PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTARY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY YAHIA FARES OF MEDEA

Faculty of **T**echnology

Department of **E**lectrical **E**ngineering

Master's End of Study Project

Branch: Telecommunication

Specialty: Telecommunications System

Study of an Intelligent System for the Detection of Brain Tumors based on MRI Images

Presented By :

KEDENDEL ISHAK

BESSNACI RAWANE

Proposed and Supervised By:

Mr. REBIAI Mohamed

Mr. BERKANI Mohamed Rafik Aymene

Promotion 2024

Acknowledgements

Before we begin, praise and gratitude to Allah the Almighty for giving us the will, health, patience, and courage to achieve our goal.

We are pleased to warmly and sincerely thank our supervisors, **Mr. REBIAI Mohamed** and **Mr. BERKANI Mohamed Rafik Aymene**. Despite their heavy tasks, they were kind enough to guide us and carefully correct our work. We are particularly grateful for their constant presence throughout the year. They are entitled to all our gratitude for their great patience, the advice, technical and material assistance, and the encouragement they generously provided, and for which we hold them in deep respect.

We also extend our sincere thanks and appreciation to all the people who contributed directly or indirectly to the completion of this work. Our thanks go to all the members of our families, especially our dear parents, brothers, sisters, and friends, for their moral support and for creating a pleasant atmosphere around us.

And of course, we do not forget our brothers in Gaza by praying for them at all times.

Dedication

In the name of **God**, the Most Gracious, the Most Merciful, and with the grace of God alone, this project has been completed by His grace upon us, through His infinite blessings, we pray to God that this work will bring us closer to Him.

Knowledge is not about attaining a status or position in society, nor about acquiring a certificate; it is about the journey of knowledge guiding you towards God Almighty.

In this way, we do not forget to thank those who worked hard to raise us and provided us with the appropriate conditions that enabled us to succeed.

We dedicate our work to

dear **father**.

And my dear **mother**.

Without them, I would not be where I am now, despite all their sacrifices and prayers.

Their love and support throughout my academic journey.

To my dear **sisters** and my **brother**.

To my dear uncles and aunts from my **Kedendel** and **Ouahrani** family.

To my dear grandfathers and grandmothers.

To everyone who participated directly or indirectly in developing this work.

To all my friends, without exception, with whom I spent enjoyable moments.

KEDENDEL ISHAK

Dedication

After a study march that lasted for years, I carried a lot of difficulties, hardship and fatigue, today I am standing on the threshold of my graduation. I pick the fruits of my tiredness and raise my hat with pride. Praise be to you before you are satisfied, and praise be to you if you are satisfied, and praise be to you after the satisfaction of your mother, and I succeeded me to complete this work and achieve my dream ... I dedicate this success to myself first and then to

The mirror that made me an ambitious girl and loves challenges and my first role model from which I knew the strength and self-confidence for those who are satisfied, her satisfaction creates for me success (my mother), may God prolong your life in health and wellness

You will get rid of cancer because I know you strong and fight.

To that great man who brought out the most beautiful thing in me and always encouraged me to reach my ambitions, a man who taught me life in the most beautiful way and did everything he could and did not stimp (my father), may God keep you our shadow and support for us

To my dear sisters (Ashwak, Sirine, Douaa, Khadija, Inaam) to my fixed rib and the safety of my days to those who intensified, my supported them to the best of my days and its description

To my aunt, that person has the soul of my mother again, despite the distance, my heart has not been forget you or my pen (Nadia boukssira)

To my dear grandfathers and grandmothers

I dedicate my graduation to my family in Bessnaci to the source of my strength, the supporters, the supporters to those whose hands extended in times of weakness, who believe in my courage. I am grateful to all of you.

Thank you to all those who wanted us to break, so God made them a bridge to cross for the better.

BESSNACI RAWANE

ملخص

وفقًا للجمعية الأمريكية لأورام الدماغ (ABTA)، يعيش حوالي 700000 شـــخص في الولايات المتحدة مع ورم أولي في المخ والجهاز العصـبي المركزي. في حين أن بعض الأورام قد تكون حميدة، إلا أن الكثير منها يمكن أن يتسـلل إلى أنسـجة المخ الطبيعية ويتطور إلى سرطان. نهدف في هذا المشروع إلى تقديم معلومات شاملة عن أورام المخ وإظهار مدى إمكانية الحصول على صور دقيقة ومفصلة للغاية للدماغ البشري باستخدام تقنية التصوير بالرنين المغناطيسي، ثم سنناقش الشبكات العصـبية التلافيفية (CNNs) و وظائفها، بعد الحصـول على مجموعة من بين 3000 صـورة من صـور التصـوير بالرنين المغناطيسـي وتطبيق المعالجة المسـبقة، سـنقوم باختبار هذه الصـور باسـتخدام نظام ذكي قمنا بتطويره يعتمد على الشـبكات العصـبية التلافيفية ثنائية الأبعاد (CNN) للكشـف عن صـور الدماغ السليمة مقابل تلك التي تحتوي على أورام. سيعمل هذا النظام على تمكين التصـنيف الثنائي من خلال اسـتخدام طرق اسـتخراج الميزات، وتحديدًا شـبكات CNN، والتي أعطتنا نتائج ممتازة، حيث حققت دقة بنسـبة 99% وخسـارة قدرها 0.02%. وأخيرًا، سـنقوم بمقارنة نتائجنا مع دراسـات أخرى ركزت على حالات مماثلة لتعزيز اكتشاف الورم وتحليله.

الكلمات المفتاحية: ورم في المخ، الشبكة العصـبية التلافيفية، التصـوير بالرنين المغناطيسـي، سـرطان الدماغ، التعلم الالي، التصوير الطبي، نظام ذكي.

Résumé

Selon l’American Brain Tumor Association (ABTA), environ 700 000 personnes aux États-Unis vivent avec une tumeur primaire au cerveau et au système nerveux central. Bien que certaines tumeurs puissent être bénignes, beaucoup peuvent infiltrer les tissus cérébraux normaux et évoluer vers un cancer. Dans ce projet, nous visons à fournir des informations complètes sur les tumeurs cérébrales et à démontrer comment des images très précises et détaillées du cerveau humain peuvent être obtenues à l'aide de la technologie IRM. Nous discuterons ensuite des réseaux neuronaux convolutifs (CNN) et de leurs fonctionnalités. Après avoir obtenu un ensemble À partir de 3 000 images IRM et en appliquant un prétraitement, nous testerons ces images à l'aide d'un système intelligent que nous avons développé basé sur des réseaux neuronaux convolutifs (CNN) 2D pour détecter les images cérébrales saines par rapport à celles contenant des tumeurs. Ce système permettra une classification binaire en utilisant des méthodes d'extraction de caractéristiques, en particulier les CNN, qui nous ont donné d'excellents résultats, atteignant une précision de 99 % et une perte de 0,02 %. Enfin, nous comparerons nos résultats avec d'autres études axées sur des cas similaires pour améliorer la détection et l'analyse des tumeurs.

Mots-clés : tumeur cérébrale, réseau neuronal convolutif, IRM, cancer du cerveau, apprentissage automatique, imagerie médicale, système intelligent.

Abstract

According to the American Brain Tumor Association (ABTA), approximately 700,000 people in the United States are living with a primary brain and central nervous system tumor. While some tumors may be benign, many can infiltrate normal brain tissue and progress into cancer. In this project, we aim to provide comprehensive information about brain tumors and demonstrate how highly accurate and detailed images of the human brain can be obtained using MRI technology, Then we will discuss Convolutional Neural Networks (CNNs) and their functionality, After obtaining a set of 3000 MRI images and applying preprocessing, we will test these images using an intelligent system we developed based on 2D Convolutional Neural Networks (CNN) for detecting healthy brain images versus those containing tumors. This system will enable binary classification by utilizing feature extraction methods, specifically CNNs, which have given us excellent results, achieving 99% accuracy and a loss of 0.02%. Finally, we will compare our results with other studies focused on similar cases to enhance tumor detection and analysis.

Keywords: brain tumor, convolutional neural network, MRI, Brain cancer, machine learning, medical imaging, Intelligent system.

Contents

Acknowledgements ..... **I**

Dedication ..... **II**

Dedication ..... **III**

ملخص ..... **IV**

Résumé ..... **IV**

Abstract ..... **V**

Contents ..... **VI**

List of Figures ..... **IX**

List of Tables ..... **XI**

List of abbreviations ..... **XII**

General Introduction **1**

Chapter I : MRI & Brain Tumors Generalities **2**

I.1-Introduction 2

I.2-Problem Statement 2

I.3-Magnetic resonance imaging (MRI) 3

I.3.1-Principle of MRI 3

I.3.2-Use of MRI in medical imaging 4

I.4-Brain tumor disease 4

I.4.1-Classification of brain tumor 5

I.4.2-Brain Tumor Levels (Grades) 6

I.5-Architecture of diagnostic support systems brain tumor disease 7

I.5.1-Data Acquisition 7

I.5.2-Preprocessing 7

I.5.3-Resolution Adjustment 10

I.5.4-Segmentation Image 10

I.5.5-Feature Extraction 11

I.5.6-Classification and Diagnosis 12
I.6-Related Work 14
I.7-Conclusion 16
Chapter II: Artificial Intelligence & Neural Networks 17
II.1-Introduction 17
II.2-Artificial intelligence 17
II.3-Machine learning 18
II.3.1-Types of Machine Learning algorithms 18
II.3.2-Applications of Machine Learning techniques 19
II.4-Deep Learning 20
II.3.3-ML vs DL 21
II.5-ML/DL in medical field 23
II.6-Neural networks 23
II.6.1-Neural Networks History 24
II.6.2-Perceptron 25
II.6.3-Multilayer Perceptron (MLP) 26
II.7-Neural Networks Architectures 26
II.7.1-Convolutional Neural Network 26
II.7.2-CNN Layers 27
II.8-Activation Layers 31
II.9-Conclusion 34
Chapter III: Data Preparation & Experimental Results 35
III.1-Introduction 35
III.2-Software Material Used 35
III.2.1-Python 35
III.3-Libraries 35
III.3.1-Scikit-learn 35
III.3.2-Keras 36

III.3.3-TensorFlow 36
III.3.4-Matplotlib 37
III.4-Development environment 37
III.4.1-Anaconda 37
III.4.2-JupyterLab 38
III.5-Hardware used materials 38
III.6-Proposed System 39
III.6.1-Dataset Description 39
III.6.2-Image Preprocessing 41
III.6.3-Decomposition 41
III.6.4-CNN Model 42
III.7-Used Methods 45
III.7.1-Model Check Point callbacks 45
III.7.2-Compiler used 45
III.8-Results and Discussion 45
III.8.1-Accuracy and loss graph 46
III.8.2-Classification report 46
III.8.3-Confusion matrix 47
III.9-Final Test 48
III.9.1-Confusion matrix 48
III.9.2-Test Classification report 49
III.9.3-Detection of Our Model 49
III.10-Comparison with related work 50
III.11-Conclusion 51
General Conclusion 52
Bibliography 53

List of Figures

Figure I.1: An example of an MRI machine ----------5

Figure I.2: MRI images with different types of MRI contrasts (T1, T2 weighted, etc) ---------3

Figure I.3: Computed tomography (CT) scan ----------7

Figure I.4: MRI Image before and after denoising ----------7

Figure I.5: Multi-parametric MRI sequences before and after intensity normalization with 256 scale (8-bit depth). All normalized images have the same scale ----------9

Figure I.6: MRI image before and after Thresholding ----------10

Figure I.7: Image segmentation using multi-level thresholding ----------12

Figure II.1: ML Types ----------19

Figure II.2 : Supervised and Unsupervised Learning ----------21

Figure II.3: Comparison between ML and DL ----------23

Figure II.4: Neural networks history ----------25

Figure II.5: Architecture of a Perceptron ----------26

Figure II.6: Basic CNN Architecture ----------27

Figure II.7 : Convolution Operation ----------28

Figure II.8: Max and Average Pooling operations ----------30

Figure II.9 : Fully Connected Layer ----------31

Figure II.10 : Flatten layer operation ----------31

Figure II.11 : Rectified Linear Unit (ReLU) ----------32

Figure II.12 : ELU and RELU ----------33

Figure II.13 Sigmoid function ----------33

Figure II.14: Hyperbolic tangent function ----------34

Figure III.1: python ----------35

Figure III.2: Scikit-learn ----------36

Figure III.3: Keras ----------36

Figure III.4 TensorFlow ----------37

Figure III.5: matplotlib ----------37

Figure III.6: Anaconda ----------37

Figure III.7: Jupyter lab ----------38

Figure III.8: System block diagram ----------39

Figure III.9: samples of the dataset ----------40

Figure III.10: Dataset Classes ----------40

Figure III.11: Dataset distribution ----------40

Figure III.12: Split dataset ----------41

Figure III.13: Schematic diagram of CNN network training procedure ----------42

Figure III.14: Best Model Architecture. ----------44

Figure III.15: Accuracy and losses graph. ----------46

Figure III.16 : Validation Classification report ----------47

Figure III.17 : Confusion Matrix Validation ----------48

Figure III.18: Confusion Matrix of Test ----------49

Figure III.19: Classification report Test ----------49

Figure III.20: Detecting of some MRI images ----------50

List of Tables

Table I-1: Summary of some related works ----------15

Table II.1: Comparison between ML and DL ----------22

Table III.1: Data distribution of 3 classes ----------41

Table III.2: Best three proposed architectures ----------43

Table III.3: Best model results ----------48

Table III.4: Test results ----------48

Table III.5: Comparison with other studies ----------50

List of abbreviations

AI: Artificial Intelligence

ANN: Artificial Neural Network

CT: Computed Tomography

CNN: Convolutional Neural Network

DL: Deep Learning

DNN: Deep Neural Network

EG: Electroencephalography

Epoch: Epoch

FLAIR: Fluid-Attenuated Inversion Recovery

GPU: Graphics Processing Unit

KNN: K-Nearest Neighbors

LSTM: Long Short-Term Memory

LU: Rectified Linear Unit

ML: Machine Learning

MEG: Magnetoencephalography

MLP: Multi-Layer Perceptron

MRI: Magnetic Resonance Imaging

NN: Neural Network

RNN: Recurrent Neural Network

R-CNN: Region-Based Convolutional Neural Network

Re Adam: Adaptive Moment Estimation

RF: Random Forest

SGD: Stochastic Gradient Descent

SVM: Support Vector Machine

T1w: T1-Weighted MRI

T2w: T2-Weighted MRI

TPU: Tensor Processing Unit

General Introduction

Each year, about 87,240 people in the U.S. will receive a primary brain tumor diagnosis, Globally, an estimated 308,102 new cases of brain and nervous system tumors were diagnosed in 2020, according to the World Health Organization (WHO) ***(1) (2)***.

Brain tumors are complex and potentially life-threatening conditions that require accurate and timely diagnosis for effective treatment planning.

Currently, most detection and diagnosis methods rely on the decisions of neuro-specialists and radiologists for image evaluation, which can be prone to human error and are time-consuming, deep learning tools can help to minimize effort and speed up the detection process brain tumor detection using MRI images and deep learning, specifically Convolutional Neural Networks (CNNs), represents a significant advancement in medical imaging and healthcare. Brain tumors are complex and potentially life-threatening conditions that require accurate and timely diagnosis for effective treatment planning.

In The first chapter, we will dive in the medical description of brain tumor disease, detailing its various levels and types. Additionally, we examine the principles of magnetic resonance imaging (MRI) and how it aids in obtaining optimal images for our intelligent system.

In The second chapter explores the field of artificial intelligence, focusing on its sub-fields of Machine Learning and Deep Learning. It includes a description of the various types of deep learning and provides a brief history of neural networks. We will examine neural networks in general, specifically artificial neural networks (ANN) and convolutional neural networks (CNN). Additionally, the chapter details the different parameters of each layer within the architecture of CNNs.

In last chapter, we will discuss the programming environment, the language, and the libraries utilized to implement our solution. We will describe the database employed and the 2D CNN network developed for the classification and detection of brain tumors. Additionally, we will provide a detailed analysis of the obtained results. Finally, we will compare our findings with other studies on the same topic and test our model with MRI images.

