



People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research  
University of Kasdi Merbah, Ouargla  
Faculty of Modern Information and Communication  
Technologies  
Department of Computer Science and Information  
Technology



---

**Domain:** Computer Science and Information Technology

**Specialty:** MASTER2 Industrial Computing

**THEME**

---

**Improving healthcare systems using deep learning techniques**

---

**Supervisor:**

Dr. Aiadi Oussama

**By:**

Hamdat Safa

Hathat Maria

---

**Academic year: 2024/2025**

---

# **Acknowledgment**

First of all, we thank our Almighty God who helped us and gave us the patience and courage to achieve this work during our years of study. we also thank our parents, who encouraged and motivated us to reach this level of study. We are sincerely grateful to our esteemed supervisor, Dr. Oussama Aiadi, for his continuous support, valuable advice, and persistent encouragement throughout our research journey. His insightful guidance and trust in our capabilities have been instrumental at every stage of our thesis writing and problem-solving process. Likewise, we extend our respectful thanks to the jury members. We would be remiss in not mentioning all my teachers for their guidance in the last five years in Kasdi Merbah Ouargla University's Department of Information Technology. A special thanks to our family and friends for always standing by us. Finally, we would like to thank all those who have helped us from near or far during all our studies and in the preparation of this dissertation, our deep gratitude and respect.

# **Dedication**

Praise be to Allah. Alhamdulillah for everything.

dedicate this work to our beloved families who supported and encouraged us through every step. I extend my heartfelt dedication to my teachers and my Department for their education and support. Special thanks to my supervisor, Dr. Oussama Aiadi, for his invaluable guidance and encouragement throughout this journey.

Lastly, to my family and friends who have stood by me, thank you for your love and support. This work is dedicated to all of you.

# ABSTRACT

Accurate image classification is a cornerstone of modern medical diagnostics, enabling healthcare professionals to extract meaningful insights from various types of medical imaging. In particular, identifying abnormalities in MRI scans plays a critical role in early detection and treatment planning for brain-related conditions. This thesis presents a deep learning-based system aimed at improving brain tumor diagnosis through the analysis of high-resolution MRI images.

The proposed approach utilizes a dataset of 5,000 multimodal MRI scans, comprising 2,000 images labeled as healthy and 3,000 as tumor-affected. These images were collected from diverse, credible medical sources and optimized to ensure high quality and diagnostic relevance. A Convolutional Neural Network (CNN) architecture forms the core of the system due to its strong performance in image-based classification tasks. Additionally, Generative Adversarial Networks (GANs) were employed to synthetically expand the dataset by generating realistic MRI images, further improving the system's learning capability.

Experimental results indicate that the proposed system achieves high classification accuracy, demonstrating the effectiveness of combining CNNs, data augmentation, and GANs. These findings underscore the potential of deep learning techniques in supporting and advancing healthcare systems by providing reliable and efficient diagnostic tools.

**Keywords:** Deep Learning, Healthcare, Brain Tumor, MRI, CNN, GAN.

# Résumé

la classification précise des images constitue une pierre angulaire du diagnostic médical moderne, permettant aux professionnels de santé d'extraire des informations significatives à partir de divers types d'imagerie médicale. En particulier, l'identification des anomalies dans les IRM joue un rôle crucial dans la détection précoce et la planification du traitement des pathologies cérébrales. Ce mémoire présente un système basé sur l'apprentissage profond visant à améliorer le diagnostic des tumeurs cérébrales à travers l'analyse d'images IRM haute résolution.

L'approche proposée utilise un ensemble de données composé de 5 000 images IRM multimodales, dont 2 000 étiquetées comme saines et 3 000 comme atteintes de tumeurs. Ces images proviennent de sources médicales diverses et crédibles, et ont été optimisées afin d'assurer une haute qualité et une pertinence diagnostique. Une architecture de réseau de neurones convolutifs (CNN) constitue le cœur du système, en raison de ses performances élevées dans les tâches de classification d'images. De plus, des réseaux antagonistes génératifs (GAN) ont été utilisés pour augmenter artificiellement l'ensemble de données en générant des images IRM réalistes, renforçant ainsi la capacité d'apprentissage du système.

Les résultats expérimentaux indiquent que le système proposé atteint une précision de classification élevée, démontrant l'efficacité de la combinaison des CNN, de l'augmentation de données et des GAN. Ces résultats mettent en lumière le potentiel des techniques d'apprentissage profond pour soutenir et améliorer les systèmes de santé grâce à des outils diagnostiques fiables et efficaces.

