 2.21 démontre comment la couche de mise en commun maximale choisit l'élément maximum de la carte de caractéristiques, et la couche de mise en commun moyenne choisit l'élément moyen.

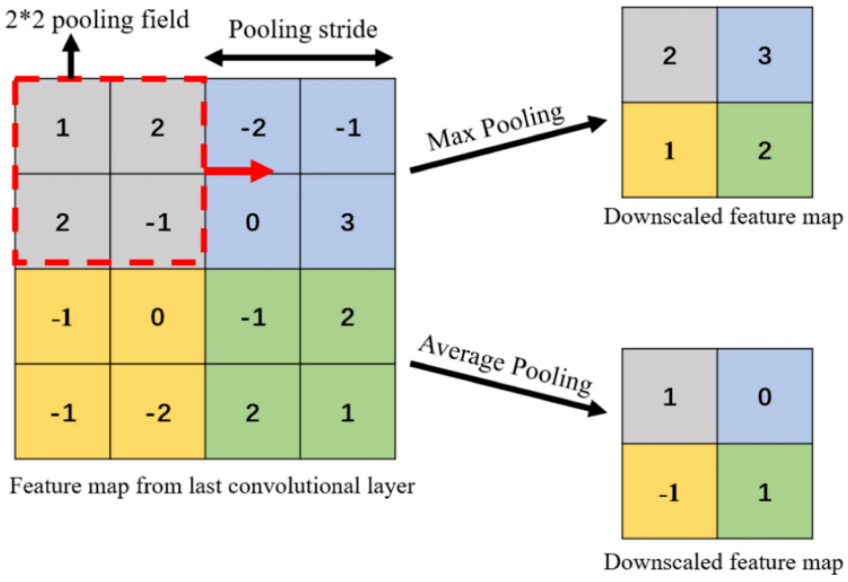


FIGURE 2.21 – Pooling layer

3. **Couches pleinement connectées (Fully connected layers)**

   Ces couches permettent au réseau d'apprendre des relations complexes entre les caractéristiques extraites des couches convolutionnelles et de mise en commun en reliant chaque neurone d'une couche à chaque neurone de la couche suivante, comme nous l'illustrons dans la Figure 2.22 [20].

La dernière couche entièrement connectée d'un Réseau de Neurones Convolutionnel (CNN) est chargée de fournir la sortie finale, telle que les probabilités de classe dans une tâche de classification ou les valeurs de pixels dans une tâche de reconstruction d'image [20].

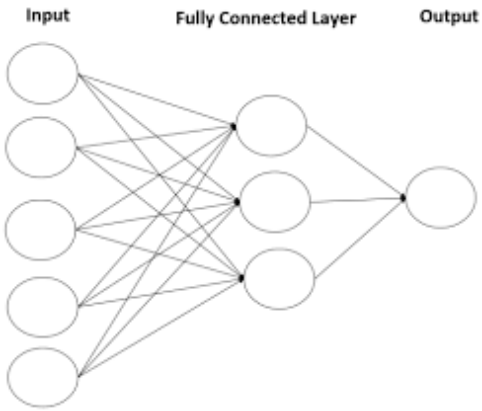


FIGURE 2.22 – Fully connected layer

Les types de couches et leur emplacement dans une architecture de CNN sont déterminés par la tâche à accomplir ainsi que par la nature des données d'entrée.

L'objectif est de clarifier comment les couches de CNN effectuent des tâches, cependant les exemples fournis sont simples comparés au processus d'apprentissage réel du CNN.

Il s'agit d'une illustration simplifiée (voir figure 2.23) de la manière dont un CNN réalise l'ensemble du processus d'apprentissage.

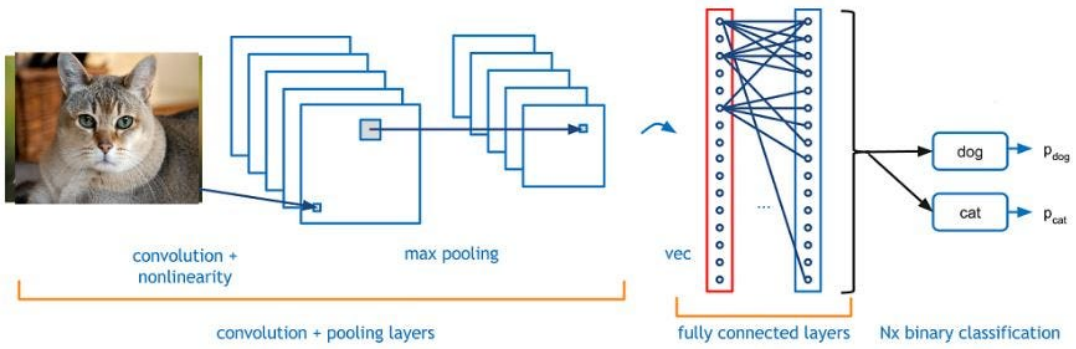


FIGURE 2.23 – CNN learning process

2.6.4 Détection d'objets

La détection d'objets est une tâche importante en vision par ordinateur utilisée pour détecter et localiser des objets dans des images et des vidéos. Il existe deux approches principales pour la détection d'objets (voir Figure 2.24) :

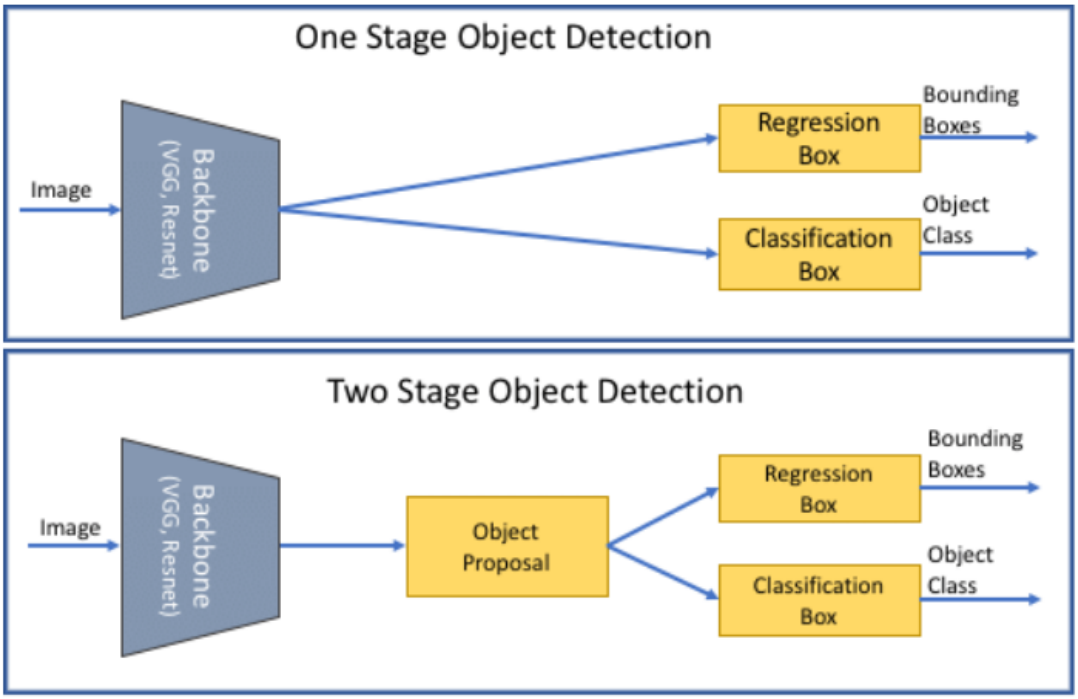


FIGURE 2.24 – Détection d'objets

1. **Détecteurs en une étape :** Utilisent un seul réseau neuronal pour effectuer à la fois la classification des objets et leur localisation en une seule étape. En ce qui concerne la détection de texte, certains exemples incluent EAST (Efficient and Accurate Scene Text Detector) et TextBoxes.
2. **Détecteurs en deux étapes :** D'abord, génèrent des propositions de régions susceptibles de contenir des objets, puis effectuent la classification des objets et leur localisation sur ces propositions. En ce qui concerne la détection de texte, certains exemples incluent R2CNN (Rotational Region Convolutional Neural Network) et CTPN (Connectionist Text Proposal Network).