Chapter I : MRI & Brain Tumors Generalities

I.1-Introduction

Medical imaging began in November 1895 with Wilhelm Conrad Roentgen's discovery of the X-ray. Working with an early cathode ray tube called a Crooke's tube, he noticed that the invisible rays were able to penetrate some solids (like human flesh) better than others (like bone or metal), and since that date, it has continued to develop, passing through the discovery of magnetic resonance imaging in 1970s by Raymond Damadian, the inventor of the first magnetic resonance scanning machine, performed the first full-body scan of a human being in 1977, While today MRI is one of the most powerful methods in the world to fight brain tumors.

In this chapter, we are interested in the medical description of brain tumor disease, including its various levels and types, as well as the principles of magnetic resonance imaging (MRI) and how it can help us in achieving the best images for our intelligent system.

I.2-Problem Statement

Most often, the cause of a brain tumor remains unknown. Symptoms such as headaches, seizures, and loss of consciousness may indicate its presence. Early detection plays a critical role in improving survival rates ***(3)***. Therefore, precise diagnosis and early detection of brain tumors are essential.

Detecting brain tumors using MRI poses several challenges and difficulties one of these difficulties is Similar Imaging Characteristics, some non-neoplastic conditions, such as vascular malformations, abscesses, or inflammation, may present with imaging features similar to brain tumors on MRI. This similarity can lead to misinterpretation and diagnostic errors.

Addressing these challenges often requires a multidisciplinary approach, including advanced imaging techniques, such as perfusion MRI, spectroscopy, and diffusion tensor imaging, as well as close collaboration between radiologists, neurosurgeons, and oncologists. Additionally, ongoing advancements in MRI technology and image analysis algorithms may help overcome some of these difficulties and improve the accuracy of brain tumor detection in the future.

In this work, we propose various approaches to feature extraction and fusion, providing a robust framework for distinguishing between unaffected individuals and those affected by brain tumors, also we will give an information on the detection of brain tumor with MRI and various image analysis algorithms.

I.3-Magnetic resonance imaging (MRI)

Magnetic resonance imaging (MRI) is a medical imaging technique that utilizes a magnetic field and computer-generated radio waves to generate detailed images of the organs and tissues within the body. Additionally, the MRI machine can produce three-dimensional images that are viewable from various angles as depicted in Figure I.1:



Figure I.1: An example of an MRI machine

I.3.1-Principle of MRI

Magnetic Resonance Imaging (MRI) is a medical imaging technique that utilizes strong magnetic fields and radio waves to generate detailed images of the internal structures of the body, including organs, tissues, and bones. The principle of MRI is based on the behavior of hydrogen atoms in the body when subjected to these magnetic fields and radio waves.

- **Hydrogen Atoms Alignment:** In MRI, the patient is placed inside a strong magnetic field produced by a large magnet. This magnetic field causes the hydrogen atoms in the body, particularly those in water molecules (which are abundant in biological tissues), to align themselves in the direction of the magnetic field.
- **Radiofrequency Pulse Excitation:** Once the hydrogen atoms are aligned, a radiofrequency (RF) pulse is applied, which temporarily disturbs the alignment of the hydrogen atoms. The RF pulse is emitted at a specific frequency that corresponds to the resonant frequency of the hydrogen nuclei.
- **Relaxation Processes:** After the RF pulse is turned off, the hydrogen atoms return to their original alignment with the magnetic field. During this process, known as relaxation, the hydrogen atoms emit energy in the form of radiofrequency signals. There are two types of relaxation processes:

- **T1 Relaxation (Longitudinal Relaxation):** In this process, the hydrogen atoms release energy as they realign with the magnetic field. The rate at which this relaxation occurs is characterized by the T1 relaxation time.
- **T2 Relaxation (Transverse Relaxation):** In this process, the hydrogen atoms lose phase coherence with each other, leading to the decay of the transverse magnetization. The rate of decay is characterized by the T2 relaxation time.
- **Signal Detection and Image Reconstruction:**
  The emitted radiofrequency signals are detected by receiver coils placed around the patient's body. These signals contain information about the spatial distribution of hydrogen atoms within the body, by manipulating the timing and strength of the magnetic fields and RF pulses, as well as employing various encoding techniques, MRI scanners can generate signals that are used to reconstruct detailed cross-sectional images of the body, Figure I.2 shows example of MRI Images of a brain.

MRI relies on the interaction between magnetic fields and hydrogen atoms to produce detailed images of the internal structures of the body, offering excellent soft tissue contrast and multiplanar imaging capabilities without using ionizing radiation ***(4)***. Figure I.2 shows example of MRI Images of a brain.

I.3.2-Use of MRI in medical imaging

Magnetic Resonance Imaging is a widely used medical imaging modality that provides detailed anatomical images of the human body without the use of ionizing radiation. By leveraging powerful magnets and radiofrequency pulses, MRI generates high-resolution images that can visualize soft tissues, organs, and structural abnormalities with exceptional clarity. Its versatility allows for imaging of various body parts, including the brain, spine, joints, and abdomen, making it invaluable in diagnosing a wide range of conditions, from neurological disorders to musculoskeletal injuries. Additionally, MRI is non-invasive and generally safe, making it suitable for patients of all ages. Its ability to offer multiplanar imaging and tissue characterization has revolutionized clinical diagnosis and treatment planning across medical specialties. As a result, MRI has become an indispensable tool in modern medicine, offering clinicians invaluable insights into patients' health and guiding personalized treatment strategies ***(5)***.

I.4-Brain tumor disease

A brain tumor is an abnormal growth of cells in the brain or surrounding tissues. These growths can be benign (non-cancerous) or malignant (cancerous). Benign tumors do not invade

nearby tissues or spread to other parts of the body, while malignant tumors can grow aggressively, invade surrounding tissue, and spread to other areas of the brain or spinal cord (processing known as metastasis) ***(6)***.

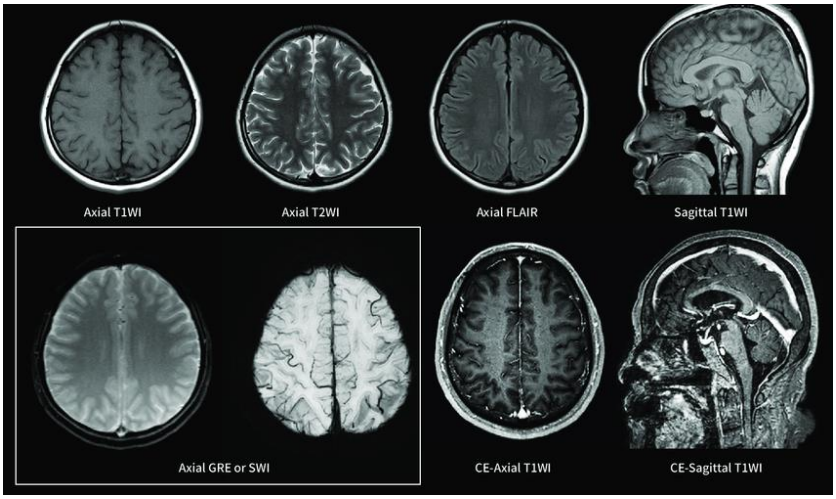


Figure I.2: MRI images with different types of MRI contrasts (T1-weighted, T2-weighted, etc.)

The reasons for most brain cancers are undiscovered ***(7)*** .However scientific research has led to the identification of some risk factors that might lead to the development of a brain tumor in ***(8) (9)***. Some factors that may increase the risk of developing a brain tumor include exposure to ionizing radiation, family history of brain tumors, certain genetic syndromes, and immune system disorders.

I.4.1-Classification of brain tumor

Brain tumors can be classified in several ways, including by their origin (primary or secondary), location within the brain, histological type (cell type), and grade (how aggressive they are). Here's a brief overview:

I.4.1.1-Origin

- **Primary Brain Tumors:** These tumors originate in the brain itself. They can be benign (non-cancerous) or malignant (cancerous).
- **Secondary Brain Tumors (Metastatic Brain Tumors):** These tumors originate elsewhere in the body (e.g., lung, breast, skin) and spread to the brain ***(10)***.

I.4.1.2-Location

Tumors can occur in different parts of the brain, such as the cerebrum, cerebellum, brainstem, or meninges (the layers covering the brain).

I.4.1.3-Histological Type

- **Gliomas:** These tumors arise from glial cells, which are supportive cells in the brain. Examples include astrocytoma, oligodendrogliomas, and ependymomas.
- **Meningiomas:** These tumors arise from the meninges, the layers of tissue that cover the brain and spinal cord.
- **Pituitary adenomas:** These tumors develop in the pituitary gland at the base of the brain.
- **Medulloblastomas:** These are fast-growing tumors that arise in the cerebellum, typically in children.
- **Schwannomas:** These tumors arise from Schwann cells, which produce the myelin sheath that covers nerves. They often occur on cranial nerves ***(11)***.
- **Craniopharyngiomas:** These rare tumors develop near the pituitary gland and can affect hormone levels and vision.
- **Hemangioblastomas:** These tumors are typically found in the cerebellum and are associated with von Hippel-Lindau disease ***(10)***

I.4.2-Brain Tumor Levels (Grades)

Brain tumors are graded based on their appearance under a microscope and their behavior. The grading system commonly used for brain tumors is called the World Health Organization (WHO) grading system, which categorizes tumors into different grades from I to IV:

- **Grade I (Low Grade):** Tumors are typically slow-growing and less likely to spread to other parts of the brain, often considered benign or less aggressive for an example Benign meningioma.
- **Grade II (Low to Intermediate Grade):** Tumors are typically slow-growing but may have more aggressive features compared to Grade I tumors, Can recur and may progress to higher grades over time like Diffuse astrocytoma.
- **Grade III (Intermediate Grade):** Tumors are more aggressive and have a higher likelihood of spreading to nearby tissues, may recur after treatment and have a worse prognosis compared to Grade I and II tumors, as Anaplastic oligodendroglioma.
- **Grade IV (High Grade):** Tumors are highly aggressive, fast-growing, and invasive, have a high risk of recurrence and spread (metastasis) to other parts of the brain, often considered malignant or cancerous, for example: Glioblastoma multiforme (GBM), the most common and aggressive type of brain tumor.

Understanding the type and grade of a brain tumor is essential for determining the appropriate treatment approach and predicting prognosis. Treatment options may include surgery, radiation therapy, chemotherapy, targeted therapy, and immunotherapy, depending on the tumor's characteristics and the individual's overall health ***(12)***.



Figure I.3: Computed tomography (CT) scan

I.5-Architecture of diagnostic support systems brain tumor disease

The architecture of a diagnostic support system for brain tumor disease typically encompasses several components designed to facilitate accurate diagnosis, treatment planning, and patient management. Here's an overview of the architecture ***(13)***.

I.5.1-Data Acquisition

The system begins with the acquisition of medical imaging data, primarily from MRI and Computed Tomography (CT) scans. These imaging modalities provide detailed information about the structure and morphology of brain tumors.

I.5.2-Preprocessing

Preprocessing in the context of medical imaging, particularly for brain tumor analysis, involves a series of steps aimed at enhancing the quality of images and preparing them for further analysis. Here's a detailed overview of preprocessing steps in Figure I.4:

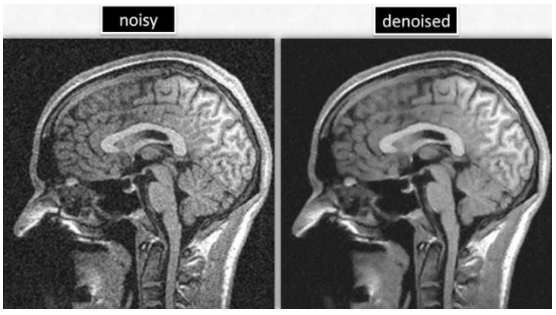


Figure I.4: MRI Image before and after denoising

I.5.2.1-Noise Reduction

Medical images, including MRI and CT scans, often contain noise introduced during the imaging process. Noise reduction techniques, such as Gaussian filtering or median filtering, are applied to enhance image clarity and remove unwanted artifacts ***(14)***.

- **Median filtering:** Is a popular image processing technique used for noise reduction and smoothing. It replaces each pixel value in an image with the median value of the neighboring pixels within a specified kernel or window size ***(15)***.

  Before filtering, the intensity values are: 46, 58, 47, 49, 41, 45, 42, 55, and 59. Arranging them in ascending order: 41, 42, 45, 46, 47, 49, 55, 58, 59. The median value is 47, which replaces the value 41 in the actual output of the median filter:

  a) Shows pixel intensity value before filtering.

  b) Shows pixel intensity value after filtering.

46	58	47
49	41	45
42	55	59

a

46	58	47
49	**47**	45
42	55	59

b

- **Gaussian filtering:** Gaussian filtering is a widely used image processing technique for smoothing or blurring an image. It is based on the Gaussian distribution, also known as the normal distribution, which describes the probability distribution of a continuous random variable ***(16)***.

I.5.2.2-Intensity Normalization

Several methods exist for intensity normalization, each with its advantages and considerations. Common approaches include linear scaling, histogram matching, z-score normalization, and statistical modeling. Linear scaling involves mapping intensity values to a predefined range, such as [0, 1] or [0, 255], through a linear transformation. Histogram matching aligns the intensity histograms of different images by adjusting intensity values to match a reference histogram. Z-score normalization standardizes intensity values to have a mean of 0 and a standard deviation of 1, facilitating comparisons across images. Statistical modeling techniques fit parametric distributions to intensity data, allowing for more flexible normalization based on the underlying intensity distribution ***(17)***.

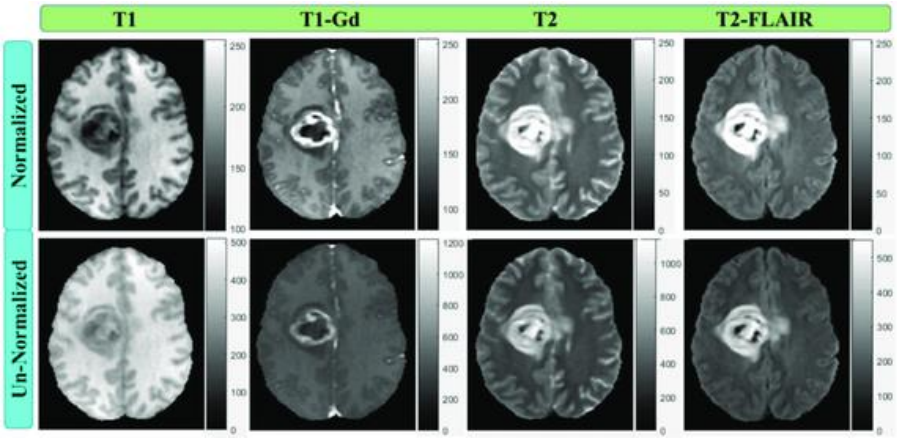


Figure I.5: Multi-parametric MRI sequences before and after intensity normalization with 256 scale (8-bit depth). All normalized images have the same scale.

I.5.2.3-Geometric Correction

Geometric distortions may occur in medical images due to factors such as patient motion or magnetic field inhomogeneities. Geometric correction techniques, including image registration and distortion correction algorithms, are applied to align images and correct for spatial distortions ***(18)***

I.5.2.4-Bias Field Correction

MRI images may exhibit intensity variations known as bias fields, caused by imperfections in the imaging process. Bias field correction methods are used to remove these intensity variations and improve the accuracy of subsequent analysis steps ***(19)***.

I.5.2.5-Image Enhancement

Image enhancement techniques, such as contrast stretching, histogram equalization, or adaptive filtering, may be applied to improve the visual quality of images and enhance relevant features for better tumor visualization ***(20)***.

I.5.2.6-Artifact Removal

Various artifacts, such as motion artifacts, susceptibility artifacts, or RF coil artifacts, may be present in medical images and can affect image quality and interpretation. Artifact removal algorithms are used to identify and suppress these artifacts, ensuring accurate analysis results ***(21)***.

I.5.3-Resolution Adjustment

In some cases, it may be necessary to adjust the spatial resolution of images to match the requirements of specific analysis tasks or computational algorithms. Up sampling or down sampling techniques may be applied to achieve the desired resolution ***(22)***.

I.5.4-Segmentation Image

Image segmentation is a critical step in medical image analysis, including the detection and characterization of brain tumors. It involves partitioning an image into multiple segments or regions to extract meaningful information and identify structures of interest ***(23)***

I.5.4.1-Thresholding

Thresholding is a simple and commonly used technique for image segmentation, can be expressed as:

$$T = T[x, y, p(x, y), f(x, y]$$ Eq I.1

The threshold value, denoted as T, is determined based on the pixel coordinates (x, y) of a specific point within the image. This point is represented by $p(x, y)$ and corresponds to a certain gray level in the image, denoted as $f(x, y)$. The threshold image, $g(x, y)$, is then defined based on this threshold value:

$$g(x,y) = \begin{cases} 1 & if\ f(x,y) > 1 \\ 0 & iff(x,y) \leq 0 \end{cases}$$ Eq I.2

Thresholding techniques are classified to into two Global Thresholding and Local Thresholding ***(24)*** ,It involves selecting a threshold value and classifying pixels in the image.

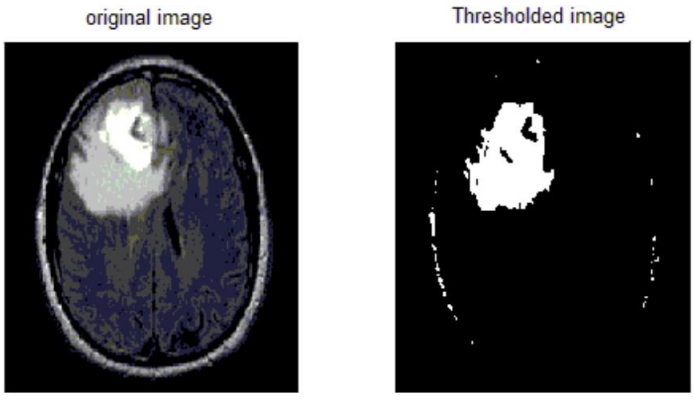


Figure I.6: MRI image before and after Thresholding

As either foreground or background based on their intensity values. Pixels with intensity values above the threshold are assigned to one class (foreground), while pixels with intensity values below the threshold are assigned to another class (background).

I.5.4.2-Region Growing

Region growing is a region-based segmentation method that starts from a seed point and iteratively expands the region by adding neighboring pixels that satisfy certain criteria, such as similarity in intensity or texture. This method is particularly useful for segmenting homogeneous regions within the tumor. ***(25)***.

I.5.4.3-Machine Learning-Based Approaches

Machine learning-based approaches for image segmentation involve training models to automatically learn the segmentation task from annotated training data. Supervised learning methods, such as convolutional neural networks (CNNs), are commonly used for this purpose, CNNs learn hierarchical representations of image features through multiple layers of convolutional and pooling operations, enabling them to capture complex patterns and structures in the data, Semantic segmentation networks, such as U-Net, FCN (Fully Convolutional Network), and SegNet, are specifically designed for pixel-level segmentation tasks, These networks are trained on labeled image datasets, where each pixel is assigned a class label corresponding to the segmented object or region ***(26)***.

I.5.5-Feature Extraction

Extracting relevant features from segmented tumor regions is crucial for characterizing tumor properties. Feature extraction methods capture quantitative information about tumor morphology, texture, intensity, and spatial relationships. These features serve as input for subsequent analysis and classification ***(27)***.

I.5.5.1-Super Pixel based Feature Extraction

Super pixel algorithms, such as SLIC (Simple Linear Iterative Clustering), partition an image into perceptually meaningful and compact regions.

The SLIC algorithm reduces intensity redundancy in an image by partitioning it into a convenient number, K, of clusters based on spatial distance for more details visit [8].

$$d_c = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$

$$d_s = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \qquad \text{Eq I.3}$$

by using a 5D Euclidean distance measure,

$$D = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_c}{S}\right)^2} \quad \text{Eq I.4}$$

S represents the sampling interval and is a fixed constant 'm' that serves to mitigate significant variations in color distance.

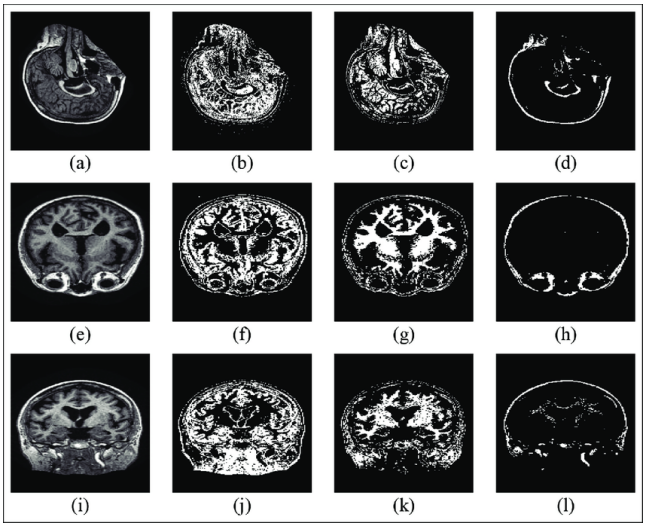


Figure I.7: Image segmentation using multi-level thresholding.

I.5.5.3-Texture feature extraction

Texture feature extraction focuses on capturing spatial patterns and variations in pixel intensities within an image, Texture features describe the arrangement of pixel values and their relationships, which can convey information about surface roughness, coarseness, or regularity within an image region, Various texture feature extraction methods exist, including statistical measures such as mean, standard deviation, and entropy, as well as more advanced techniques like co-occurrence matrices (GLCM), local binary patterns (LBP), and Gabor filters, more info in ***(28)***.

I.5.6-Classification and Diagnosis

Machine learning algorithms, including supervised learning, unsupervised learning, and deep learning, are employed for tumor classification and diagnosis. These algorithms analyze extracted features and classify tumors into different categories based on their characteristics, such as tumor type, grade, and prognosis ***(27).***

I.5.6.1-Supervised Techniques

- **Machine Learning Algorithms:** Supervised learning methods, such as support vector machines (SVM), random forests, and convolutional neural networks (CNNs), are commonly used for detecting and segmenting brain abnormalities ***(29)***. These algorithms are trained on labeled datasets, where medical images are annotated by experts to indicate the presence and location of abnormalities.
- **Feature-Based Methods:** Supervised techniques often rely on handcrafted features extracted from medical images, including intensity, texture, or shape features. These features are used as input to machine learning classifiers, aiding in the discrimination between abnormal and normal brain tissue.
- **Semantic Segmentation:** CNNs have demonstrated significant success in semantic segmentation tasks by directly predicting pixel-wise labels for medical images. These deep learning models learn hierarchical features from raw image data, enabling accurate segmentation of complex structures and abnormalities ***(30)***.

I.5.6.2-Unsupervised Techniques

- **Clustering Algorithms:** Unsupervised learning methods, such as k-means clustering or Gaussian mixture models, can identify clusters of similar intensity or spatial patterns within medical images. By grouping voxels or pixels based on similarity measures, these algorithms can reveal potential regions of abnormality without prior knowledge or labeled data ***(31)***
- **Atlas-Based Methods:** Atlas-based approaches involve registering a template or atlas image to the patient's image space and propagating labeled information from the atlas to the target image. This method leverages spatial correspondence to transfer knowledge from a labeled atlas to the patient's image, aiding in abnormality detection and segmentation.
- **Generative Models:** Generative models, like variational autoencoders (VAEs) or generative adversarial networks (GANs), learn the underlying distribution of normal brain anatomy from unlabeled data. By identifying deviations from this learned distribution, these models can highlight potentially abnormal regions in medical images ***(32)***.

I.5.6.3-Decision Support and Visualization

The system provides decision support tools to assist clinicians in interpreting diagnostic results and making informed treatment decisions. Visualization techniques, such as 3D

rendering, volume rendering, and cross-sectional views, facilitate the visualization of tumor morphology and spatial relationships ***(33)***.

I.6-Related Work

Convolutional Neural Networks (CNNs) excel in processing and analyzing visual data, offering several advantages. Their key strength lies in their ability to automatically learn hierarchical patterns and features from images through convolutional layers, enabling them to capture spatial dependencies. CNNs are adept at feature extraction and can adaptively learn representations, reducing the need for manual feature engineering. They also incorporate parameter sharing and pooling layers, which enhance efficiency and robustness by reducing the number of parameters and capturing translation invariance. With their capacity to handle large-scale datasets and parallel processing, CNNs have become the cornerstone of numerous computer vision tasks, including image classification, object detection, and segmentation.

In ***(34)*** , a convolutional neural network designed for brain tumor detection was introduced by the authors. They developed two CNN models for comparison to identify the optimal classification model. The initial model comprises a single convolutional layer, while the second integrates two layers. The findings revealed that augmenting the convolutional layers enhances model performance, yielding an accuracy of 93% and a loss value of 0.23.

Transfer learning also is a powerful technique in artificial neural networks where knowledge gained from training one model on a specific task is transferred and applied to another related task. It's particularly useful when you have a small dataset for the target task but a larger dataset for a related task, Transfer learning can save significant time and computational resources, especially in domains where labeled data is scarce or expensive to acquire. It's a valuable technique for leveraging the knowledge encoded in pre-trained models and applying it to new tasks.

A study published in ***(35)*** aimed to enhance accuracy through the utilization of transfer learning strategies. They employed three transfer learning approaches utilizing pre-trained CNN models: VGG19, Inception V3, and MobileNet V2. The achieved accuracy rates were 88.22%, 91%, and 92%, respectively. The authors concluded that MobileNet V2 exhibited the highest efficiency compared to the other models.

Utilizing the Artificial Neural Network (ANN) methodology, researchers in ***(36)*** focused on discerning between benign and malignant tumors. Initially, they conducted image preprocessing using filters, followed by implementing the average color moment technique to

extract key features from the images. Subsequently, these characteristic maps were fed into the ANN for classification purposes, achieving an impressive accuracy rate of 91.8%

In a study featured in ***(37)*** , a model leveraging the histogram statistical equalization technique was explored. This method involves applying a transformation to individual pixels in an image, utilizing various statistical metrics like average sum, variance, entropy, and dissimilarity. The model was specifically tailored for analyzing both low-grade and high-grade class images of cervical glioma. Impressively, the results of this approach yielded promising outcomes, with accuracy reaching 83.6%, sensitivity at 80.88%, and specificity reaching 86.84%.

Table I-1: Summary of some related works

Study	Dataset	Processing technique	Classifier	Result
(34)	Kaggel dataset	Data Augmentation	CNN	Ac = 93% Loss = 0.23
(35)	Brain Tumor Dataset	Transfer learning	VGG19	Ac=88% F1score=88.18%
			Inception V3	Ac=91% F1score=90.98%
			MobileNetV2	Ac=88% F1score=88.18%
(36)	Harvard Medical School Website Dataset	Noise Filtering Average color Moment	ANN	Ac=91.8%
(37)	LGG Flair MRI images	ROI feature extraction	Random forest	Ac=83.6% Sensitivity=80.88% Specificity=86.84%

I.7-Conclusion

Brain tumor disease is a dangerous disease that destroys the cells of the brain. Through this chapter, we have explained this disease levels and types, the principles of acquisition and formation of MRI images, and how we can detect and analyze the disease using different types of preprocessing techniques to get the best image from MRI machine, as well as a summary of the related work emphasizing the massive use of deep learning today, without abandoning traditional systems which always offer satisfactory performance. The objective of this work is to take advantage of these two approaches by proposing a fusion of hybrid characteristics in order to implement a system to aid in the diagnosis of brain tumor disease, The following chapter will therefore present convolutional neural networks, as well as the pre-trained architectures studied.

Chapter II: Artificial Intelligence & Neural Networks

II.1-Introduction

Artificial Intelligence and neural networks offer numerous benefits that are transforming various industries and aspects of daily life. Neural networks, a core component of AI, excel in pattern recognition and learning from vast amounts of data.

This chapter, consists of discovering the field of artificial intelligence and its sub-fields such as Machine Learning and Deep Learning with description of the different types of deep learning. Also a also a brief history about neural networks then we illustrate: neural networks in general (artificial neural networks and CNN (convolutional neural networks) while detailing the different parameters of each layer in the architecture of CNN networks.

II.2-Artificial intelligence

Artificial Intelligence (AI) has become one of the most transformative technologies of our time, revolutionizing industries ranging from healthcare to finance, and from transportation to entertainment. At its core, AI involves the development of computer systems that can perform tasks that typically require human intelligence. These tasks include but are not limited to visual perception, speech recognition, decision-making, and language translation ***(38)***.

One of the key aspects of AI is machine learning, a subset of AI that enables computers to learn from data without being explicitly programmed. Machine learning algorithms are trained on large datasets, allowing them to identify patterns and make predictions or decisions based on new data. Deep learning, a type of machine learning inspired by the structure and function of the human brain's neural networks, has particularly driven breakthroughs in tasks such as image recognition and natural language processing.

In recent years, AI has made significant strides in various domains:

- **Healthcare:** AI is being used to analyze medical images, assist in diagnosis, and personalize treatment plans. For example, AI algorithms have demonstrated remarkable accuracy in detecting certain diseases from medical scans, sometimes outperforming human experts ***(39)***.
- **Finance:** AI is utilized for fraud detection, algorithmic trading, and risk management. AI-powered chatbots are also becoming increasingly common in customer service, providing personalized assistance and support around the clock.
- **Automotive industry:** is another area where AI is having a profound impact. Self-driving cars rely heavily on AI algorithms to perceive their surroundings, make decisions in real-time, and navigate safely through complex environments. Companies

like Tesla, Waymo, and Uber are leading the way in developing autonomous vehicle technology.

Ethical considerations surrounding AI, including issues of bias, transparency, and accountability, have also garnered significant attention. AI systems are only as good as the data they are trained on, and biased datasets can lead to biased outcomes, perpetuating existing inequalities. Therefore, it is crucial for AI developers to prioritize fairness and transparency throughout the development process ***(40)***

Despite the tremendous progress made in AI, there are still many challenges and opportunities on the horizon. Research areas of explainable AI, making AI systems more understandable and interpretable to humans, and AI safety, ensuring that AI systems behave in a predictable and beneficial manner, is ongoing ***(41)***

II.3-Machine learning

Machine Learning (ML) is a subfield of AI that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. Instead, these algorithms learn from data, identifying patterns and making decisions or predictions based on that data. ML has become increasingly important across various industries, from healthcare and finance to marketing and entertainment ***(42)***.

II.3.1-Types of Machine Learning algorithms

There are several types of ML algorithms, including:

- **Supervised Learning:** In supervised learning, the algorithm is trained on a labeled dataset, where each example is paired with a corresponding label or outcome. The goal is for the algorithm to learn a mapping from input to output based on the examples provided. Common supervised learning algorithms include linear regression, decision trees, and neural networks ***(43)***.
- **Unsupervised Learning:** Unsupervised learning involves training algorithms on unlabeled data, where the goal is to uncover hidden patterns or structures within the data. Clustering algorithms, such as k-means clustering and hierarchical clustering, are examples of unsupervised learning techniques ***(43)***.
- **Semi-supervised Learning:** Semi-supervised learning combines elements of both supervised and unsupervised learning. It involves training algorithms on a dataset that contains both labeled and unlabeled examples, leveraging the labeled data where available while also seeking to generalize from the unlabeled data ***(44)***.

- **Reinforcement Learning:** Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties based on its actions, and the goal is to learn a policy that maximizes the cumulative reward over time. Reinforcement learning has been successfully applied in domains such as robotics, game playing, and autonomous vehicle control ***(45)***.