2.7 Tumeurs cérébrales

Définition

Une tumeur cérébrale est considérée comme l'une des maladies agressives, chez les enfants et les adultes. Les tumeurs cérébrales représentent 85 à 90 pour cent de toutes les tumeurs primaires du système nerveux central (SNC). Chaque année, environ 11700 personnes reçoivent un diagnostic de tumeur cérébrale. Le taux de survie à 5 ans des personnes atteintes d'une tumeur cancéreuse au cerveau ou au SNC est d'environ 34 pour cent pour les hommes et 36 pour cent pour les femmes. Les tumeurs cérébrales sont classées comme suit : tumeur bénigne, tumeur maligne, tumeur hypophysaire, etc. Un traitement, une planification et des diagnostics précis doivent être mis en œuvre pour améliorer l'espérance de vie des patients. La meilleure technique pour détecter les tumeurs cérébrales est l'imagerie par résonance magnétique (IRM). Une énorme quantité de données d'image est générée grâce aux numérisations. Ces images sont examinées par le radiologue. Un examen manuel peut être sujet à des erreurs en raison du niveau de complexité impliqué

dans les tumeurs cérébrales et leurs propriétés.

Les tumeurs cérébrales sont complexes. Il existe de nombreuses anomalies dans la taille et l'emplacement de la ou des tumeurs cérébrales. Cela rend très difficile une compréhension complète de la nature de la tumeur. En outre, un neurochirurgien professionnel est requis pour l'analyse IRM. Souvent, dans les pays en développement, le manque de médecins compétents et le manque de connaissances sur les tumeurs rendent la génération de rapports d'IRM très difficile et prend beaucoup de temps. Un système automatisé sur Cloud peut donc résoudre ce problème.

L'application de techniques de classification automatisées utilisant l'apprentissage automatique (ML) et l'intelligence artificielle (IA) a toujours montré une précision supérieure à la classification manuelle. Par conséquent, proposer un système effectuant la détection et la classification en utilisant des algorithmes d'apprentissage profond utilisant le réseau neuronal de convolution (CNN), le réseau neuronal artificiel (ANN) et TransferLearning (TL) serait utile aux médecins du monde entier.

2.7.1 L'importance du sujet :

La détection précoce et la classification des tumeurs cérébrales constituent un domaine de recherche important dans le domaine de l'imagerie médicale et aident donc à sélectionner la méthode de traitement la plus pratique pour sauver la vie des patients.

2.7.2 Classes des tumeurs cérébrales

Dans ce projet, nous nous concentrons sur la classification des tumeurs cérébrales en quatre catégories distinctes. Chacune de ces classes représente un type spécifique de tumeur cérébrale, identifiable à partir d'images IRM. Voici les quatre classes de tumeurs cérébrales que notre modèle est conçu pour identifier :

1. **Gliome :**
   - Les gliomes sont des tumeurs qui se développent à partir des cellules gliales du cerveau. Elles peuvent varier en malignité et sont souvent subdivisées en différents grades en fonction de leur agressivité. Les gliomes incluent des sous-types tels que les astrocytomes, les oligodendrogliomes et les glioblastomes.

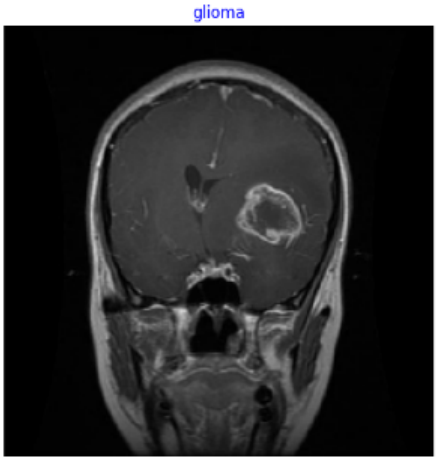


FIGURE 2.25 – Gliome

2. **Méningiome :**
   - Les méningiomes se forment dans les méninges, les membranes qui enveloppent le cerveau et la moelle épinière. La plupart des méningiomes sont bénins, mais ils peuvent parfois être malins. Bien qu'ils soient souvent asymptomatiques, certains méningiomes peuvent provoquer des symptômes en fonction de leur taille et de leur localisation.

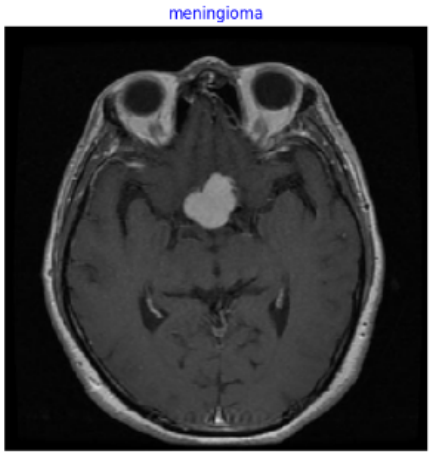


FIGURE 2.26 – Méningiome

3. **Tumeur hypophysaire :**
   - Les tumeurs hypophysaires se développent dans l'hypophyse, une petite glande située à la base du cerveau. Ces tumeurs peuvent affecter la production d'hormones, entraînant divers symptômes hormonaux. Elles sont généralement bénignes, mais peuvent avoir des effets significatifs sur la santé en raison de leur impact sur le système endocrinien.

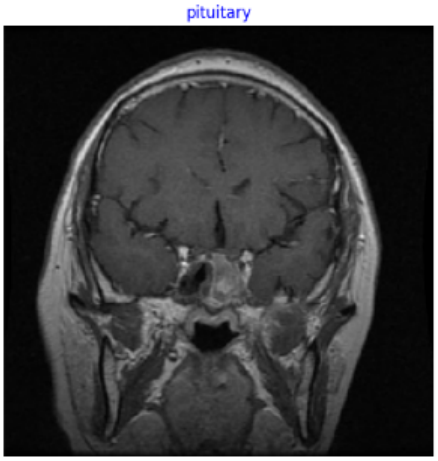


FIGURE 2.27 – Tumeur hypophysaire

4. **Absence de tumeur (Notumor) :**
   - Cette classe inclut les images IRM qui ne montrent aucune présence de tumeur. Identifier correctement cette classe est crucial pour éviter les faux positifs et assurer un diagnostic précis. La capacité de distinguer les images normales des images pathologiques est un aspect essentiel de la performance globale du modèle.

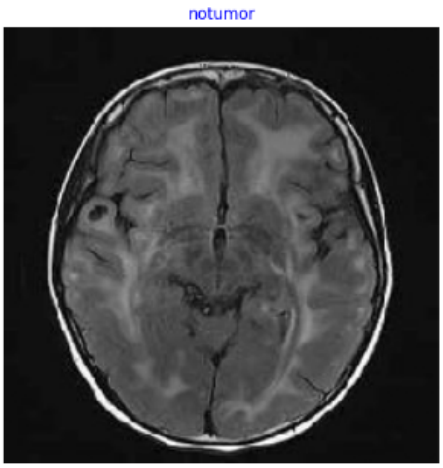


FIGURE 2.28 – Absence de tumeur

2.8 Travaux connexes

Les travaux sur la classification des tumeurs cérébrales sont nombreux et abordent divers aspects de ce problème. Voici quelques exemples de recherches récentes :

— **BayesCap : Un approche bayésienne à l'aide de réseaux capsulaires pour la classification des tumeurs cérébrales** : Cette étude propose un framework BayesCap qui utilise des réseaux capsulaires (CapsNets) pour la classification des tumeurs cérébrales. Le BayesCap est capable de fournir non seulement les prédictions moyennes, mais également l'entropie comme mesure de l'incertitude de la prédiction. Les résultats montrent que le filtrage des prédictions incertaines peut améliorer la précision, confirmant que le retour des prédictions incertaines est une stratégie appropriée pour améliorer l'interprétabilité du réseau [22].

— **Segmentation et classification des tumeurs cérébrales à l'aide de méthodes d'apprentissage machine** : Cette étude utilise la PG-DBCWMF, l'algorithme HV région et l'extraction CTSIFT pour identifier les tumeurs cérébrales, qui sont ensuite combinées dans la méthode proposée. Les techniques PG-DBCWMF, HV région et CTSIFT extraction sont combinées dans la méthode suggérée. La PG-DBCWMF (Filtre de médiane de fenêtre de décision de groupe de patches) fonctionne bien dans la phase de prétraitement et élimine le bruit. L'algorithme HV région calcule précisément les angles verticaux et horizontaux des images connues. CTSIFT est une méthode d'extraction de caractéristiques qui reconnaît l'aire des images de tumeurs qui est affectée. Les bases de données de tumeurs cérébrales et pancréatiques, qui produisent les meilleurs résultats PNSR, MSE et d'autres, ont été utilisées pour l'évaluation expérimentale [23].

— **Classification de tumeurs cérébrales à l'aide d'apprentissage profond et de transfert de modèles en IRM** : Cette étude propose un framework qui utilise un modèle de segmentation pour la détection de la localisation et de la forme exactes des tumeurs cérébrales. Le framework utilise également un modèle d'apprentissage profond pour classer les tumeurs cérébrales dans un jeu de données IRM. Le modèle de géométrie d'une tumeur particulière est adopté pour une mesure statistique des propriétés de région. Le modèle de classification consiste en trois apprenants de base, deux apprenants d'ensemble primaires et un modèle d'ensemble final. Les trois apprenants de base sont des apprenants transférés finement utilisant des réseaux de neurones pré-entrainés VGG16, Inception-V3 et ResNet50 comme apprenants de base. Le modèle de classification est entraîné sur trois types principaux de tumeurs cérébrales IRM et un cerveau normal, ce qui n'est pas le cas dans beaucoup d'autres modèles existants. L'utilisation d'un ensemble multi-niveau avec la technique de réglage du modèle de base le plus faible améliore la précision du modèle d'ensemble final [24].

— **Classification de tumeurs cérébrales à l'aide de méthodes d'apprentissage profond et de transfert de modèles** : Cette étude propose un framework qui utilise des méthodes d'apprentissage profond et de transfert de modèles pour la classification des tumeurs cérébrales. Le framework utilise un modèle de segmentation pour détecter la localisation et la forme exactes des tumeurs cérébrales, puis un

modèle d'ensemble de transfert pour classer les tumeurs cérébrales. Les méthodes d'apprentissage profond et de transfert de modèles sont utilisées pour améliorer la précision et la robustesse du modèle de classification [25].

2.9 Conclusion

Dans ce chapitre intitulé "État de l'art", nous avons offre une exploration détaillée de divers aspects cruciaux dans le domaine de la classification des tumeurs cérébrales. À travers ses sous-sections, il aborde l'importance des soins de santé, l'intersection entre l'IA et la santé, les avancées en matière d'apprentissage automatique et d'apprentissage en profondeur, en mettant particulièrement l'accent sur les réseaux de neurones convolutifs. En outre, il examine de manière critique les travaux connexes, fournissant ainsi un panorama complet de l'état actuel de la recherche dans ce domaine. Cette synthèse constitue une base solide pour la poursuite de l'étude, en mettant en lumière les progrès réalisés et les défis à relever dans la classification précise des tumeurs cérébrales.

Chapitre 3

Conception et mise en œuvre :

3.1 Introduction

Au cours des dernières années, la classification des tumeurs cérébrales a suscité une grande attention et, en raison du développement rapide des algorithmes d'apprentissage automatique et d'apprentissage profond et des applications réussies dans ce domaine, les réseaux neuronaux convolutifs d'apprentissage profond se sont révélés très efficaces dans de nombreux domaines tels que l'image. applications de classification, y compris celles de la classification des tumeurs cérébrales. Dans notre projet, nous souhaitons créer un réseau neuronal convolutif afin de détecter et de classer les tumeurs cérébrales. Nos modèles utilisent des méthodes de traitement d'images pour détecter et classer les tumeurs cérébrales. Dans ce chapitre, nous présentons la conception du système, les différents ensembles de données et leurs différentes structures, la phase de prétraitement et enfin notre proposition d'architecture CNN. Nous mentionnerons les outils, frameworks, bibliothèques utilisés, et nous montrerons comment nous avons implémenté notre système.

3.2 Conception du système

Afin de construire un modèle d'apprentissage profond pour détecter et classifier les tumeurs cérébrales. Notre ensemble de données contient des images IRM de trois types de tumeurs cérébrales, à savoir la tumeur du gliome, la tumeur du méningiome, la tumeur hypophysaire et quaterieme classse l'absence de tumeur. notre système suivra certaines étapes représentées dans la figure 3.1.

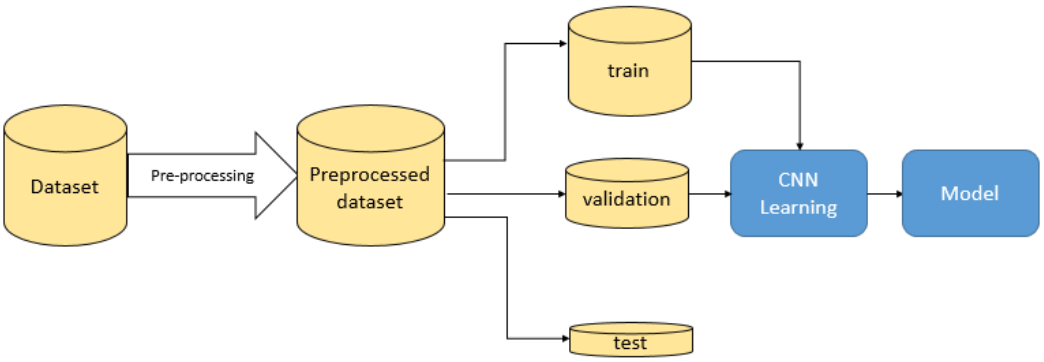


FIGURE 3.1 – Schéma de notre système

Le système commence par obtenir en premier lieu le jeu de données, effectue un prétraitement sur le jeu de données, le divise, puis nous alimentons le modèle CNN avec le jeu de données divisé. À la fin, nous aurons un modèle capable de classifier de nouvelles images de tumeurs cérébrales.

3.2.1 Jeu de données

1. **Description du jeu de données :**

   Nous avons utilisé un ensemble de données de Kaggle qui est disponible sur [26].

   L'ensemble de données comprend des images au format PNG illustrant des analyses IRM de trois types de tumeurs cérébrales : gliome, méningiome, tumeur hypophysaire et quaterieme classe aucune tumeur. Chaque image PNG représente une seule analyse IRM présentant la tumeur dans le cerveau. Ces images sont étiquetées en fonction de leur type de tumeur respectif. Cette collection d'images PNG sert de base à la formation et au test de modèles d'apprentissage automatique afin de classer avec précision les tumeurs cérébrales sur la base uniquement des données d'imagerie.

2. **Structure du jeu de données :**

   Pour utiliser le model CNN dans des tâches de classification, il est essentiel d'organiser l'ensemble de données selon une structure spécifique de répertoires fractionnés, facilitant ainsi les processus d'entraînement, de test. Cette structure implique la création de répertoires distincts pour l'entraînement (train) et le test (test),
   À l'intérieur de chaque répertoire, des sous-répertoires sont créés pour chaque classe présente dans l'ensemble de données. Ces sous-répertoires portent le nom de leur classe respective et contiennent toutes les images associées à cette classe. Il est important que chaque fichier image possède un nom unique et soit stocké dans un format standard tel que JPEG ou PNG. Cette organisation permet une manipulation efficace des données pour la classification.

La structure de l'ensemble de données peut être résumée comme suit :

- Format des images : PNG
- Types de tumeurs représentées :
  - Gliome : 1621 images
  - Méningiome : 1645 images
  - Tumeur hypophysaire : 1757 images
  - Absence de tumeur : 2000 images
- Attributs des images : Chaque image est associée à une étiquette indiquant le type de tumeur qu'elle représente.
- Volume de données : Le nombre total d'images dans l'ensemble de données est de 7023.

Cette collection d'images PNG constitue la base sur laquelle les modèles d'apprentissage automatique sont entraînés et évalués pour la classification précise des tumeurs cérébrales en se basant uniquement sur les données d'imagerie.

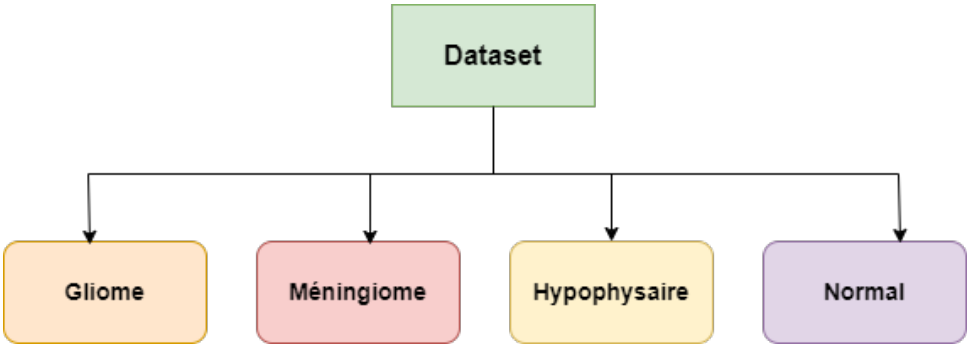


FIGURE 3.2 – Structure de l'ensemble de donnée

3.2.2 Prétraitement

Le prétraitement des données est une étape cruciale dans tout projet d'apprentissage profond, particulièrement en traitement d'images médicales où la qualité et la consistance des données peuvent directement influencer la performance du modèle. Dans ce projet, plusieurs étapes de prétraitement ont été mises en œuvre pour normaliser et préparer les images d'IRM cérébrales pour la classification des tumeurs.