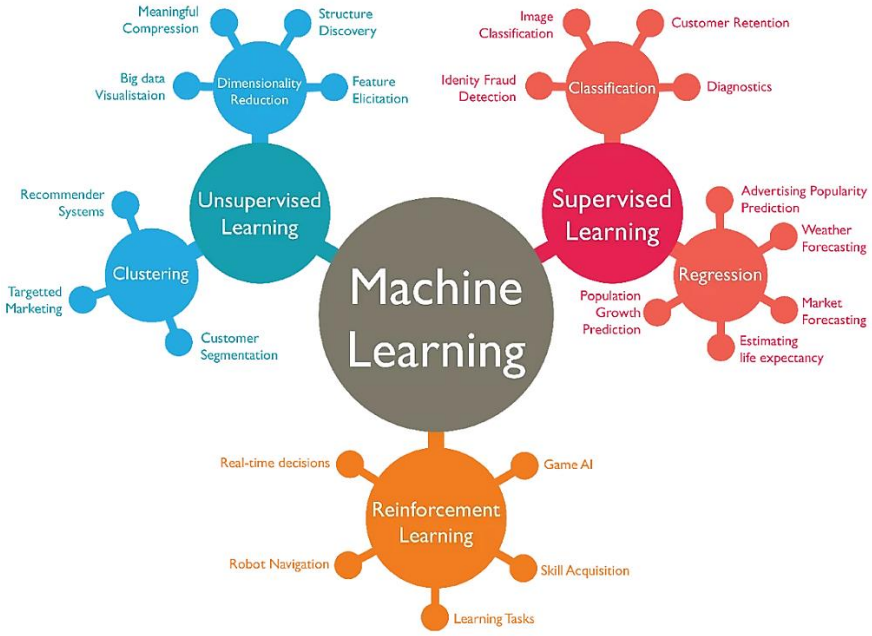


Figure II.1: ML Types

II.3.2-Applications of Machine Learning techniques

ML techniques have been applied to a wide range of applications, including:

- **Natural Language Processing (NLP):** NLP involves tasks such as text classification, sentiment analysis, machine translation, and question answering. Techniques like recurrent neural networks (RNNs) and transformers have been particularly successful in NLP tasks.
- **Computer Vision:** Computer vision focuses on enabling computers to interpret and understand visual information from the real world. Convolutional neural networks (CNNs) have revolutionized computer vision tasks such as image classification, object detection, and image segmentation.
- **Healthcare:** Machine learning is being used in healthcare for tasks such as disease diagnosis, personalized treatment recommendation, and medical image analysis. ML

models trained on large healthcare datasets have shown promising results in improving diagnostic accuracy and patient outcomes.

- **Finance:** In finance, machine learning algorithms are used for tasks such as stock price prediction, fraud detection, algorithmic trading, and credit risk assessment. These algorithms analyze vast amounts of financial data to identify patterns and make predictions.
- **Recommendation Systems:** Recommendation systems use machine learning to analyze user preferences and behavior in order to recommend relevant items or content. These systems are widely used in e-commerce platforms, streaming services, and social media websites to personalize user experiences ***(46)***.

II.4-Deep Learning

Deep learning (DL) is a subset of ML that focuses on the development and training of neural networks, which are algorithms inspired by the structure and function of the human brain. These neural networks consist of interconnected layers of nodes, or artificial neurons, that process and transform input data to produce output. DL algorithms are capable of automatically learning representations of data through the use of multiple layers of abstraction.

One of the fundamental aspects of DL is its ability to learn hierarchical representations of data. Each layer in a deep neural network learns to represent features at different levels of abstraction. For example, in an image recognition task, lower layers might learn to detect simple features like edges and corners, while higher layers might learn to recognize more complex patterns like shapes or objects.

Deep learning has achieved significant success in various fields, including computer vision, natural language processing, speech recognition, and reinforcement learning. Some notable applications include image and video recognition, language translation, medical image analysis, autonomous driving, and game playing ***(38)***.

- **Supervised Learning:** In supervised learning, the model is trained on a labeled dataset, meaning that each input is associated with a corresponding target output, during training, the model learns the mapping from inputs to outputs by minimizing the difference between its predictions and the true labels, Examples include image classification, speech recognition, and language translation ***(47)***.
- **Unsupervised Learning:** Unsupervised learning (Figure II.3) involves training models on unlabeled data, where the algorithm tries to find patterns or representations in the

data without explicit guidance, Clustering, dimensionality reduction, and generative modeling are common tasks in unsupervised learning, The model learns to represent the underlying structure of the data without being explicitly told the correct output ***(48)***.

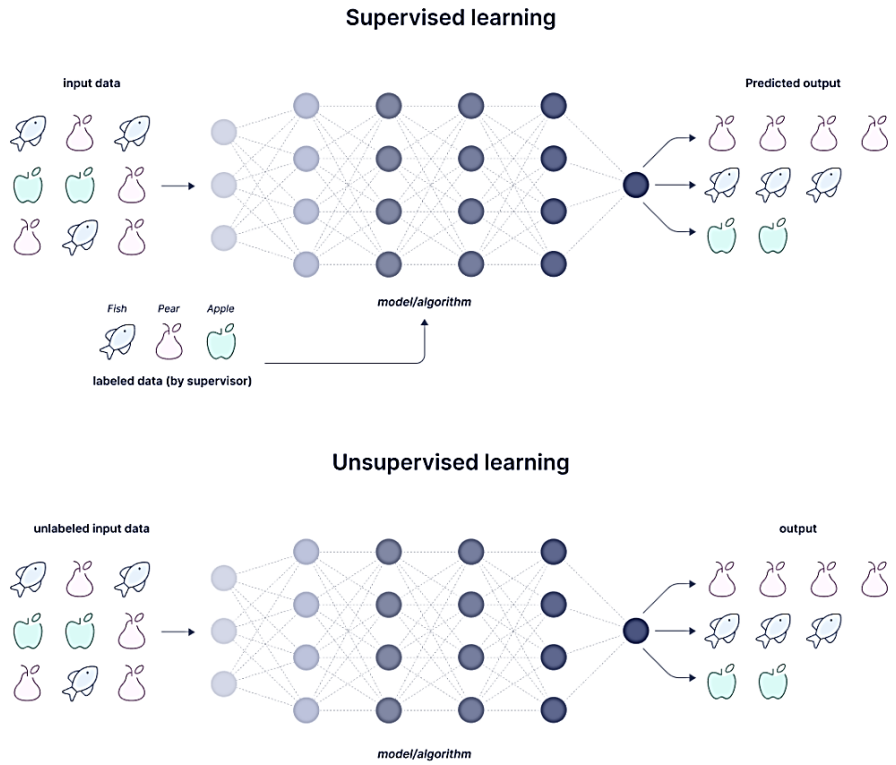


Figure II.3: Supervised and Unsupervised Learning ***(49)***

- **Semi-supervised Learning:** Semi-supervised learning combines elements of both supervised and unsupervised learning. It leverages a small amount of labeled data along with a large amount of unlabeled data to improve the model's performance, By exploiting the structure or relationships within the unlabeled data, semi-supervised learning can generalize better to unseen examples, Techniques such as self-training, co-training, and pseudo-labeling are commonly used in semi-supervised learning ***(50)***.

II.3.3-ML vs DL

ML and DL are both subsets of AI, enabling machines to learn from data and make decisions or predictions to require like human:

Table II.1: Comparison between ML and DL

	Approach	Architecture	Training	Applications	Models
DL	Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement learning	Multiple layers of interconnected neurons.	uses artificial neural networks to mimic the learning process of the human brain	Computer vision, NLP, Speech recognition, and more...	CNN, RNN, LSTM
ML	Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement learning	simple algorithm for tasks.	can automatically adapt with minimal human interference	Healthcare, Finance, Marketing, Classification, Regression, Clustering, Anomaly detection.	Decision Trees, Random Forests, SVM, K-Nearest Neighbors (KNN), Linear Regression, Logistic Regression, etc.

II.3.3.1-Key Differences

- **Representation:** While both ML and DL learn from data but DL focuses on learning hierarchical representations of data through multiple layers of neural networks.
- **Feature Engineering:** In traditional ML, feature engineering is often crucial, where DL models can automatically learn relevant features from raw data.
- **Performance:** DL models often outperform traditional ML algorithms, especially in tasks involving unstructured data like images, text, and speech.
- **Data Requirements:** DL models generally require more data and computational resources for training compared to traditional ML algorithms ***(51)***.

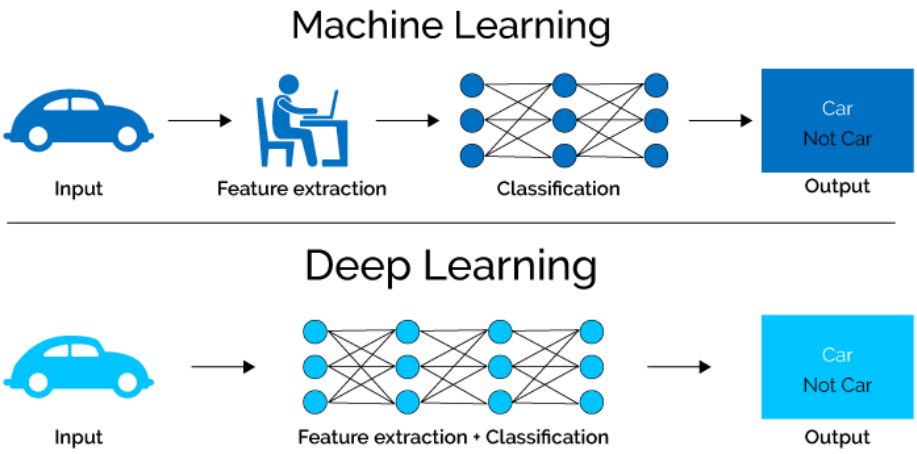


Figure II.2: Comparison between ML and DL

II.5-ML/DL in medical field

ML in the medical field has shown promising applications across various domains, from disease diagnosis to personalized treatment plans. Here are some key areas where ML is making an impact:

- **Medical Image Analysis:** ML algorithms, especially DL models, have demonstrated remarkable performance in tasks such as image segmentation, lesion detection, and classification in medical imaging modalities like MRI, CT, and X-ray ***(52)***.
- **Disease Diagnosis and Prognosis:** ML models can assist in diagnosing diseases by analyzing patient data, including medical records, genetic information, and diagnostic tests, to identify patterns indicative of various conditions ***(53)***.
- **Personalized Medicine:** ML algorithms analyze patient-specific data to tailor treatment plans and predict individual responses to therapies, leading to more personalized and effective healthcare interventions ***(54)***.
- **Healthcare Operations and Management:** ML models optimize hospital operations, resource allocation, patient scheduling, and disease outbreak prediction, enhancing the efficiency of healthcare delivery systems ***(55)***.

II.6-Neural networks

Neural networks are computational models inspired by the human brain's structure and function, forming the backbone of many modern machine learning and artificial intelligence applications. Comprising interconnected nodes, or "neurons," these networks are organized into layers: an input layer, one or more hidden layers, and an output layer. Each neuron processes input data, applies a weight, adds a bias, and then passes the result through an activation function, which introduces non-linearity into the model, enabling it to learn complex patterns.

Training a neural network involves adjusting these weights and biases to minimize the error between the predicted and actual outputs, typically using algorithms such as backpropagation and gradient descent. Neural networks excel in tasks like image and speech recognition, natural language processing, and autonomous systems due to their ability to model intricate relationships within large datasets. Their development has significantly advanced fields such as computer vision and robotics, making neural networks fundamental to the progress of artificial intelligence ***(56)***.

II.6.1-Neural Networks History

Neural networks have undergone significant evolution since their inception in the 1940s. Here's a brief overview of their development over time :

- **1940s - 1950s:** Perceptrons and Early Models: The concept of artificial neurons and perceptrons was introduced by Warren McCulloch and Walter Pitts in 1943. These early models laid the foundation for neural network research, Frank Rosenblatt developed the perceptron algorithm in 1957, which could learn simple binary classifiers [29] [30].
- **1960s - 1970s:** Limitations and Disinterest: Neural network research faced limitations due to the perceptron's inability to solve non-linear problems, leading to a decline in interest, Marvin Minsky and Seymour Papert's book "Perceptrons" (1969) highlighted the limitations of single-layer perceptrons, dampening enthusiasm for neural networks.
- **1980s:** Resurgence with Backpropagation: The development of backpropagation by Paul Werbos in 1974 and its popularization by Rumelhart, Hinton, and Williams in 1986 revolutionized neural network training, multi-layer perceptrons (MLPs) gained popularity due to their ability to solve complex problems through deep architectures [31].
- **1990s - Early 2000s:** Rise and Fall of Neural Networks using widely in various fields, including finance, medicine, and engineering. However, limitations such as vanishing gradients and computational constraints hindered further progress, leading to a decline in interest in neural networks in favor of other machine learning techniques like SVMs and decision trees ***(57)***.
- **Late 2000s - Early 2010s:** Deep Learning Renaissance advanced in hardware, such as GPUs, and the availability of large datasets facilitated the training of deep neural networks. Breakthroughs in deep learning architectures, such as convolutional neural networks (CNNs) for image recognition and recurrent neural networks (RNNs) for sequential data, revolutionized various domains, including computer vision, natural language processing, and speech recognition ***(58)***

- **2010s-Present:** Continued advancements and applications of DL continues to advance, with improvements in model architectures (e.g., attention mechanisms, transformers), training techniques (e.g., transfer learning, self-supervised learning), and optimization algorithms (e.g., Adam, SGD variants). Applications of neural networks span across industries, including autonomous vehicles, healthcare, finance, and more ***(59)***

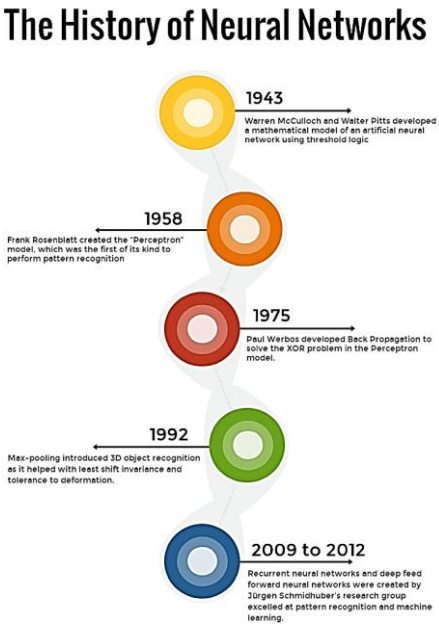


Figure II.4: Neural networks history

II.6.2-Perceptron

The perceptron is one of the simplest forms of artificial neural networks, introduced by Frank Rosenblatt in 1957. It serves as the building block for more complex neural network architectures.

A perceptron is a single-layer neural network with binary output. It takes multiple binary inputs, computes a weighted sum of these inputs, and then applies a step function to produce a binary output, the perceptron takes multiple binary inputs $x_1, x_2, \ldots, x_n$, Each input is associated with a weight, $w_1, w_2, \ldots, w_n$, which determines the importance of that input ***(60)***.

The weighted sum of inputs and weights is calculated:

$$Z = \sum_{i=1}^{n} (x_i \times w_i) + b \qquad \text{Eq II.1}$$

where b is the bias term.

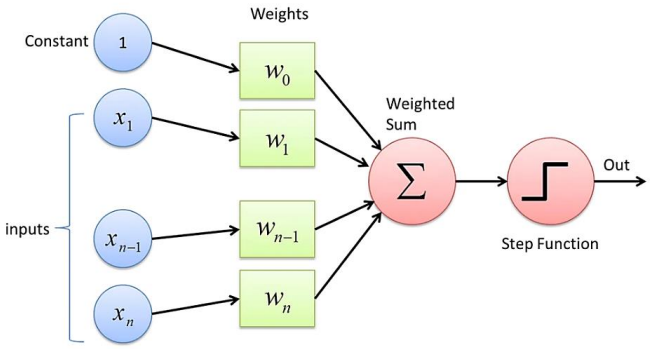


Figure II.5: Architecture of a Perceptron

II.6.3-Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is a type of feedforward artificial neural network with multiple layers of neurons, consisting of an input layer, one or more hidden layers, and an output layer. Each neuron in one layer is connected to every neuron in the subsequent layer. MLPs are one of the most common and widely used types of neural networks, particularly in supervised learning tasks like classification and regression.

- **Forward Propagation:** Each neuron in the network computes a weighted sum of its inputs, applies an activation function, and passes the result to the neurons in the next layer. Mathematically, this can be represented as :

$$z^{(l)} = W^{(l)}.a^{(l-1)} + b^{(l)} \quad \text{Eq II.2}$$

where:

$$a^{(l)} = g(z^{(l)}) \quad \text{Eq II.3}$$

$z^{(l)}$ is the weighted sum of inputs for layer l.

$a^{(l)}$ is the activation of layer l.

$W^{(l)}$ is the weight matrix for layer l.

$b^{(l)}$ is the bias vector for layer l.

$g(.)$ is the activation function.

II.7-Neural Networks Architectures

II.7.1-Convolutional Neural Network

Convolutional neural networks (CNNs) are derived from multilayer perceptron, engineered to mimic the functionality of a visual cortex. Unlike traditional MLPs, these architectures address the complexities by capitalizing on the inherent spatial correlations found in real-world images.