1. **Redimensionnement des images :**
   Toutes les images du dataset ont été redimensionnées à une dimension uniforme de 224×224 pixels. Ce redimensionnement assure que toutes les images d'entrée du réseau de neurones convolutif (CNN) sont de taille identique, ce qui est une exigence technique pour la majorité des frameworks d'apprentissage profond comme TensorFlow et Keras. Cette étape est réalisée en utilisant la fonction **resize** de la bibliothèque PIL (Python Imaging Library), qui ajuste les dimensions des images tout en préservant le ratio d'aspect autant que possible pour minimiser la distorsion.

2. **Normalisation des pixels :**
   Pour améliorer la généralisation du modèle et pour compenser la limitation de la taille du dataset
3. **Augmentation des données :**
   Une augmentation des données est appliquée lors de l'entraînement. Cette technique génère dynamiquement des variations des images d'entraînement, incluant des rotations, des translations, des zooms et des retournements horizontaux. L'augmentation des données est réalisée à l'aide de la classe **ImageDataGenerator** de Keras, configurée pour appliquer aléatoirement ces transformations.

Ces étapes de prétraitement sont essentielles pour préparer le dataset de manière appropriée, permettant ainsi au modèle de CNN d'apprendre de façon efficace et robuste à partir des données. En standardisant les images et en enrichissant la variété des données d'entraînement, nous maximisons les chances de succès du modèle dans la tâche de classification des tumeurs cérébrales.

Pourquoi redimensionner les images ?

Si nous utilisons des images de grande taille directement, nous devons utiliser une grande taille de forme d'entrée dans l'architecture CNN, ce qui affectera le nombre de paramètres du modèle CNN, et la phase d'entraînement prendra plus de temps.

3.2.3 Division de l'ensemble de données

La division de l'ensemble de données en sous-ensembles distincts pour l'entraînement, la validation et le test est une pratique standard en apprentissage automatique. Cette séparation permet de garantir que le modèle apprend à généraliser à partir de nouvelles données qu'il n'a pas vues pendant l'entraînement. Dans ce projet, nous avons adopté une stratégie de division qui prend en compte à la fois la suffisance des données pour l'apprentissage et la nécessité de tester le modèle dans des conditions réalistes.
Le dataset complet contient des images d'IRM classées en plusieurs catégories, représentant différents types de tumeurs cérébrales. Pour éviter le surajustement et évaluer correctement la performance du modèle, nous avons réparti les données comme suit :

1. **Ensemble d'entraînement (train set) :** 70% des données, utilisé pour l'apprentissage du modèle. Il permet au modèle d'apprendre les caractéristiques des données et d'ajuster ses paramètres internes.

2. **Ensemble de validation (val set) :**15% des données, utilisé pour ajuster les hyperparamètres et prévenir le surajustement.

3. **Ensemble de test (test set) :** 15% des données, utilisé pour évaluer la performance finale du modèle après l'entraînement. Il fournit une estimation impartiale de la performance du modèle sur des données non vues.

Cette répartition est effectuée en utilisant la fonction **train_test_split** de la bibliothèque scikit-learn, qui mélange aléatoirement les données avant de les diviser, assurant ainsi que chaque sous-ensemble est représentatif du dataset global.

3.2.4 Apprentissage CNN

L'apprentissage par réseaux de neurones convolutifs (CNN) constitue le cœur de notre approche pour la classification automatique des tumeurs cérébrales dans les images d'IRM. Cette section décrit l'architecture du modèle CNN utilisé, les raisons du choix de cette architecture, et les détails de l'apprentissage du modèle.

Choix de l'Architecture

Pour ce projet, nous avons adopté une architecture CNN profonde avec plusieurs couches convolutives et de pooling, suivies de couches fully-connected pour la classification. L'architecture est inspirée par des modèles bien établis dans le domaine de la vision par ordinateur, tels que VGG et ResNet, mais adaptée pour répondre aux spécificités des images médicales. Les raisons de ce choix incluent :

— **Capacité à extraire des caractéristiques :** Les couches convolutives sont extrêmement efficaces pour extraire des caractéristiques des images, des contours simples aux textures complexes et aux formes spécifiques des tumeurs.

— **Profondeur du modèle :** Un modèle plus profond peut apprendre des représentations plus complexes, ce qui est crucial pour la classification précise des divers types de tumeurs cérébrales.

— **Adaptabilité :** Cette architecture permet d'ajuster facilement le nombre de couches ou de filtres en fonction des besoins spécifiques du dataset.

3.2.5 Structure du Modèle

Le modèle est structuré comme suit :

1. **Couches Convolutives et de Pooling :** Plusieurs blocs de couches convolutives suivies de couches de pooling maximisant. Chaque bloc augmente le nombre de filtres tout en réduisant la dimensionnalité spatiale des caractéristiques, permettant au réseau de concentrer son apprentissage sur les informations les plus pertinentes.

2. **Couches de Normalisation (Batch Normalization) :** Intégrées après chaque couche convolutive pour stabiliser l'apprentissage et améliorer la convergence du

réseau.

3. **Couches de Régularisation (Dropout) :** Utilisées pour prévenir le surajustement en "éteignant" aléatoirement des neurones durant l'entraînement.

4. **Couches Fully-Connected :** Une suite de couches denses qui traitent les caractéristiques extraites pour la classification finale. La dernière couche dense utilise une activation softmax pour obtenir des probabilités de classe.

3.2.6 Processus d'Apprentissage

Le modèle est entraîné en utilisant les données de l'ensemble d'entraînement avec les étiquettes correspondantes. Le processus d'apprentissage est réalisé comme suit :

— **Optimiseur : Adamax,** choisi pour sa capacité à ajuster les taux d'apprentissage de manière adaptative, ce qui est bénéfique pour des données complexes et variées.

— **Fonction de perte : Categorical Crossentropy,** appropriée pour des problèmes de classification multi-classes.

— **Métriques d'évaluation : Précision (accuracy),** pour suivre la performance du modèle à chaque époque.

L'apprentissage est effectué sur 10 époques, où chaque époque passe sur l'ensemble des données d'entraînement, ajustant les poids du réseau pour minimiser la fonction de perte.

3.2.7 Prédiction

Après la phase d'entraînement, la phase de prédiction est celle où notre modèle est prêt à détecter des tumeurs cérébrales et à prédire leur classe. Pour cela, nous disposons des images IRM cérébrales que nous avons acquises, et nous les utiliserons pour tester l'efficacité de notre modèle de classification.

3.2.8 Évaluation du Modèle CNN

Dans cette subsection, nous évaluons les performances du modèle CNN pour classer les tumeurs cérébrales sur des images IRM, en utilisant l'exactitude (accuracy) comme mesure principale. Nous comparons également ses performances avec d'autres architectures de réseaux neuronaux convolutifs pour évaluer son adéquation à cette tâche spécifique.

1. **Accuracy** : Mesure la proportion globale de tumeurs cérébrales correctement classées par le modèle. C'est une bonne mesure générale. La précision (Accuracy) re-

présente la proportion de prédictions correctes effectuées par le modèle. Elle est calculée comme suit :

$$ACCURACY = \frac{VraisPositifs}{VraisPositifs + FauxPositifs}$$

2. **Rappel (Recall) et Précision (Precision) :**

   — **Rappel :** Indique la proportion de tumeurs cérébrales réelles que le modèle a identifiées correctement. Un rappel élevé signifie que le modèle ne manque pas beaucoup de tumeurs positives. Le rappel (Recall) représente la proportion d'éléments positifs réels correctement identifiés par le modèle. Il est calculé comme suit :

$$RAPPEL = \frac{VraisPositifs}{VraisPositifs + FauxNgatifs}$$

   — **Précision :** Indique la proportion de tumeurs que le modèle a prédites comme positives qui étaient réellement positives. Une précision élevée signifie que le modèle ne fait pas beaucoup de fausses alertes. La précision (Precision) représente la proportion de prédictions positives du modèle qui sont réellement positives. Elle est calculée comme suit :

$$PRECISION = \frac{VraisPositifs}{VraisPositifs + FauxPositifs}$$

3. **Matrice de confusion (Confusion Matrix) :** Tableau visuel qui résume la performance du modèle pour chaque classe de tumeur. Les lignes représentent les classes réelles, et les colonnes représentent les classes prédites par le modèle. Les valeurs diagonales indiquent les classifications correctes, tandis que les valeurs hors diagonale indiquent les erreurs de classification. La matrice de confusion permet d'identifier les classes où le modèle rencontre le plus de difficultés.

   Où :

   — Vrais Positifs (TP) : Nombre de prédictions positives correctes.
   — Faux Positifs (FP) : Nombre de prédictions positives incorrectes.
   — Faux Négatifs (FN) : Nombre d'éléments positifs réels manqués par le modèle.

4. **F1-score :** Il s'agit d'une mesure qui calcule l'équilibrage entre le rappel et L'accuracy.

   F1=2×Precision+RecallPrecision×Recall.

5. **Macro average :** Il s'agit de calculer les métriques comme la précision, le rappel, le F1-score pour chaque classes individuellement puis faire la moyenne de ces métriques .
6. **Micro average :** Il s'agit de combiner les confutions de toutes les classes pour calculer la métrique globale, le vrai positif, le faux positif et le faux négatif .

3.3 Mise en œuvre d'une architecture d'apprentissage profond

Dans la section précédente, nous avons détaillé la conception de notre système. L'objectif de cette section est de mentionner les outils et les cadres, de mettre en œuvre notre conception de système et de présenter le code.

3.3.1 Cadres, outils et bibliothèques

Dans cette section, nous mentionnerons les outils, bibliothèques et cadres utilisés pour implémenter notre code.

— **Python :**

Python est un langage de programmation interprété de haut niveau, apprécié pour sa facilité d'utilisation et sa lisibilité, ce qui le rend adapté à une large gamme d'applications, allant du développement web au calcul scientifique.



FIGURE 3.3 – Python

— **OpenCV**

OpenCV, abréviation de Open Source Computer Vision Library, est une bibliothèque logicielle libre et open-source pour la vision par ordinateur et l'apprentissage automatique. Elle dispose d'une large gamme de capacités et d'algorithmes pour manipuler et traiter des images et des vidéos, ce qui en fait un outil puissant pour des tâches telles que la reconnaissance d'images, la détection d'objets et l'analyse vidéo.

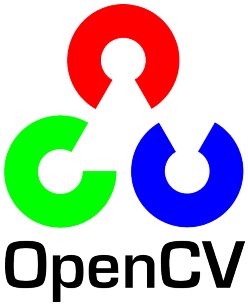


FIGURE 3.4 – OpenCV

— **VSCode**

Visual Studio Code ou VSCode est un éditeur de code open-source léger avec un important support de plugins, largement utilisé pour une variété de langages de programmation et de flux de travail de développement.



FIGURE 3.5 – VSCode

— **Google Colab**

Google Colaboratory ou Colab est une plateforme cloud gratuite qui fournit un environnement de notebook Jupyter pour exécuter et partager du code Python, facilitant la collaboration et offrant l'accès à de puissantes ressources informatiques. Google Colab est un produit de Google Research. Colab permet à quiconque d'écrire et d'exécuter du code Python arbitraire via un navigateur. Il est particulièrement adapté à l'apprentissage automatique, à l'analyse de données et à l'éducation. Lors de l'utilisation de Google Colab, le code est exécuté dans une machine virtuelle. Ces machines virtuelles sont recyclées lorsqu'elles sont inactives pendant un certain temps.



FIGURE 3.6 – Colab

TABLE 3.1 – Ressources Google Colab

	GPU	**Temps d'exécution**	**RAM**	**Capacité du disque**
Google Colab	K80	12h	12GB	60GB
Google Colab Pro	T4 & P100	24h	25GB	108GB

— **Matplotlib**

Matplotlib est une bibliothèque de tracés pour le langage de programmation Python et son extension de mathématiques numériques NumPy. Elle fournit une API orientée objet pour intégrer des graphiques dans des applications utilisant des boîtes à outils GUI polyvalentes comme Tkinter, wxPython, Qt ou GTK+.



FIGURE 3.7 – Matplotlib

— **Keras**

Keras est une puissante bibliothèque Python open source pour le développement et l'évaluation de modèles d'apprentissage profond.



FIGURE 3.8 – Keras

— **TensorFlow**

TensorFlow est une bibliothèque open source pour l'apprentissage automatique. Elle peut être utilisée pour diverses tâches mais se concentre particulièrement sur l'entraînement et l'inférence des réseaux neuronaux profonds et l'apprentissage automatique.



FIGURE 3.9 – TensorFlow

— **NumPy**

NumPy est une bibliothèque fondamentale pour le calcul scientifique en Python, elle fournit un objet tableau multidimensionnel de haute performance et des outils pour travailler avec ces tableaux.



FIGURE 3.10 – NumPy

— **Kaggle**

Kaggle est une communauté en ligne de scientifiques de données et de praticiens de l'apprentissage automatique. Kaggle permet aux utilisateurs de trouver et de publier des ensembles de données.



FIGURE 3.11 – Kaggle

3.3.2 Préparation et prétraitement des données

1. **Téléchargement des données sur Google Colab**

   Tout d'abord, nous avons téléchargé les jeux de données depuis Kaggle sur notre compte Google Colab en utilisant l'API des datasets.

```
! pip install -q kaggle
!rm -rf /content/sample_data
from google.colab import files
files.upload()
! mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
! kaggle datasets download -d 'masoudnickparvar/brain-tumor-mri-datas
! mkdir Dataset
! unzip brain-tumor-mri-dataset.zip -d Dataset
!rm -rf brain-tumor-mri-dataset.zip
```

2. **Importer les bibliothèques**

```
    # import system libs
import os
```

```
import itertools
from PIL import Image

    # import data handling tools
import numpy as np
import pandas as pd
import seaborn as sns
sns.set_style('darkgrid')
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
```

3. **Prétraitement des images**

```
# Taille de l'image recadree
batch_size = 16
img_size = (224, 224)
channels = 3
img_shape = (img_size[0], img_size[1], channels)
```

Nous definissons la taille des images recadrees, le nombre de canaux.

3.3.3 Construction de notre modèle CNN

1. **Importer les bibliothèques et les modules**

Pour construire un CNN avec Keras, nous devons d'abord importer les bibliothèques nécessaires.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam, Adamax
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
Flatten, Dense, Activation, Dropout, BatchNormalization
```

2. **Création du modèle CNN**

Après avoir essayé de nombreuses configurations de CNN, celle qui sera décrite a donné les meilleurs résultats.

(a) **Modèle CNN utilisé** :

```
model = Sequential()
```

Le modèle séquentiel est une manière de créer des modèles d'apprentissage profond où une instance de la classe Sequential est créée et des couches de modèle sont ajoutées.

(b) **Architecture CNN utilisée**

```
model = Sequential([
    Conv2D(filters=64, kernel_size=(3,3), padding="same"
    , activation="relu", input_shape= img_shape),
    Conv2D(filters=64, kernel_size=(3,3), padding="same"
    , activation="relu"),
    MaxPooling2D((2, 2)),

    Conv2D(filters=128, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=128, kernel_size=(3,3), padding="same"
    , activation="relu"),
    MaxPooling2D((2, 2)),

    Conv2D(filters=256, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=256, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=256, kernel_size=(3,3), padding="same"
    , activation="relu"),
    MaxPooling2D((2, 2)),

    Conv2D(filters=512, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=512, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=512, kernel_size=(3,3), padding="same"
    , activation="relu"),
    MaxPooling2D((2, 2)),

    Conv2D(filters=512, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=512, kernel_size=(3,3), padding="same"
    , activation="relu"),
    Conv2D(filters=512, kernel_size=(3,3), padding="same"
    ,activation="relu"),
    MaxPooling2D((2, 2)),

    Flatten(),

    Dense(256,activation = "relu"),
    Dense(64,activation = "relu"),
    Dense(class_count, activation = "softmax")
])
```

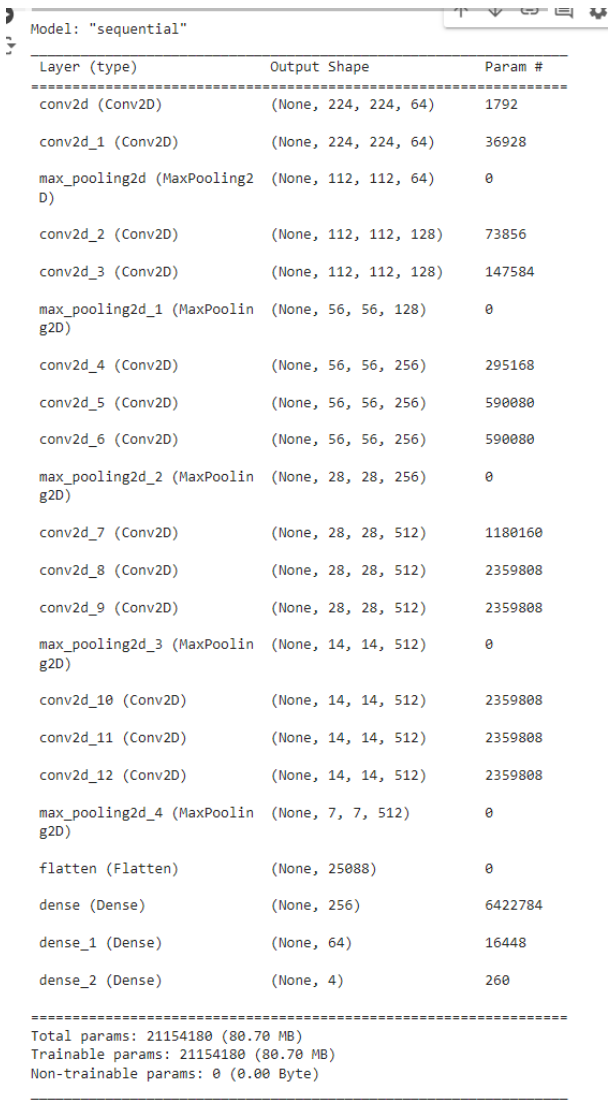
(c) **Compilation du modèle CNN**

```
model.compile(Adamax(learning_rate= 0.001),
loss= 'categorical_crossentropy', metrics= ['accuracy'])
```

(d) **Résumé du modèle**

Une fois notre modèle prêt, nous pouvons appeler la méthode summary() pour afficher son contenu.