CNNs draw inspiration from the visual processing system of animals, particularly the receptive fields found in the visual cortex. In the 1950s and 1960s, researchers such as Hubel and Wiesel conducted groundbreaking work on the structure and function of the visual cortex, which laid the foundation for the development of CNNs ***(61)***.

CNNs are a class of deep neural networks specifically designed for processing structured grid-like data, such as images, CNNs have become the backbone of many computer vision tasks, including image classification, object detection, segmentation, and more.

Each input image undergoes processing through two blocks of layers. The first block comprises convolutional layers, responsible for generating features. Following this, there's a second classification block containing fully connected layers with an architecture similar to MLPs. (Figure II.6).

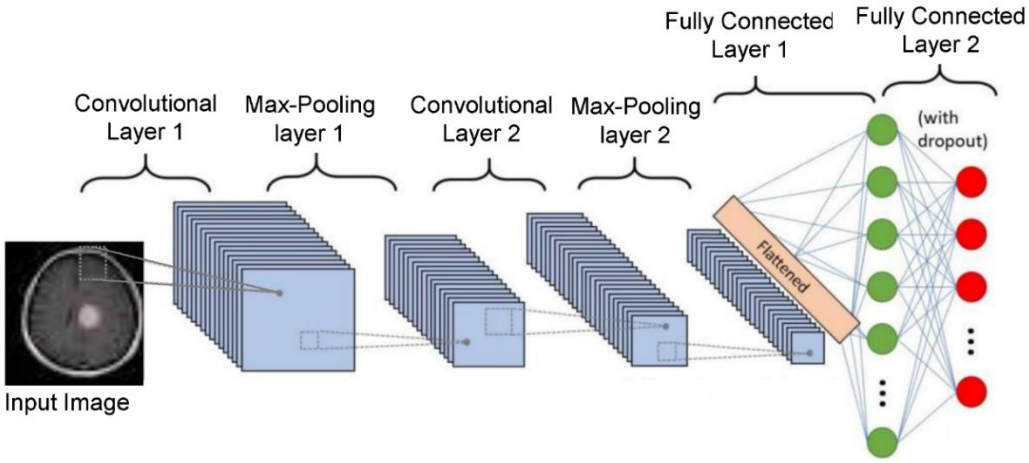


Figure II.6: Basic CNN Architecture

II.7.2-CNN Layers

- **Convolution:** The convolutional layer is a fundamental building block of CNN and plays a crucial role in extracting features from input data, such as images. Understanding the mathematics behind the convolutional layer requires grasping concepts related to convolution operations and parameter sharing. it is an operation that expresses the relationship between two functions as a third function. In the context of CNN, the convolution operation is applied between the input data (often an image represented as a matrix) and a learnable filter (also known as a kernel or weight matrix).

  The convolution operation involves sliding the filter over the input data and computing element-wise multiplications followed by summation at each position. This produces a feature map, which highlights different features or patterns present in the input data.

- **Filter Size:** The size of the filter/kernel determines the spatial extent of the local connectivity between input and output neurons. Common filter sizes are 3x3, 5x5, or 7x7.
- **Stride:** The stride specifies how much the filter shifts (or strides) when moving across the input data during convolution. A stride of 1 means the filter moves one pixel at a time, while a larger stride skips pixels.
- **Padding:** Padding refers to adding additional border pixels to the input data, which helps preserve spatial dimensions in the output feature map. Common padding types include 'valid' (no padding) and 'same' (pad to ensure output size matches input size).
- **Number of Filters:** Each convolutional layer typically consists of multiple filters, each responsible for detecting different features in the input data. The number of filters determines the depth (or number of channels) of the output feature map.

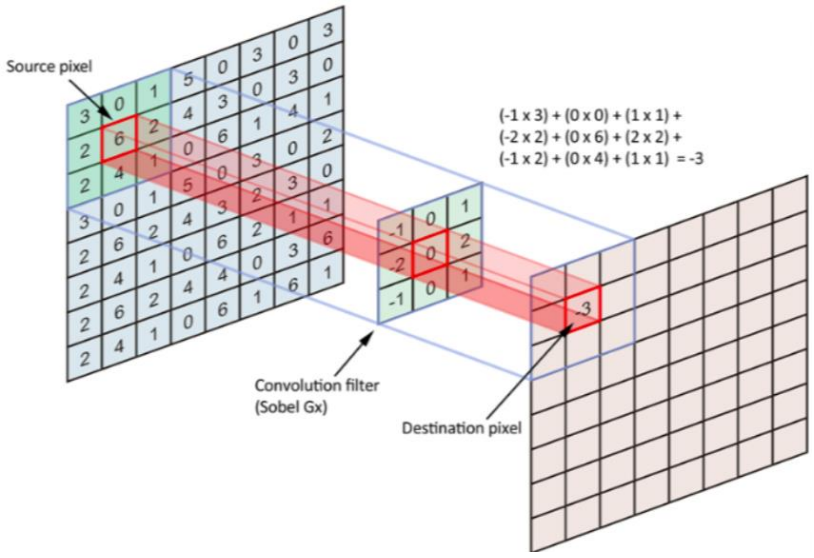


Figure II.7 : Convolution Operation

- **Pooling:** Pooling in Convolutional Neural Networks (CNNs) is a technique used to down-sample the spatial dimensions (width and height) of feature maps generated by convolutional layers while retaining important information. Pooling helps reduce the computational complexity of the network and makes the learned representations more robust to variations in input.
  - **Max Pooling:** Max pooling is a technique used in convolutional neural networks (CNNs) for reducing the spatial dimensions (width and height) of the feature maps while retaining the most important information. It is typically applied after

convolutional layers in CNN architectures. Max pooling divides the input feature map into non-overlapping rectangular regions (often referred to as pooling regions or pooling windows). For each region, it retains only the maximum value.

Similar to convolutional layers, max pooling also has a parameter called stride, which determines the step size at which the pooling regions are moved across the input feature map. A stride of 2, for example, means that the pooling regions are moved by 2 units horizontally and vertically.

- ❖ **Size of Pooling Window:** The size of the pooling window (usually denoted by $n \times n$ is another parameter that determines the spatial extent of the pooling operation. Common choices for the size of the pooling window include 2×2 or 3×3.
- ❖ **Dimension Reduction:** By selecting only the maximum value within each pooling region, max pooling effectively reduces the spatial dimensions of the feature maps. For example, applying 2×2 max pooling with a stride of 2 reduces the width and height of the feature maps by half.

- **Average Pooling:** On the other hand, In average pooling, the input region covered by each pooling window is replaced by the average value of the elements within that region.

  After applying convolutional layers to extract features from the input data, CNNs often include pooling layers to reduce the spatial dimensions of the feature maps while retaining important information. Pooling helps in controlling overfitting, reducing computational complexity, and increasing the receptive field.

  - ❖ **Pooling Windows:** Average pooling divides the input feature map into non-overlapping regions or windows. Each window typically has a fixed size (e.g., 2x2 or 3x3) and a fixed stride (the amount by which the window moves across the feature map).
  - ❖ **Average Calculation:** Within each pooling window, the average pooling operation computes the average value of the elements (usually pixel values or feature activations). This average value becomes the output of the pooling operation for that particular region.
  - ❖ **Down-sampling:** By replacing each region with its average value, the spatial dimensions of the feature map are reduced. For example, if a 2x2

average pooling layer is applied with a stride of 2, the output feature map will have half the height and width of the input feature map.

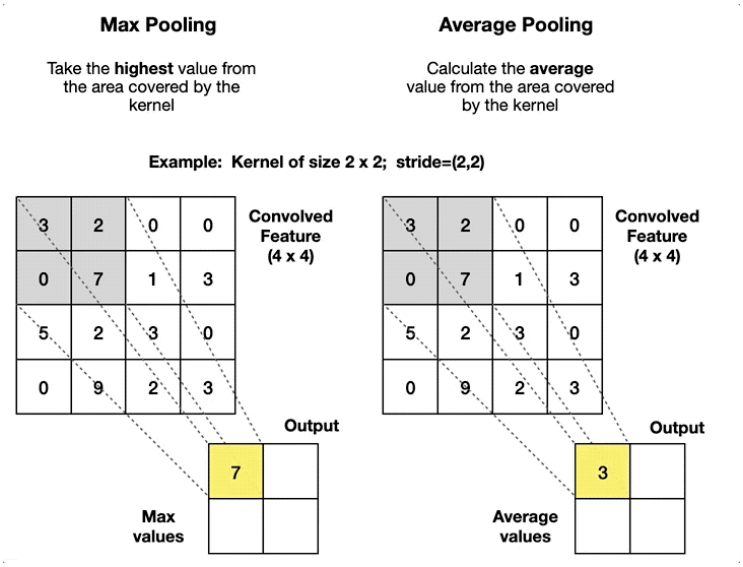


Figure II.8: Max and Average Pooling operations

- **Fully Connected Layer:** Fully Connected Layer, also known as a Dense Layer, is a type of layer where each neuron in the layer is connected to every neuron in the previous layer, similar to traditional neural networks. However, in the context of CNNs, Fully Connected Layers are typically used at the end of the network to perform high-level reasoning on the features extracted by the preceding convolutional and pooling layers.

  Fully Connected Layers are responsible for learning global patterns in the data extracted by the preceding convolutional and pooling layers. They integrate spatial information from the previous layers and produce the final output of the network, often used for classification or regression tasks.

  Before passing the feature maps from the convolutional and pooling layers to the Fully Connected Layer, they are typically flattened into a 1-dimensional vector. This flattening process reshapes the multi-dimensional feature maps into a single vector, which can then be fed into the Fully Connected Layer.

  Each neuron in the Fully Connected Layer has its own set of weights and biases, which are learned during the training process using backpropagation. These weights and biases determine how the input features are combined to produce the output of the layer.

  Just like in traditional neural networks, Fully Connected Layers often apply an activation function (e.g., ReLU, sigmoid, tanh) to the weighted sum of inputs from the previous layer. This introduces non-linearity into the network, allowing it to learn complex relationships in the data ***(38)***.

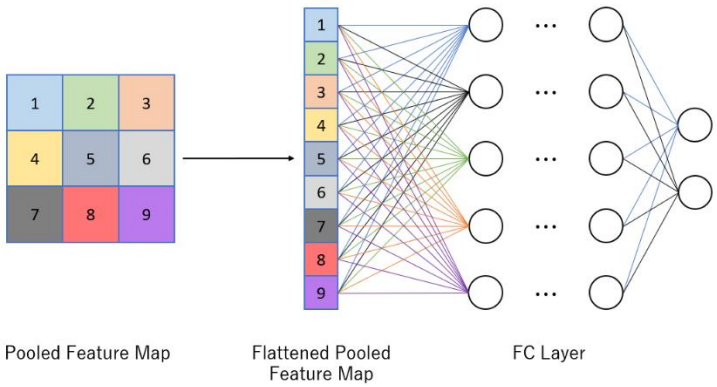


Figure II.9 : Fully Connected Layer

- **Flatten layer:** In convolutional neural networks (CNNs), a flatten layer is typically used to transform the multi-dimensional feature maps produced by convolutional and pooling layers into a one-dimensional vector. This flattened vector serves as the input to the fully connected layers (also known as dense layers) in the network.

  The flatten layer reshapes the input tensor into a one-dimensional vector by concatenating all the elements of the input feature maps along a single axis. For example, if the input feature maps have dimensions [batch-size, height, width, channels], the flatten layer will produce an output vector of shape [batch-size, height * width * channels].

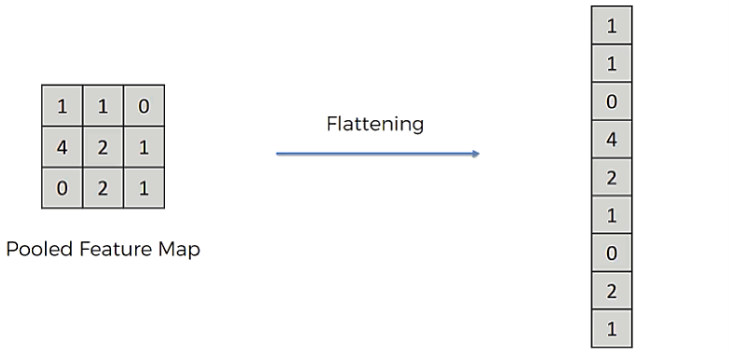


Figure II.10 : Flatten layer operation

- **Classification layer:** A classification layer typically refers to the final layers of a neural network architecture specifically designed for performing classification tasks. These layers are responsible for transforming the high-level features learned by the preceding layers into class probabilities or scores, indicating the likelihood of each class.

II.8-Activation Layers

In CNN, activation functions play a crucial role in introducing non-linearity into the network, enabling it to learn complex patterns and relationships in the data.

- **Rectified Linear Unit:** The Rectified Linear Unit (ReLU) is one of the most widely used activation functions in neural networks, particularly in deep learning architectures. It is a simple yet effective non-linear function that introduces non-linearity into the network, allowing it to learn complex patterns in the data. ReLU activation function is defined as :

$$f(x) = \max(0, x) = \begin{cases} E & if \quad E \geq 0 \\ 0 & otherwise \end{cases} \qquad \text{Eq II.4}$$

Where x is the input to the neuron. In other words, ReLU outputs the input value if it is positive and zero otherwise. Visually, ReLU looks like a piecewise linear function with a slope of 1 for positive inputs and zero for negative inputs ***(62)***.

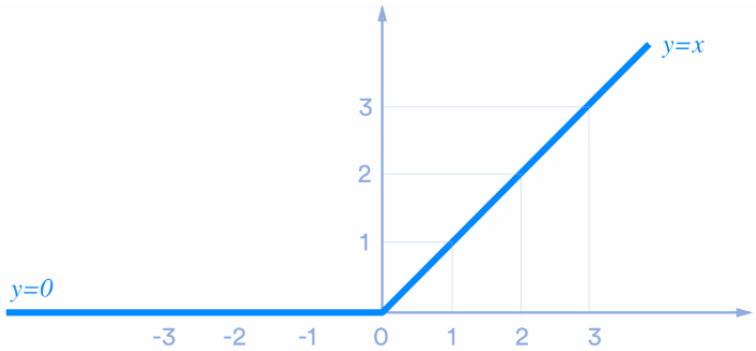


Figure II.11 : Rectified Linear Unit (ReLU)

- **Leaky ReLu:** Leaky Rectified Linear Unit (Leaky ReLU) is an activation function commonly used in neural networks, particularly in deep learning architectures. It is an extension of the traditional Rectified Linear Unit (ReLU) activation function and addresses some of its limitations, particularly the "dying ReLU" problem. Leaky ReLU activation function is defined as :

$$f(x) = \begin{cases} x, & if\ x > 0 \\ \alpha x, & if\ x \leq 0 \end{cases} \qquad \text{Eq II.5}$$

where α is a small positive slope coefficient (typically a small constant, such as 0.01), which is introduced to address the "dying ReLU" problem.

- **Exponential Linear Unit:** The Exponential Linear Unit (ELU) is an activation function commonly used in neural networks, particularly in deep learning architectures. It was introduced to address some of the limitations of other activation functions like the rectified linear unit (ReLU). ELU is defined as follows :

$$ELU(x) = \begin{cases} x & if\ x \geq 0 \\ \alpha(e^x - 1) & if\ x < 0 \end{cases} \qquad \text{Eq II.6}$$

Here, α is a hyperparameter controlling the value to which the function saturates for negative inputs ***(63)***.

Unlike ReLU, ELU is a smooth activation function. It is continuously differentiable everywhere, including at $x = 0$, which can help mitigate some optimization challenges associated with ReLU, such as the "dying ReLU" problem.

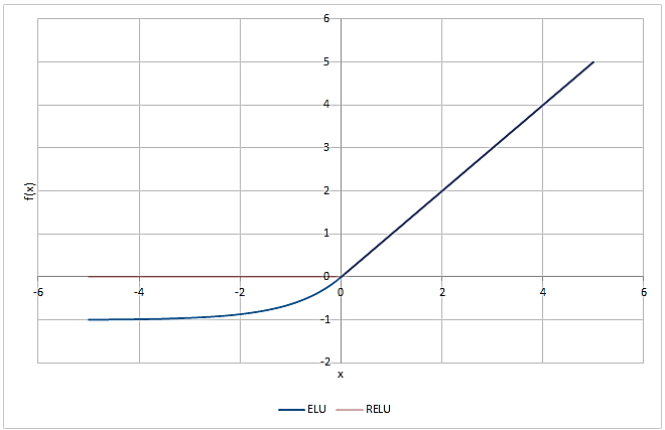


Figure II.12 : ELU and RELU

- **Sigmoid:** The sigmoid function, also known as the logistic function, is a popular activation function used in artificial neural networks, particularly in the early days of neural network research. It maps any real-valued number to a value between 0 and 1, making it useful for binary classification tasks where the output needs to represent probabilities. The formula for the sigmoid function is :

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \text{Eq II.7}$$

where z is the input to the function ***(38)***.

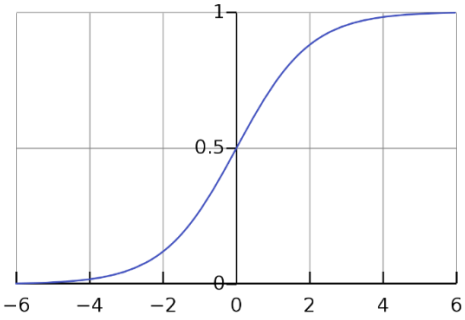


Figure 2.13 Sigmoid function

- **Hyperbolic tangent:** The hyperbolic tangent function, often denoted as tanh, is a non-linear activation function commonly used in artificial neural networks, including multilayer perceptrons (MLPs). It squashes the input values to the range $(-1,1)$, making it suitable for tasks where the output needs to be normalized or centered around zero. Here's an explanation of the tanh function along with some references. tanh function is defined as:

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad \text{Eq II.8}$$

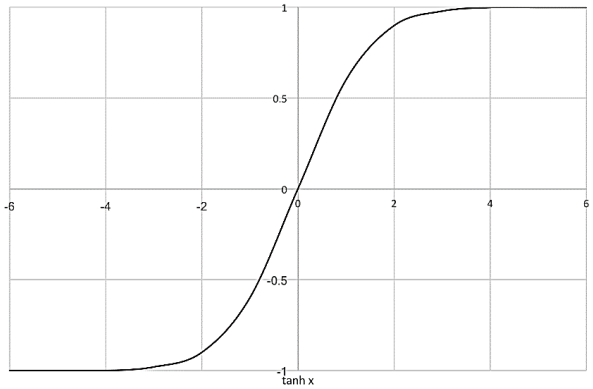


Figure II.14: Hyperbolic tangent function

It is an S-shaped curve that is symmetric around the origin. The tanh function maps any real-valued input to the range (−1,1), with values close to -1 for negative inputs, close to 1 for positive inputs, and approximately 0 for inputs around 0 ***(64)***.

II.9-Conclusion

This chapter provided an overview of artificial intelligence and its subdomains, including machine learning (ML) and deep learning (DL), along with their various types, illustrated using explanatory diagrams. Subsequently, we delved into neural networks in a broad sense, followed by an in-depth exploration of Convolutional Neural Networks (CNNs), supported by figures and flowcharts elucidating the distinct layers of CNN architecture.

Chapter III: Data Preparation & Experimental Results

III.1-Introduction

In this chapter, we will discuss the programming environment, the language, and the libraries that helped us implement our solution. We will also describe the database we used and the 2D CNN network we created for the classification and detection of brain tumor diseases. Furthermore, we will provide a detailed analysis of the results we obtained. Finally, we will conclude with a comparison of the results with other studies focused on the same case and test our model with some MRI images.

III.2-Software Material Used

III.2.1-Python

Python is a high-level, interpreted programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and allows programmers to express concepts in fewer lines of code compared to other languages. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, One of the key features of Python is its extensive standard library, which provides support for many common programming tasks such as string manipulation, file I/O, networking, and web development. Additionally, Python has a large and active community, which contributes to its ecosystem with a vast array of third-party libraries and frameworks for various purposes, including data analysis, machine learning, web development, and scientific computing, Python's syntax is clean and easy to learn, making it an ideal language for beginners. Its simplicity, combined with its versatility and power, has contributed to its widespread adoption in various domains, including web development, data science, artificial intelligence, and automation ***(65)***.



Figure III.1: python

III.3-Libraries

III.3.1-Scikit-learn

Scikit-learn, often abbreviated as skle2arn, is a popular open-source machine learning library for the Python programming language. It provides simple and efficient tools for data mining and data analysis and is built on top of other popular Python libraries such as NumPy,

SciPy, and matplotlib. Scikit-learn is designed to be user-friendly and accessible, making it an excellent choice for both beginners and experienced machine learning practitioners, The library includes various tools for classification, regression, clustering, dimensionality reduction, model selection, and preprocessing of data. It also provides a consistent interface for a wide range of machine learning algorithms, allowing users to easily experiment with different models and techniques ***(66)***.



Figure III.2: Scikit-learn

III.3.2-Keras

Keras is an open-source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit (CNTK), or Theano. Keras was developed by François Chollet and was released in March 2015. It provides a simple and elegant way to create deep learning models and is widely used for developing and experimenting with neural networks, One of the key features of Keras is its user-friendliness and simplicity, which makes it suitable for beginners as well as experts in deep learning. It allows for easy and fast prototyping of neural networks due to its modular design, which enables the building of complex models using simple building blocks called layers. Keras also supports both convolutional and recurrent neural networks, along with their combinations ***(67)***.



Figure III.3: Keras

III.3.3-TensorFlow

TensorFlow is an open-source machine learning framework developed by Google Brain Team and released in 2015. It provides a comprehensive ecosystem of tools, libraries, and community resources for building and deploying machine learning models efficiently. TensorFlow is designed to facilitate the development of various machine learning tasks, including but not limited to neural networks, deep learning, reinforcement learning, and natural language processing, at its core, TensorFlow represents computations as data flow graphs. Nodes in the graph represent mathematical operations, while edges represent the multidimensional data arrays (tensors) communicated between them. This enables efficient

parallel execution across multiple devices like CPUs, GPUs, and TPUs (Tensor Processing Units) ***(68)***.



Figure III.4: TensorFlow

III.3.4-Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides a wide variety of plotting functions and supports various types of plots including line plots, scatter plots, bar plots, histograms, and many more. Matplotlib is highly customizable, allowing users to adjust almost every aspect of their plots, including colors, labels, axes, and annotations, Matplotlib is widely used in the scientific and data analysis communities due to its flexibility and ease of use. It integrates well with other Python libraries such as NumPy and Pandas, making it a powerful tool for data visualization and analysis.



Figure III.5: matplotlib

III.4-Development environment

III.4.1-Anaconda

Anaconda is an open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. It provides a rich collection of packages and tools for data science, machine learning, and scientific computing, including libraries such as NumPy, SciPy, pandas, and scikit-learn. Anaconda also includes its own package manager called conda, which makes it easy to install, update, and manage packages and dependencies ***(69)***.



Figure III.6: Anaconda

III.4.2-JupyterLab

JupyterLab is an open-source web-based interactive development environment (IDE) for Jupyter notebooks, code, and data. It provides a flexible and extensible architecture that supports various workflows in data science, scientific computing, and machine learning, JupyterLab builds upon the capabilities of Jupyter Notebooks, allowing users to create and edit documents containing live code, equations, visualizations, and narrative text. It also offers a modular and customizable interface, enabling users to arrange multiple documents and interactive components (such as terminals, text editors, file browsers, and custom views) in a single workspace, One of the key features of JupyterLab is its support for various programming languages beyond Python, including R, Julia, and Scala, making it a versatile tool for data analysis and scientific computing across different disciplines ***(70)***.



Figure III.7: Jupyter lab

III.5-Hardware used materials

Systems implementation is carried out under anaconda development environment in python. As learning CNN is a very greedy task in capacity to calculation, the execution of the programs was carried out on a computer with specs are:

- **Processor:** CPU AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz 6 Cores 12 Threads
- **Graphics Card:** AMD Radeon (TM) Graphics
- **Ram:** 8 GB DDR 4
- **Windows Edition:** Windows 11 Core (64 bit)

as well as on virtual machines in the cloud of gradient paperspace. The latter is dedicated to Deep Learning applications which require the use of several specific libraries such as Keras, Pytorch, and Tensorflow, the virtual machine have a GPU with a NVIDIA RTX5000 graphics card and 16 GB of Memory.

III.6-Proposed System

Figure III-9 illustrates the various steps undertaken prior to training the CNN model with our images. Initially, we preprocess the data, followed by data decomposition. Subsequently, we train the model and conclude with testing and evaluating its performance.

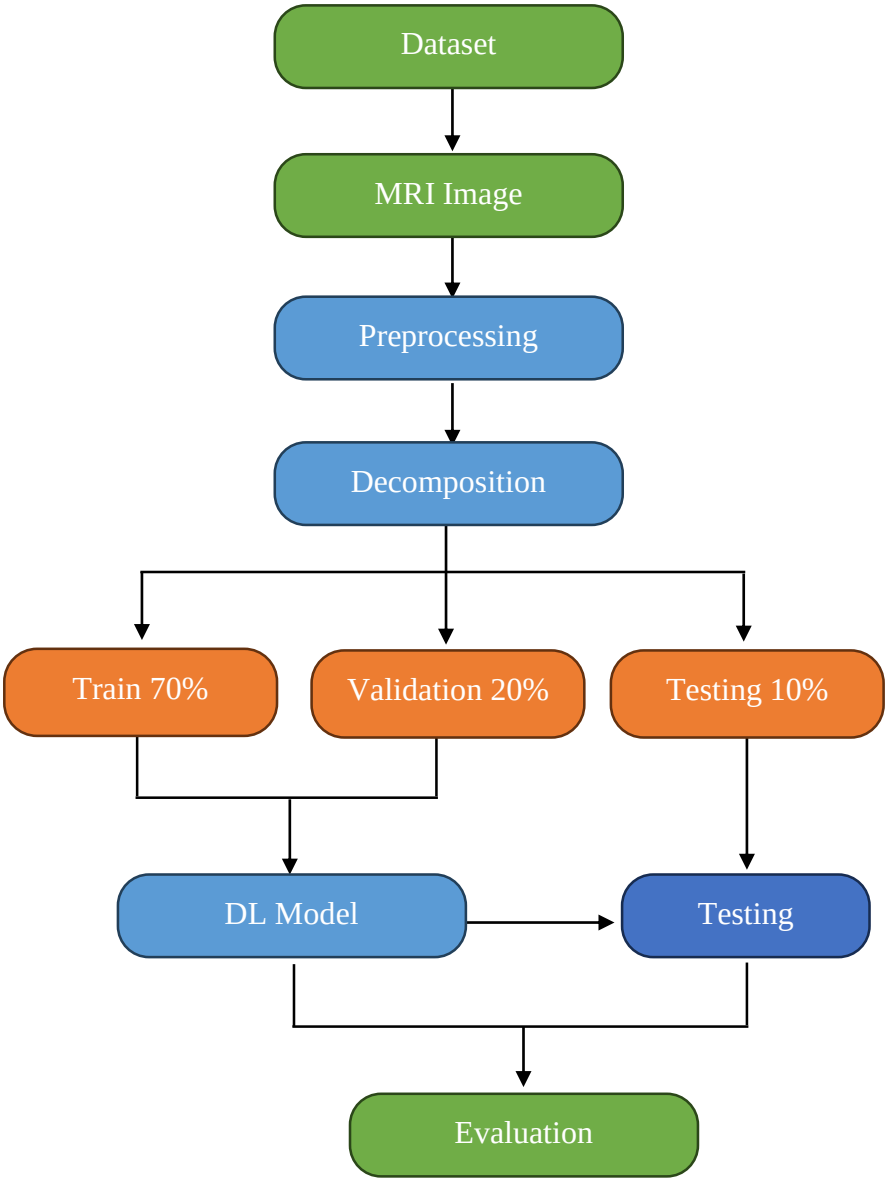


Figure III.8: System block diagram

III.6.1-Dataset Description

The dataset used in this study is Brain MRI Images for Brain Tumor Detection obtained from kaggle.com, The dataset consists of 3000 images grouped into 2 groups, 1500 brain images that have tumors, and 1500 brain images that do not have tumors, (Figure III.2) shows images in the dataset.

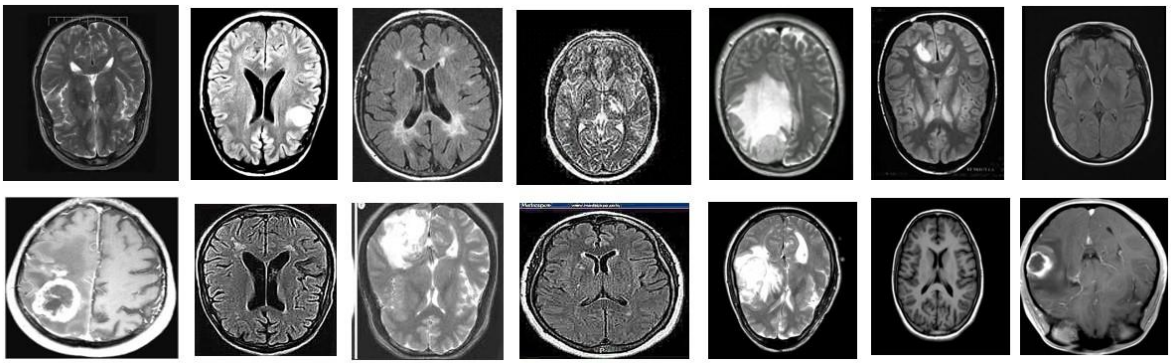


Figure III.9: samples of the dataset

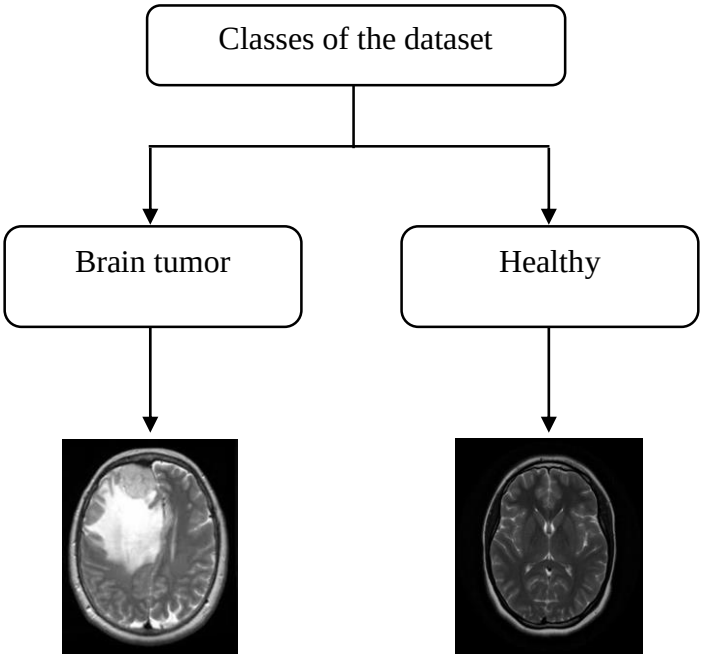


Figure III.10: Dataset Classes

Figure III.11 shows the dataset distribution.

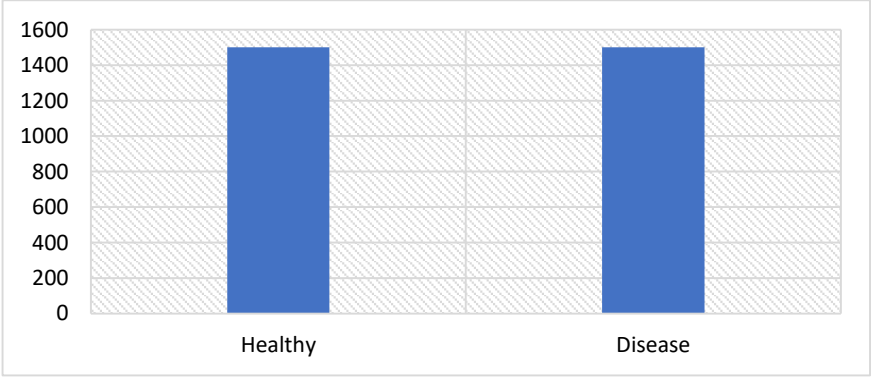


Figure III.11: Dataset distribution

III.6.2-Image Preprocessing

Pre-processing is essential for ensuring smooth training due to the varying intensity, contrast, and size of images. The input image undergoes several pre-processing steps, starting with wrapping and cropping. During wrapping, the input image is aligned with the main object's edge.