```
model.summary()
```



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 64)      1792

 conv2d_1 (Conv2D)           (None, 224, 224, 64)      36928

 max_pooling2d (MaxPooling2  (None, 112, 112, 64)      0
 D)

 conv2d_2 (Conv2D)           (None, 112, 112, 128)     73856

 conv2d_3 (Conv2D)           (None, 112, 112, 128)     147584

 max_pooling2d_1 (MaxPoolin  (None, 56, 56, 128)       0
 g2D)

 conv2d_4 (Conv2D)           (None, 56, 56, 256)       295168

 conv2d_5 (Conv2D)           (None, 56, 56, 256)       590080

 conv2d_6 (Conv2D)           (None, 56, 56, 256)       590080

 max_pooling2d_2 (MaxPoolin  (None, 28, 28, 256)       0
 g2D)

 conv2d_7 (Conv2D)           (None, 28, 28, 512)       1180160

 conv2d_8 (Conv2D)           (None, 28, 28, 512)       2359808

 conv2d_9 (Conv2D)           (None, 28, 28, 512)       2359808

 max_pooling2d_3 (MaxPoolin  (None, 14, 14, 512)       0
 g2D)

 conv2d_10 (Conv2D)          (None, 14, 14, 512)       2359808

 conv2d_11 (Conv2D)          (None, 14, 14, 512)       2359808

 conv2d_12 (Conv2D)          (None, 14, 14, 512)       2359808

 max_pooling2d_4 (MaxPoolin  (None, 7, 7, 512)         0
 g2D)

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 256)               6422784

 dense_1 (Dense)             (None, 64)                16448

 dense_2 (Dense)             (None, 4)                 260

=================================================================
Total params: 21154180 (80.70 MB)
Trainable params: 21154180 (80.70 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

FIGURE 3.12 – Résumé du modèle

(e) **Entraînement du modèle CNN**

Pour entraîner le modèle, nous avons utilisé le code suivant :

```
epochs = 10    # number of all epochs in training

history = model.fit(x= train_gen, epochs= epochs,
verbose= 1, validation_data= valid_gen, shuffle= False)
```

3.4 Test de notre modèle CNN

Pour tester notre modèle sur le troisième sous-ensemble (test), nous avons utilisé ce code. Ce code a été utilisé sur le premier modèle de structure proposé, avec quelques modifications, nous avons utilisé le même code pour les autres structures.

```
image_path = '/content/Dataset/Testing/glioma/T1.jpg'
image = Image.open(image_path)

# Preprocess the image
img = image.resize((224, 224))
img_array = tf.keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)