The maximum edge distance from the image's edge is determined to preserve the integrity of the object during cropping. Following cropping, the image is resized to the shape (224, 224, 3) to standardize dimensions across the dataset, accommodating images of different sizes.

III.6.3-Decomposition

The images will be divided into 3 important subsets (figure III-12), (Table III-1):

- **Training set:** 70% of the data will be used in the training phase.
- **Validation set**: 20% of the data will be used in the validation phase.
- **Test sets:** 10% of the data will be used in the phase test.

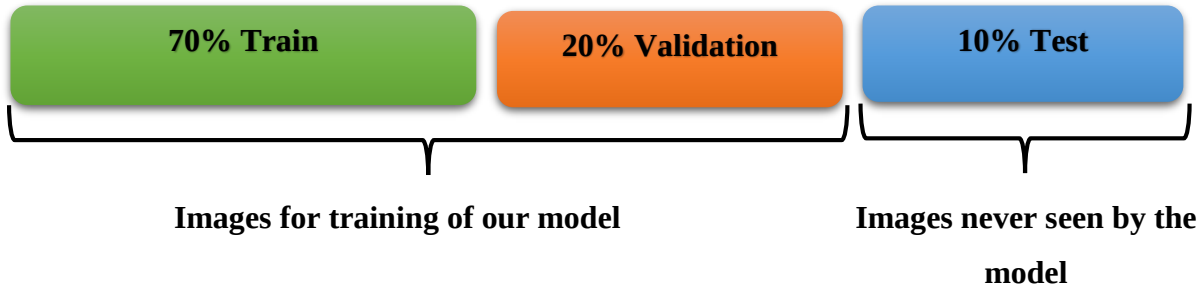


Figure III.12: Splitting dataset

Table III.1: Data distribution of 3 classes

Sets	Number of images
Training set	2100
Validation set	600
Test set	300
Total	**3000**

- **Training**: Used to choose the best parameters for the examples.
- **Validation:** Used to choose the best hyper-parameters and best model for the task.
- **Test:** Estimates true final performance on unseen data and can make the difference between putting it into production and scrapping the whole thing.

III.6.4-CNN Model

We propose the next CNN models architecture, for detecting brain tumors in MRI images depends on various factors, including the size and complexity of the dataset, computational resources available, and specific requirements of the task, having good accuracy.

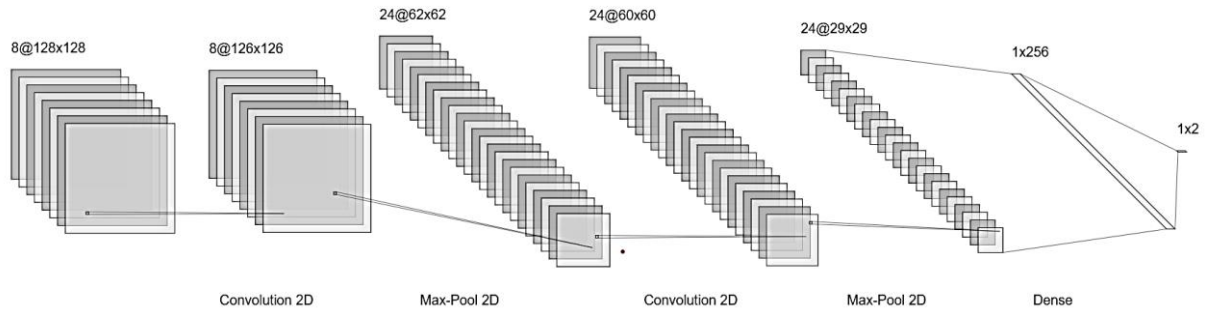


Figure III.13: Schematic diagram of CNN network training procedure

III.6.4.1-Architectures tested

After carrying out several tests, we were able to develop the three best architectures in order to have good results. (Table III-2). To improve the efficiency of processing, we interpose between the processing layers an activation function “ReLu (Rectified Linear Unit)” on the output signals.

Table III-2 shows the three different architectures we obtained in our site, searching for the best architecture and the best model, we add and remove some layers

Number of epochs used is 15 . The results obtained are expressed in terms of Accuracy, training and validation losses. The first architecture that we present in Table III-2 is composed of:

- 2 Convolution layers.
- 2 Layers of Max Pooling.
- Flatten layer.

- 2 Fully connected layers.

The architecture 1 give us the best result.

Table III.2: Best three proposed architectures

		Architecture 1	**Architecture 2**	**Architecture 3**
		Conv2D (3,3,32)	Conv2D (3,3,16)	Conv2D (3,3,32)
		MaxPooling2D ((3,3) strides (2))	MaxPooling2D (2,2)	MaxPooling2D (2,2)
		Conv2D (3,3,32)	Conv2D (3,3,32)	Conv2D (3,3,64)
		MaxPooling2D ((3,3) strides (2))	MaxPooling2D (2,2)	MaxPooling2D (2,2)
		Flatten	Conv2D (3,3,16)	Conv2D (3,3,128)
		Dense (256)	MaxPooling2D (2,2)	MaxPooling2D (2,2)
		Dense (2)	Flatten	Flatten
			Dense (256)	Dense (256)
			Dense (2)	Dense (2)
Accuracy	**Train**	100%	100%	100%
	Val	99%	97.8%	98.1%
	Test	99%	96.6%	96.6%
Loss	**Train**	0.003	0.007	0.003
	Val	0.02	0.11	0.15
	Test	0.13	0.14	0.37

The first convolutional layer of the model. It has 32 filters/kernels of size 3x3 with ReLU activation. The input shape parameter specifies the shape of the input data, which is (224, 224,3) for images with a height and width of 224 pixels and 3 color channels (RGB).

Then we apply a max pooling layer to the model. Max-pooling reduces the spatial dimensions of the input volume. Here, it's reducing the size of the feature maps by taking the maximum value within each 3x3 window and moving the window by 2 pixels (strides=2).

We proceed in the same way for the seccend and third layer with another convolutional layer with 32 filters/kernels of size 3x3 and ReLU activation and max-pooling layer with the same configuration as the previous one.

The fourth layer is the flatten layer that flattens the 2D output from the convolutional layers into a 1D vector. This is necessary before passing the data to fully connected layers.

The last two are fully connected (dense) layer with 256 neurons and ReLU activation, with another one that adds the output layer with the number of neurons equal to the number of classes in your dataset (represented by len(image_classes)). The activation function used here is sigmoid, which is suitable for binary classification tasks where each class is independent.

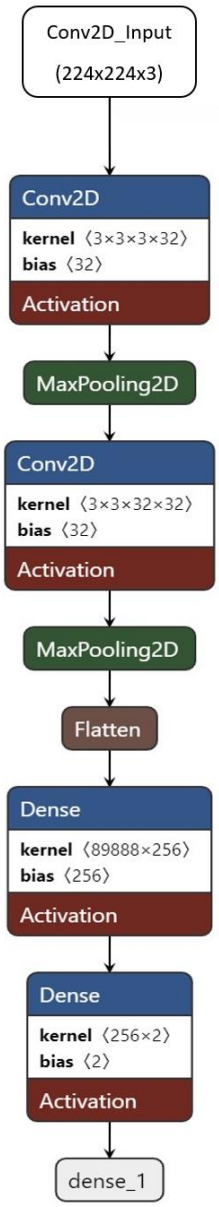


Figure III.14: Best Model Architecture.

III.7-Used Methods

III.7.1-Model Check Point callbacks

First, we create a TensorBoard callback object. Callbacks are functions that can be applied at various stages of the training process. In this case, the TensorBoard callback is used to log data for visualization in TensorBoard. It takes the log dir parameter, specifying the directory where the logs will be saved.

III.7.2-Compiler used

Compiler is a method used to configure the learning process of the model:

- **Optimizer adam:** Adam is a popular optimization algorithm used to update network weights iteratively based on training data.
- **Loss:** we used **sparse categorical crossentropy**, This parameter defines the loss function used to compute the difference between the model's predictions and the actual labels during training. **Sparse categorical crossentropy** is commonly used when the labels are integers. It's like a penalty score that tells us how well (or, to be precise, how poorly) the model is doing on the training data.
- **Metrics:** we used **accuracy**, during training, you typically want to monitor some metrics to evaluate the performance of the model. Here, 'accuracy' is specified as the metric to monitor, which calculates the accuracy of the model predictions.

III.8-Results and Discussion

In order to show the results obtained by our 2D CNN model, we have illustrated the different results in terms of:

- Classification report.
- Accuracy.
- Loss.
- Confusion matrix.

We took a number of epochs equal to 15 epochs because. We took a number of epochs equal to 15 epochs, we observed that the performance stops increasing and stabilizes at 15 epochs, indicating that further epochs are unnecessary. Reducing the number of epochs will also benefit us by saving time during the extraction process of our program, we save the best model during the training.

III.8.1-Accuracy and loss graph

Figure III.15 plotting the training history of a machine learning model. It visualizes the training and validation loss as well as the training and validation accuracy over different epochs, shows the graphs of precision and loss for both training and validation as a function of the number of epochs. We observe that the precision for both training and validation increases proportionally with the number of epochs. Conversely, the loss for both training and validation decreases inversely with the number of epochs.

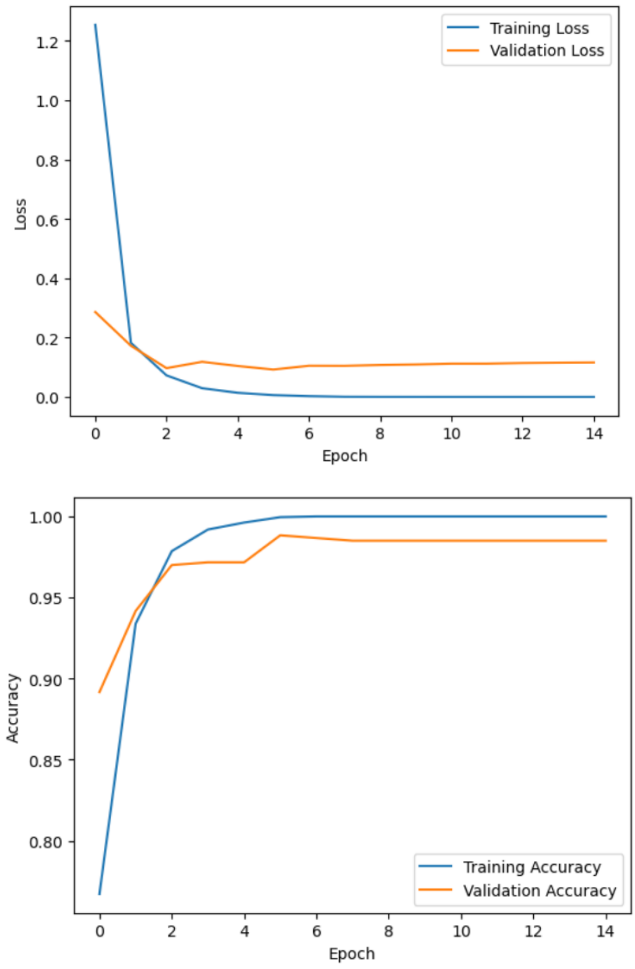


Figure III.15: Accuracy and losses graph.

III.8.2-Classification report

A classification report (figure III.16) is a summary of the performance of a classification model on a dataset. It provides several evaluation metrics, typically including precision, recall, F1-score, and support for each class in the classification task.

- 1 Indicates brain tumor.
- 0 Indicates healthy.

	precision	recall	f1-score	support
0	1.00	0.99	1.00	309
1	0.99	1.00	0.99	291
accuracy			0.99	600
macro avg	0.99	1.00	0.99	600
weighted avg	1.00	0.99	1.00	600

Figure III.16 : Validation Classification report

- **Precision:** Is the ratio of correctly predicted positive observations to the total predicted positives. It measures the accuracy of positive predictions. A high precision indicates a low false positive rate**.**

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

- **Recall:** also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all actual positives. It measures the ability of the classifier to find all positive instances. A high recall indicates a low false negative rate.
- **F1-score:** is the harmonic mean of precision and recall. It provides a single score that balances both precision and recall. It is useful when you have uneven class distribution.

$$F1 - score = \frac{\text{Precision } + \text{Recal}}{2}$$

- **Support:** is the number of actual occurrences of the class in the specified dataset. It is the number of true instances for each class.

III.8.3-Confusion matrix

The confusion matrix (figure III.17) is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It presents a summary of the model's performance, showing the number of correct and incorrect predictions categorized by class. It's typically organized into rows and columns, where rows represent the true classes and columns represent the predicted classes. Each cell in the matrix represents the count of instances where the true class (row) was predicted as the predicted class (column).

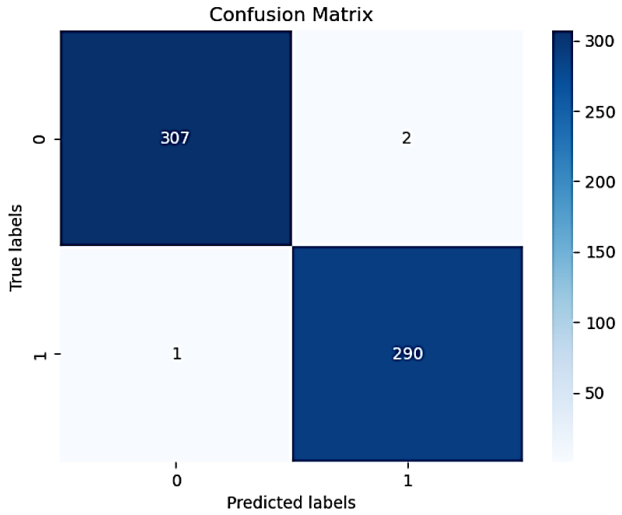


Figure III.17 : Confusion Matrix Validation

We save the best model during the training phase using callbacks Model check point and we obtain the best results as depicted in Table III.5.

Table III.3: Best model results

	Accuracy	**Loss**
Train	100 %	0.003
Validation	99%	0.02

III.9-Final Test

In this section we utilize the best model saved from the previous phase and assess its performance on the testing set, as represented in Table III.4, consists of data that were not utilized by the model during the training or validation phases. This evaluation provides insights into the model's generalization and performance on unseen data.

Table III.4: Test results

	Accuracy	**Loss**
Test	99%	0.13

III.9.1-Confusion matrix

Figure III.18 displays the confusion matrix, offering a visual representation of the prediction results for different types of activities based on the final test data, which were not utilized during the model's training or validation stages.

According to this matrix we notice that there is a problem in the classification of the images Our CNN model classifies 2 images as containing tumors but that does not contain any tumor from total of 144 that has a tumor, we believe that these two images indicate a potential for tumor development in the future.

And successfully predict all 156 images that does not has brain tumor as healthy.

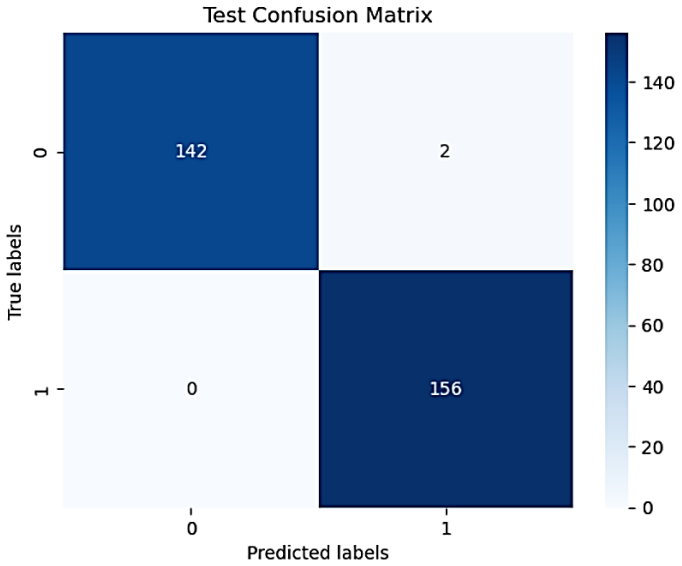


Figure III.18: Confusion Matrix of Test

III.9.2-Test Classification report

After testing, a classification report (figure III.19) represent the test result we get a accuracy of 99% similar to f1 score and that's because our data are balanced.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	144
1	0.99	1.00	0.99	156
accuracy			0.99	300
macro avg	0.99	0.99	0.99	300
weighted avg	0.99	0.99	0.99	300

Figure III.19: Classification report Test

III.9.3-Detection of Our Model

After we chose the model that gives better results with an accuracy of 99%, we chose MRI images Never seen by the model of brain tumors randomly and we tested them on this model, and the following (figure III.20) represent the detection for a classification of 2 classes.