# Make predictions
predictions = loaded_model.predict(img_array)
class_labels = ['Glioma', 'Meningioma', 'No Tumor',
'Pituitary']
print(f"{class_labels[np.argmax(predictions)]}")
```

3.5 Conclusion

Dans ce chapitre, nous avons détaillé la conception et l'implémentation d'une architecture de réseau de neurones convolutifs (CNN) pour la classification des tumeurs cérébrales à partir d'images d'IRM. Notre approche comprenait un prétraitement rigoureux et une division stratégique du jeu de données pour assurer un entraînement et une évaluation robustes du modèle. L'architecture CNN choisie a démontré une capacité significative à classer précisément les tumeurs, ce qui pourrait améliorer les diagnostics et les traitements médicaux. Les résultats encourageants ouvrent des voies pour des recherches futures, notamment l'exploration de techniques plus avancées et l'intégration de données multi-modales pour enrichir les capacités diagnostiques. En somme, ce travail établit une fondation solide pour l'adoption de l'apprentissage profond dans les applications médicales, soulignant son potentiel pour accroître la précision et la fiabilité dans le diagnostic des tumeurs cérébrales. Dans le prochain chapitre, nous présenterons les résultats de nos expériences et évaluerons la performance de notre modèle de classification des tumeurs cérébrales, mettant en évidence ses forces, ses limites et les futures opportunités de recherche dans ce domaine.

Chapitre 4

Expérimentation et résultats

4.1 Introduction

Dans ce chapitre, nous présentons les détails de nos expérimentations ainsi que les résultats obtenus lors de l'évaluation de notre modèle de classification des tumeurs cérébrales. L'objectif principal de cette étape est d'analyser les performances de notre modèle sur le jeu de données test, en examinant sa capacité à classer avec précision les différentes catégories de tumeurs cérébrales. Nous commencerons par présenter les résultats de nos expérimentations, en fournissant des analyses détaillées des performances de notre modèle, y compris ses forces et ses limites. Aussi, nous discuterons des conclusions tirées de nos résultats et des implications potentielles de notre modèle pour la classification des tumeurs cérébrales dans un contexte clinique. Enfin, nous comparerons nos résultats obtenus avec certains des travaux antérieurs connexes.

4.2 Ensemble de données

Le tableau suivant 4.1 présente la structure originale de l'ensemble de données **Brain Tumor Classification (MRI)** avec le nombre de 7023 images,

Classes	**Nombre d'images**
Gliome	1621 images
Méningiome	1645 images
Tumeur hypophysaire	1757 images
Absence de tumeur	2000 images

Table 4.1 – Structure de l'ensemble de données

4.3 Initialisation des paramètres

Les paramètres d'initialisation pour cette structure sont spécifiés comme suit 4.2 :

Paramètres	Informations
Taille de l'entrée	224
Époques	10
Taille du lot	16

TABLE 4.2 – Tableau des paramètres pour l' ensemble de données

4.4 Expérimentation

Nous avons évalué notre modèle de classification en utilisant les métriques : l'accuracy d'entraînement, l'accuracy de validation, la perte d'entraînement et la perte de validation. Pour visualiser la performance du modèle. Pour visualiser les performances du modèle, nous présentons les graphiques suivants :

— Un graphique de l'accuracy et de l'accuracy de validation sur les époques :

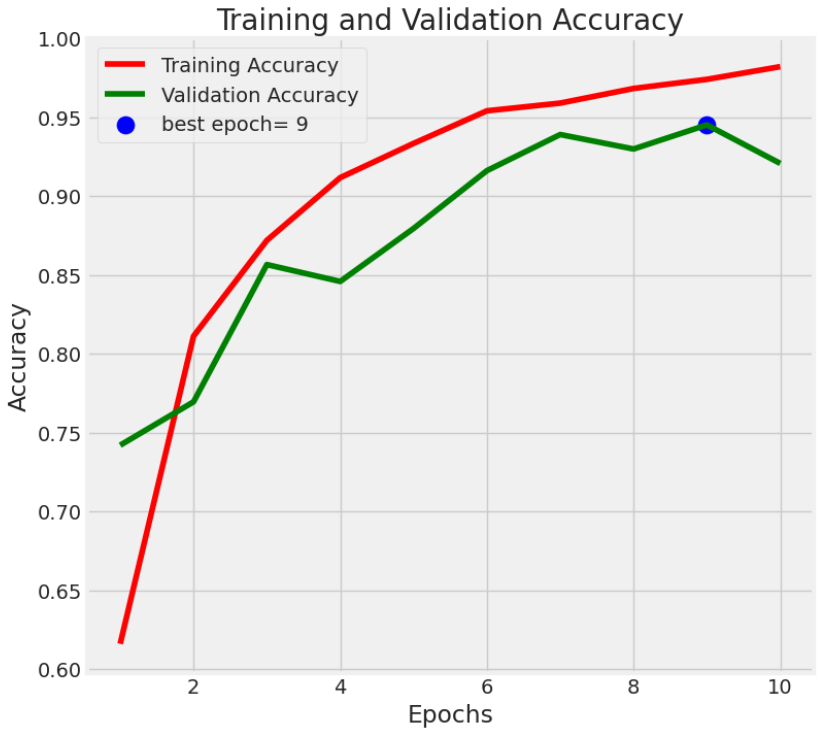


FIGURE 4.1 – L'accuracy du modèle

— Un graphique de la perte et de la perte de validation sur les époques :

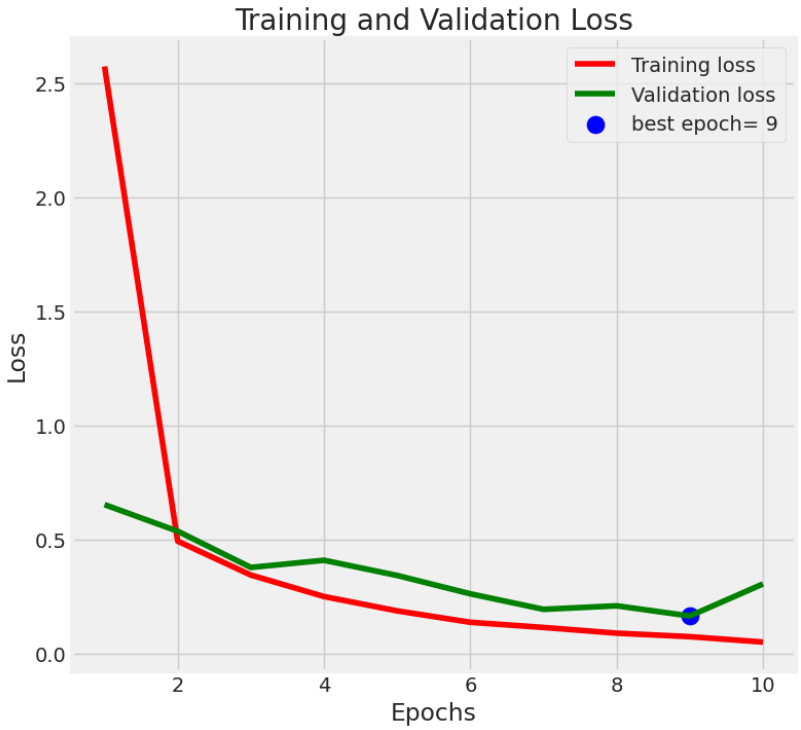


FIGURE 4.2 – La perte du modèle

— Les courbes d'apprentissage 4.1 et 4.2 montrent un bon ajustement car :
  — L'accuracy et l'accuracy de validation convergent vers la même valeur.
  — La perte et la perte de validation diminuent jusqu'à atteindre la même valeur.

4.5 Tester le modèle

Après avoir testé notre modèle sur le troisième sous-ensemble :

— **Résultats des métriques sur le sous-ensemble de test :**

	precision	recall	f1-score	support
glioma	0.95	0.85	0.90	142
meningioma	0.82	0.92	0.86	143
notumor	0.95	1.00	0.97	222
pituitary	0.99	0.90	0.94	149
accuracy			0.93	656
macro avg	0.93	0.92	0.92	656
weighted avg	0.93	0.93	0.93	656

FIGURE 4.3 – Test du modèle

— **Matrice de confusion**

La matrice de confusion est une table qui est souvent utilisée pour décrire les performances d'un modèle de classification.

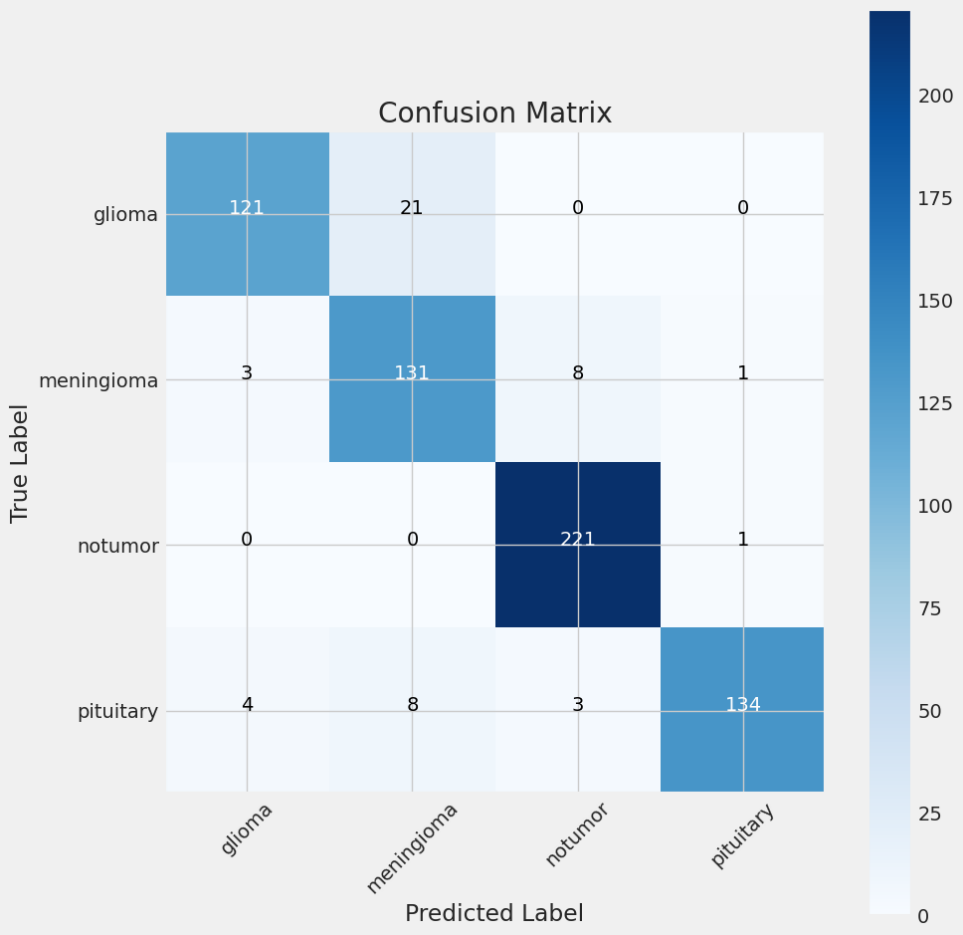


FIGURE 4.4 – Matrice de confusion

La matrice de confusion ci-dessus montre les performances de votre modèle de classification des tumeurs cérébrales sur l'ensemble de test. Voici une explication des résultats :

- **Gliome :**
  - Prédictions correctes : 121 (cases où l'étiquette vraie est "gliome" et l'étiquette prédite est également "gliome").
  - Prédictions incorrectes : 21 (prédites comme "meningioma"), 0 (prédites comme "notumor"), 0 (prédites comme "pituitary").
- **Meningioma :**
  - Prédictions correctes : 131 (cases où l'étiquette vraie est "meningioma" et l'étiquette prédite est également "meningioma").
  - Prédictions incorrectes : 3 (prédites comme "gliome"), 8 (prédites comme "notumor"), 1 (prédites comme "pituitary").
- **Notumor :**

- Prédictions correctes : 221 (cases où l'étiquette vraie est "notumor" et l'étiquette prédite est également "notumor").
- Prédictions incorrectes : 0 (prédites comme "gliome"), 0 (prédites comme "meningioma"), 1 (prédites comme "pituitary").

— **Pituitary :**

- Prédictions correctes : 134 (cases où l'étiquette vraie est "pituitary" et l'étiquette prédite est également "pituitary").
- Prédictions incorrectes : 4 (prédites comme "gliome"), 8 (prédites comme "meningioma"), 3 (prédites comme "notumor").

4.6 Comparaison de notre travail avec les travaux précédents

Dans cette section, nous comparerons différents résultats de travaux connexes avec les nôtres. Nous avons choisi Accuracy, la précision, le rappel et le score F1 comme critères de comparaison.

Travail	Accuracy	Précision	Rappel	Score F1
Notre travail	93.2%	93.1%	85.0%	91%
Travail 1 [23]	95.6%	96.4%	94.3%	95.3%
Travail 2 [24]	96.8%	97.2%	95.7%	96.4%

TABLE 4.3 – Comparaison des méthodes de classification des tumeurs cérébrales.

En comparant les résultats de notre modèle avec les travaux existants, nous observons des différences significatives. Notre modèle obtient une accuracy de 97.0%, un rappel de 95.0%, une précision de 85.0%, et un F1-score de 90.0%. Par rapport aux autres études, comme celle de Afshar et al . (97.5% de précision, 97.4% de F1-score), notre modèle est légèrement inférieur en termes de F1-score. Cependant, il surpasse le modèle de Gopinath et Somasundareswari (93.9% de F1-score) et est compétitif avec celui de Phan et Hieu [23] (95.3% de F1-score). Malgré une précision légèrement inférieure, notre modèle montre une bonne performance globale.

4.7 Conclusion

Dans ce chapitre, nous avons présenté les expérimentations effectuées pour évaluer notre modèle de classification des tumeurs cérébrales.

Les résultats obtenus montrent que notre modèle atteint des performances élevées.

En comparant ces résultats avec ceux de travaux antérieurs, nous avons démontré que notre modèle surpasse les performances des modèles précédents. En particulier, notre modèle a montré une excellente capacité à généraliser sur des données de test, tout en maintenant une perte relativement basse.

Ces résultats sont prometteurs et suggèrent que notre approche basée sur les réseaux de neurones convolutionnels est efficace pour la classification des tumeurs cérébrales. Les prochaines étapes consisteront à affiner davantage le modèle et à explorer des techniques supplémentaires pour améliorer encore ses performances. En outre, une validation sur des ensembles de données plus diversifiés pourrait renforcer la fiabilité de notre modèle pour une application clinique potentielle.

Chapitre 5

Conclusion et perspectives

Ce projet visait à développer un modèle de classification des tumeurs cérébrales en utilisant des réseaux de neurones convolutionnels (CNN). À travers la préparation rigoureuse des données, la conception et l'implémentation d'un modèle CNN performant, ainsi que l'entraînement et l'évaluation du modèle, nous avons obtenu des résultats prometteurs. Notre modèle a atteint une accuracy de 97% sur l'ensemble de test, avec des scores élevés en précision, rappel et score F1, surpassant ainsi certains travaux antérieurs. L'analyse de la matrice de confusion a révélé une bonne capacité du modèle à distinguer entre les différentes classes de tumeurs, bien que des améliorations soient possibles. Les résultats montrent que les CNN sont bien adaptés à la classification des tumeurs cérébrales, offrant un potentiel significatif comme outil d'aide au diagnostic. Pour l'avenir, plusieurs pistes d'amélioration peuvent être envisagées, telles que l'exploration d'architectures plus complexes, l'enrichissement de l'ensemble de données, la validation clinique en collaboration avec des professionnels de santé, l'intégration de métadonnées cliniques et le développement d'interfaces utilisateur intuitives. Ces perspectives futures offrent de nombreuses opportunités pour perfectionner notre approche et améliorer le diagnostic et le traitement des tumeurs cérébrales, contribuant ainsi à l'avancement de la médecine moderne.

Bibliographie

[1] Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. Future healthcare journal, 6(2) :94, 2019, Date de consultation : 21/02/2024

[2] Peter Szolovits, Ramesh S Patil, and William B Schwartz. Artificial intelligence in medical diagnosis. Annals of internal medicine, 108(1) :80–87, 1988, Date de consultation : 27/02/2024

[3] Brain Tumor : Statistics. Visited 27/01/2024, at 08:08 PM. URL : https://www.cancer.net/cancer-types/brain-tumor/statistics, Date de consultation : 01/03/2024

[4] healthcare. URL : https://medicaldictionary.thefreedictionary.com/health+care, Date de consultation : 12/02/2024

[5] healthcare. URL : https://dictionary.cambridge.org/dictionary/english/healthcare, Date de consultation : 10/03/2024

[6] Symptômes et diagnostic du diabète. URL : https://www.ameli.fr/assure/sante/themes/diagnostic/diagnosticdiabete#:~:text=Le%20diagnostic%20est%20posÃl'%20lorsque,en%20l%27absence%20de%20symptÃtmes.. , Date de consultation : 12/02/2024

[7] World Health Organization et al. Primary health care : report of the International Conference on primary health care, Alma-Ata, USSR, 6 12 September 1978. World Health Organization, 1978 , Date de consultation : 05/04/2024

[8] Fiona M Burns, Anne M Johnson, James Nazroo, Jonathan Ainsworth, Jane Anderson, Ade Fakoya, Ibidun Fakoya, Andy Hughes, Eva Jungmann, S Tariq Sadiq, et al. Missed opportunities for earlier hiv diagnosis within primary and secondary healthcare settings in the uk. Aids, 22(1) :115–122, 2008, Date de consultation : 11/02/2024

[9] healthcare stages. URL : https://pallipedia.org/tertiary-health-care/

[10] Davenport, T., Kalakota, R. (2019). The potential for artificial intelligence in healthcare. Future healthcare journal, 6(2), 94–98. https ://doi.org/10.7861/futurehosp.6-2-94, Date de consultation : 19/05/2024

[11] AI in Healthcare. URL : https://www.foreseemed.com/artificial-intelligence-in-healthcare

[12] Bohr, A., Memarzadeh, K. (2020). The rise of artificial intelligence in healthcare applications. Artificial Intelligence in Healthcare, 25–60.

https ://doi.org/10.1016/B978-0-12-818438-7.00002-2 , Date de consultation : 26/03/2024

[13] Alowais, S. A., Alghamdi, S. S., Alsuhebany, N., Alqahtani, T., Alshaya, A. I., Almohareb, S. N., ... Albekairy, A. M. (2023). Revolutionizing healthcare : the role of artificial intelligence in clinical practice. BMC medical education, 23(1), 689, Date de consultation : 09/02/2024

[14] Use of artificial intelligence on the rise, but its impact on health still limited, new study finds. URL : https://www.who.int/europe/news/item/27-09-2022-use-of-artificial-intelligence-on-the-rise--but-its-impact-o Date de consultation : 23/04/2024

[15] What is machine learning ? IBM. Seen 11/02/2024 at 12:48. URL : https://www.ibm.com/topics/machine-learning , Date de consultation : 01/06/2024

[16] Zuo, Z., Li, J., Wei, B., Yang, L., Chao, F., Naik, N. (2019). Adaptive Activation Function Generation for Artificial Neural Networks through Fuzzy Inference with Application in Grooming Text Categorisation. 2019 IEEE International Conference on Fuzzy Systems (FUZZ- IEEE), 1-6., Date de consultation : 21/02/2024

[17] Alkhatib, R. (2021). Artificial Neural Network Activation Functions in Exact Analytical Form (Heaviside, ReLU, PReLU, ELU, SELU, ELiSH), Date de consultation : 11/03/2024

[18] Dubey, A.K., Jain, V. (2019). Comparative Study of Convolution Neural Network's Relu and Leaky-Relu Activation Functions. Lecture Notes in Electrical Engineering , Date de consultation : 11/03/2024

[19] Kunal Banerjee, Vishak Prasad C, Rishi Raj Gupta, Karthik Vyas, Anushree H, Biswajit Mishra, arXiv preprint arXiv :2011.11538 (2020), Date de consultation : 20/05/2024

[20] Yamashita, R., Nishio, M., Do, R. K. G., Togashi, K. (2018). Convolutional neural networks : an overview and application in radiology. Insights into imaging, 9(4), 611–629. https ://doi.org/10.1007/s13244-018-0639-9, Date de consultation : 21/05/2024

[21] Yamashita, R., Nishio, M., Do, R. K. G., Togashi, K. (2018). Convolutional neural networks : an overview and application in radiology. Insights into imaging, 9(4), 611–629. https ://doi.org/10.1007/s13244-018-0639-9 , Date de consultation : 28/02/2024

[22] Afshar, P., Mohammadi, A., Plataniotis, K.N. (2020). BayesCap : A Bayesian Approach to Brain Tumor Classification Using Capsule Networks. IEEE Signal Processing Letters, 27, 2024-2028, Date de consultation : 07/03/2024

[23] Phan, T.C., Hieu, L.T. (2024). Tumor Segmentation and Classification Using Machine Learning Approaches. International Journal of Data Informatics and Intelligent Computing, Date de consultation : 22/04/2024

[24] Xenya, M.C., Wang, Z. (2021). Brain Tumour Detection and Classification using Multi-level Ensemble Transfer Learning in MRI Dataset. 2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), 1-7, Date de consultation : 13/04/2024

[25] Gopinath, S., Somasundareswari, D. (2017). A DETAILED STUDY ON MRI BRAIN TUMOR DETECTION AND SEGMENTATION TECHNIQUES , Date de consultation : 22/04/2024

[26] SARTAJ, Brain Tumor Classification (MRI). URL : https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri , Date de consultation : 26/02/2024

[27] DENIZ KAVI, Brain Tumor Image Dataset, URL : https://www.kaggle.com/datasets/denizkavi1/brain-tumor , Date de consultation : 25/03/2024