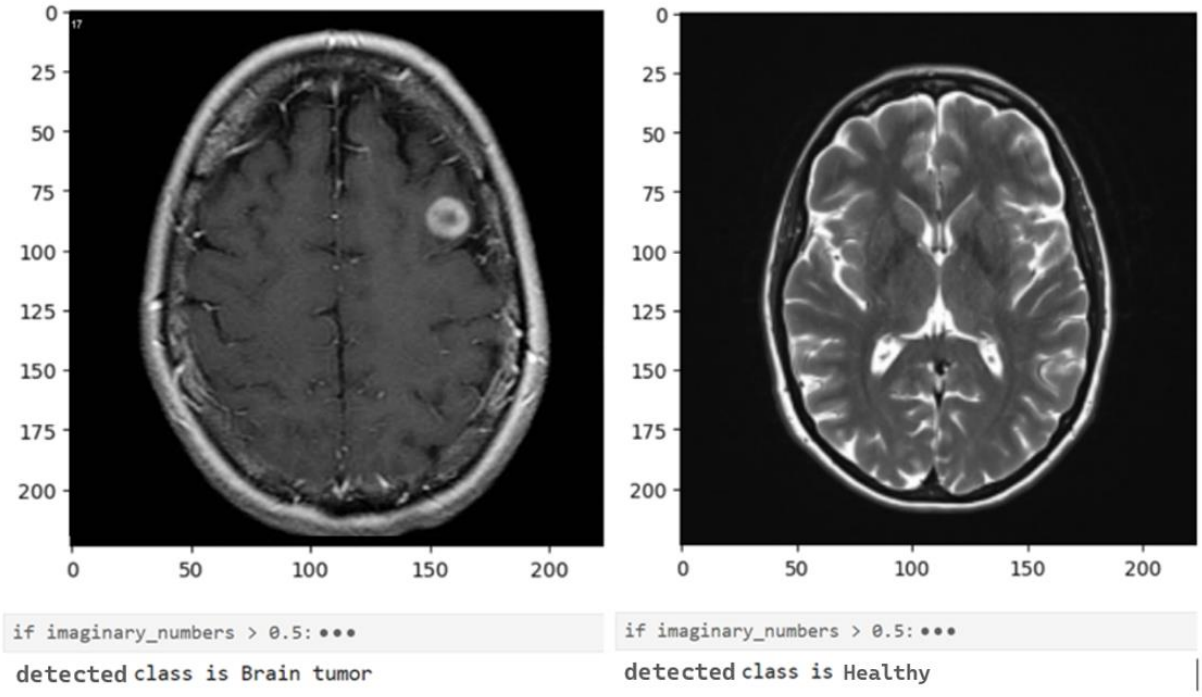


Figure III.20: Detection of some MRI images

III.10-Comparison with related work

We took best results from different articles and compare it with our result (Table III.6):

Table III.5: Comparison with other studies

Study	Method/Classifier	Dataset	Performance
(71)	CNN	2065 images with augmentation methods	Accuracy 93%
(72)	MobileNet V2	4600 images	Accuracy 92%
	Inception V3		Accuracy 91%
	VGG19		Accuracy 88%
(73)	Correlation learning mechanism (CLM) with CNN	3064, CT brain scan images	Accuracy 96% recall 95%
(74)	KNN classifier (proposed), RF, NB, and RBF	227 MRI images	Accuracy 95.6% AUC 99%
Our Study	**CNN Model**	**3000 MRI Images**	**Accuracy 99%** **Loss 0.02%**

In addition to the related work that we describe in the first chapter, the performance of our model has been compared with the results obtained by other studies focused on the same case study of brain tumors (Table III.)

For example, in article ***(71)***, a 10-layered CNN with data augmentation was proposed, achieving an accuracy of 93%.

The second article ***(72)*** aimed to boost accuracy by using a transfer learning strategy. The authors implemented three transfer learning approaches using pre-trained CNN models, namely VGG19, Inception V3, and MobileNet V2. They obtained accuracy rates of 88.22%, 91%, and 92%, respectively. The authors concluded that MobileNet V2 is the most efficient model compared to the other models.

III.11-Conclusion

In this chapter, a CNN model for the segmentation of MRI images of brain tumors into two classes tumors and non-tumors is presented. This model provided the best accuracy achieved, and we compared it with other neural network models. These medical images underwent preprocessing and resizing before being processed by our CNN model. Training, validation, and testing were performed on 3,000 high-resolution MRI images.

The performance of the CNN model is evaluated using several evaluation metrics. Through this experiment, the proposed model outperformed other CNN models in several performance aspects, achieving an accuracy of 99% and a loss of 0.02.

The CNN proved to be the best technique for detecting the presence of brain tumors. Based on performance evaluation metrics and curve analysis, this work demonstrates the ability of a CNN network to detect and classify tumors in the brain with a high accuracy rate.

General Conclusion

The objective of this project is to build a robust CNN model that can classify whether a subject has a tumor or not based on Brain MRI scan images, with an acceptable accuracy for medical-grade applications.

The CNN proved to be the best technique for predicting and detecting the presence of brain tumors. Based on performance evaluation metrics and curve analysis, this work demonstrates the ability of a CNN network to detect and classify tumors in the brain with high accuracy. We propose a simple CNN architecture that achieved good accuracy results compared to more complex models used by other researchers.

There are several potential reasons why our simple convolutional neural network model might achieve higher accuracy than more complex models like VGG19 and MobileNet V2 for image classification. One reason is that our dataset is relatively simple, and more complex models might not perform as well. Simpler models often generalize better on smaller datasets because they have fewer parameters and are less likely to overfit.

The training process, including hyperparameters like learning rate, batch size, and the number of epochs, can significantly impact model performance. Additionally, a simpler model might focus on more relevant features for our specific classification task, while complex models might extract a lot of redundant or irrelevant features due to their depth and complexity. After several attempts and adjustments, we chose the best-balanced model that allowed us to achieve high accuracy and good results. We conducted tests and detect on it and were pleased with the outcome.

We sincerely hope that our intelligent system for detecting brain tumors using MRI images can be used by neuro-specialists and radiologists and will be acceptable for medical-grade applications, including hospital-level emergency departments where precision and time can save lives.

Bibliography

1. (Accessed may 2023) American Brain Tumor Association (ABTA),Brain Tumor Statistics. Available at: https://www.abta.com/

2. World Health Organization (WHO), Cancer Today. Available at: https://www.who.int/

3. Abdusalomov AB, M. MWT. (2023) Brain Tumor Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging. *Cancers (Basel)*, 2.

4. Hajianfar, G. ea. (2021) A comprehensive review of principles, methodologies and advancements in magnetic resonance imaging. *International Journal of Biomedical Imaging* **2021**.

5. Smith-Bindman, R. &M. (2019) The Use of Magnetic Resonance Imaging in Medical Practice. *New England Journal of Medicine* **380(24)**, 2271–2279.

6. American Brain Tumor Association. Available at: https://www.abta.org/about-brain-tumors/types-of-brain-tumors/

7. Raghavapudi, H. SP,&KV. (2021) Brain tumor causes, symptoms, diagnosis and radiotherapy treatment. *Current Medical Imaging* **17**, 3.

8. Lee, J. WaWPYaHSaBPaKSaDJaGAJaWWMaWSKaKRao. (2010) Morphological characteristics of brain tumors causing seizures. *Archives of neurology* **67**, 336--342.

9. Waxweiler, R. JaAVaLSSaHMaLJW. (1983) Mortality from brain tumor and other causes in a cohort of petrochemical workers. *Journal of the National Cancer Institute* **70**, 75--81.

10. Louis, D. N,OH,WOD,CWK,BPC,JA.&K. (2007) The 2007 WHO classification of tumours of the central nervous system. *Acta neuropathologica* **114**, 97--109.

11. Reuss, D. E,&S. (2016) WHO Classification of Tumours of the Central Nervous System..

12. Louis, D. NaPAaRGaVDAaFBDaCWKaOHaWODaKPaEDW. (2016) The 2016 World Health Organization classification of tumors of the central nervous system: a summary. *Acta neuropathologica* **131**, 803--820.

13. Smith, J. JA,&LC(. (2021) Architecture of a Diagnostic Support System for Brain Tumor Disease: Integration of Imaging Technologies and Computational Algorithms. *Journal of Medical Imaging* **17(3)**, 245-261.

14. Smith, S. M. (2002) Fast robust automated brain extraction. *Human brain mapping* **17**, 143--155.

15. Roy, S. aNSaMIKaBSK. (2013) *A review on automated brain tumor detection and segmentation from MRI of brain*. arXiv preprint arXiv:1312.6150.

16. Woods, R. CGaRE. (2018) Gaussian Filtering. In : *Digital Image Processing (4th Edition)*. Pearson.

17. Patenaude, B. SSM,KDN,&J. (2011) *A Bayesian model of shape and appearance for subcortical brain segmentation*. NeuroImage, 56(3).

18. Sled, JG. ZAP,EAC. (1998) A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Transactions on Medical Imaging* **17**, 87–97.

19. Gonzalez, RC. WRE. (2008) Digital Image Processing..

20. Maes, F. CA,VD,MG,SP. (1997) Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging* **16**, 187–198..

21. Tao, X. CL,SY,ZZ. (2017) A review of image denoising algorithms, with a new one. Multimedia Tools and Applications., 3205–3228.

22. Ward, A. CK. (2015) An overview of gradient-descent optimization algorithms. *Journal of Machine Learning Research* **16**, 277–303.

23. Krueger, R. L. (2011) Introduction to Biomedical Imaging..

24. Senthilkumaran, N. aVS. (2016) *Image segmentation by using thresholding techniques for medical images* 6. Academy and Industry Research Collaboration Center (AIRCC).

25. Atam P. Dhawan, J. SSaSL. (2011) *Medical Image Analysis*..

26. Ronneberger, O. PFaTB. (2015) *U-net: Convolutional networks for biomedical image segmentation*. Springer, Cham.

27. Smith, J. JA,&LC. (2021) rchitecture of a Diagnostic Support System for Brain Tumor Disease: Integration of Imaging Technologies and Computational Algorithms. *Journal of Medical Imaging* **17**, 245-261.

28. Zhang, H. aFJEaGSA. (2005) *A fast texture feature extraction method for region-based image segmentation* 5685. SPIE.

29. Menze BH, J. ABSea. (2015) *The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)*. IEEE Trans Med Imaging.

30. CNNs: LeCun, Y. BY,&HG. (2015) Deep learning.. *Nature* **521**, 436-444.

31. MacQueen, J. B. (1967) Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* **Vol. 1, No. 14**, 281-297.

32. Kingma, D. P,&WM. (2013) Auto-encoding variational bayes. *arXiv preprint*.

33. Shah, A. &RP. (2007) A survey of decision support systems for brain tumor diagnosis. *Journal of healthcare engineering* **1**, 1-17.

34. Febrianto, D. aSIaNH. (2020) *Convolutional neural network for brain tumor detection* 771. IOP Publishing.

35. Tazin, T. aSSaGPaAFIaISaMKMaBSaISAaAHao. (2021) A robust and novel approach for brain tumor classification using convolutional neural network. *Computational Intelligence and Neuroscience* **2021**.

36. Nazir, M. aWFaAKS. (2015) A simple and intelligent approach for brain MRI classification. *Journal of Intelligent \& Fuzzy Systems* **28**, 1127--1135.

37. I. Soesanti, M. HAaIA. (2020) Classification of Brain Tumor MRI Image using Random Forest Algorithm and Multilayers Perceptron., 6.

38. Goodfellow, I. aBYaCA. (2016) Deep learning..

39. Esteva, A. aKBaNRAaKJaSSMaBHMaTS. (2017) Dermatologist-level classification of skin cancer with deep neural networks. *nature* **542**, 115--118.

40. Doshi-Velez, F. aKB. (2017) Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

41. Bostrom, N. (2014) *Superintelligence: Paths, dangers, strategies*. Oxford University Press.

42. Bishop, C. MaNNM. (2006) *Pattern recognition and machine learning* 4. Springer.

43. Goodfellow, I. BY,&CA. (2016) Deep Learning. MIT Press 136-142.

44. Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*. MIT Press.

45. Sutton, R. SaBAGao. (1999) Reinforcement learning. *Journal of Cognitive Neuroscience* **11**, 126--134.

46. Sutton, R. SaBAG. (2018) Reinforcement learning: An introduction..

47. Bishop, C. M. (2006) Pattern recognition and machine learning. *Springer google schola* **2**,

48. Bengio, Y. aCAaVP. (2013) Representation learning: A review and new perspective. *IEEE transactions on pattern analysis and machine intelligence* **35**, 1798--1828.

49. Nikolopoulou, K. (Accessed 2023) https://www.scribbr.com/ai-tools/supervised-vs-unsupervised-learning/.

50. Chapelle, O. aSBaZA. (2009) Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks* **20**, 542--542.

51. Murphy, K. P. (2012) Machine learning: a probabilistic perspective..

52. Litjens, G. aKTaBBEaSAAAaCFaGMaVDLJAaVGBaSCi. (2017) A survey on deep learning in medical image analysis. *Medical image analysis* **42**, 60--88.

53. Obermeyer, Z. aEEJ. (2016) Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine* **375**, 1216.

54. Vamathevan, J. aCDaCPaDIaFEaLGaLBaMAaSPaSMao. (2019) Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery* **18**, 463--477.

55. Choi, E. aSAaSWFaSJ. (2017) Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association* **24**, 361--370.

56. Goodfellow, I. aBYaCA. (2016) *Deep learning*. MIT press.

57. Hochreiter, S. (1998) The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 107-116.

58. LeCun, Y. BY,&HG. (2015) Deep Learning. *Nature*, 436-444.

59. Vaswani, A. SN,PN,UJ,JL,GAN.&PI. Attention is All You Need. *Advances in Neural Information Processing Systems* **30**, 2017.

60. McCulloch, W. SaPW. (1943) A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **5**, 115--133.

61. LeCun, Y. aBLaBYaHP. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278--2324.

62. Glorot, X. aBY. *Understanding the difficulty of training deep feedforward neural networks*. JMLR Workshop and Conference Proceedings, 2010.

63. Clevert, DA. aUTaHS. (2015) Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

64. Goodfellow, I. aBYaCA. (2016) *Deep learning*. MIT press.

65. Python. Available at: https://www.python.org/about/

66. Pedregosa, F. aVGaGAaMVaTBaGOaBMaPPaWRaDVao. (2011) Scikit-learn: Machine learning in Pytho. *the Journal of machine Learning research* **12**, 2825--2830.

67. Chollet (Accessed 2015) Keras. Available at: https://keras.io/

68. Abadi, M. aBPaCJaCZaDAaDJaDMaGSaIGaIMao. (2016) $\

69. Available at: https://www.anaconda.com/

70. Available at: https://jupyterlab.readthedocs.io/en/stable/

71. Febrianto, D. aSIaNH. (2020) Convolutional neural network for brain tumor detection. *IOP Conference Series: Materials Science and Engineering* **771**, 012031.

72. Tazin, T. aSSaGPaAFIaISaMKMaBSaISAaAHao. (2021) A robust and novel approach for brain tumor classification using convolutional neural network. *Computational Intelligence and Neuroscience* **2021**.

73. Wo (2023) Deep neural network correlation learning mechanism for CT brain tumor detection. *Neural Computing and Applications* **35**, 14611--14626.

74. Lotlikar, V. SaSNaGA. (2022) Brain tumor detection using machine learning and deep learning: a review. *Current Medical Imaging* **18**, 604--622.

75. kaggle. Available at: www.kaggle.com.