**Mots-clés:** Apprentissage profond, Santé, Tumeur cérébrale, IRM, CNN, GAN.

## ملخص

التصنيف الدقيق للصور يُعدّ حجر الأساس في تشخيصات الطب الحديث، إذ يُمكن المتخصصين في الرعاية الصحية من استخراج معلومات ذات مغزى من أنواع مختلفة من الصور الطبية. ويُعدّ تحديد الشذوذات في صور الرنين المغناطيسي (MRI) أمراً بالغ الأهمية في الكشف المبكر والتخطيط العلاجي للأمراض المتعلقة بالدمغ. تهدف هذه الرسالة إلى تقديم نظام معتمد على تقنيات التعلم العميق لتحسين تشخيص أورام الدماغ من خلال تحليل صور الرنين المغناطيسي عالية الدقة. يعتمد النهج المقترح على مجموعة بيانات تتكون من 5000 صورة رنين مغناطيسي متعددة الأنماط، منها 2000 صورة مصنفة على أنها سليمة و3000 صورة تشير إلى وجود ورم. وقد جُمعت هذه الصور من مصادر طبية متنوعة وموثوقة، وتم تحسين جودتها لضمان أعلى درجات الدقة والأهمية التشخيصية. يُشكل نموذج الشبكة العصبية الالتفافية (CNN) الأساس لهذا النظام نظراً لأدائه القوي في مهام التصنيف المعتمدة على الصور. كما تم استخدام الشبكات التوليدية التنافسية (GANs) لتوسيع مجموعة البيانات بشكل اصطناعي من خلال إنتاج صور رنين مغناطيسي واقعية، مما ساهم في تعزيز قدرة النموذج على التعلم.

تشير النتائج التجريبية إلى أن النظام المقترح يحقق دقة تصنيف عالية، مما يبرز فعالية الجمع بين شبكات CNN وتقنيات تعزيز البيانات والشبكات التوليدية في تحسين الأداء. وتؤكد هذه النتائج الإمكانيات الكبيرة لتقنيات التعلم العميق في دعم وتطوير أنظمة الرعاية الصحية، من خلال تقديم أدوات تشخيصية موثوقة وفعالة.

**الكلمات المفتاحية:** التعلم العميق، الرعاية الصحية، أورام الدماغ، الرنين المغناطيسي، الشبكات العصبية الالتفافية، الشبكات التوليدية التنافسية.

# Contents

<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>vii</b>
<b>General Introduction</b>	<b>1</b>
1    Introduction . . . . .	1
2    Problematic . . . . .	2
3    Motivation Behind the Thesis . . . . .	2
4    Contributions . . . . .	3
5    Structure of the Thesis . . . . .	3
<b>Chapter 1: Work Background</b>	<b>5</b>
1    Introduction . . . . .	5
2    Definition of Artificial Intelligence . . . . .	5
3    Definition of Machine learning . . . . .	5
4    Definition of Deep learning . . . . .	6
5    Types of learning . . . . .	6
5.1    Learning problems . . . . .	6
6    The difference between ML and DL . . . . .	7
7    Computer Vision . . . . .	7
8    Applications of Computer Vision . . . . .	8
8.1    Computer Vision for Fabric Defect Detection . . . . .	8
8.2    Computer Vision for Healthcare research . . . . .	8
8.3    Computer Vision for Dynamic Intrusion Surveillance and Threat Analysis . . . . .	8
8.4    Computer Vision for Analyse medical images . . . . .	8
8.5    Computer Vision for Remote health monitoring . . . . .	9
8.6    Computer Vision for Reduce treatment time . . . . .	9
8.7    Computer Vision for Lowering diagnostic mistakes . . . . .	10
9    Medical image . . . . .	10
9.1    X-rays . . . . .	10
9.2    Magnetic resonance imaging (MRI) . . . . .	11
9.3    Computerized tomography (CT scan) . . . . .	11

---

9.4	Ultrasound (US)	12
9.5	Emission Computed Tomography	12
10	Image segmentation	13
11	Artificial Neural Network (ANN)	13
11.1	The biological inspiration for ANN	15
11.2	Architecture of Artificial Neural Network	16
12	Convolutional Neural Network (CNN)	17
12.1	Definition of Convolutional Neural Network	17
12.2	Convolutional layer	17
12.3	Pooling layer	18
12.4	Fully connected layer	19
13	CNNS Work	20
13.1	Input (The Training Data)	20
13.2	Filter	20
13.3	Convolved Feature	20
13.4	Receptive Field	20
13.5	Depth	20
13.6	Depth Column	20
13.7	Stride	21
13.8	Zero Padding	21
13.9	Parameter Sharing	21
13.10	Dilation	21
13.11	ReLU Layer	21
14	Popular CNN Architectures	22
14.1	LeNet	22
14.2	AlexNet	22
14.3	VGGNet	22
15	Generative Adversarial Networks (GANs)	23
15.1	What Is The Generative Adversarial Networks	23
15.2	How Does The GAN Work	23
16	Conclusion	24
<b>Chapter 2: Proposed Method</b>		<b>25</b>
1	Introduction	25
2	An Overview on The Proposed Method	25
3	The CNN Model	27
4	The GAN Model	28
5	Masure The Size of The Model using FLOPs	30
6	CNN-GAN hybrid scheme	30
7	Conclusion	31

---

<b>Chapter 3: Experimental Results</b>	<b>32</b>
1 Introduction . . . . .	32
2 Experimental Setup . . . . .	32
2.1 Dataset Description . . . . .	32
2.2 Development Frameworks . . . . .	33
2.3 Evaluation metrics . . . . .	35
3 Comparison And Results . . . . .	36
3.1 Experiment 1:CNN Model . . . . .	36
3.2 Experiment 2:Generating images using results . . . . .	37
3.3 Experiment 3:The CNN+GAN Model . . . . .	40
4 Conclusion . . . . .	40
<b>Conclusion</b>	<b>41</b>

# List of Figures

1.1	difference between DL and ML . . . . .	7
1.2	pine and Feet X-RAY (left and right) . . . . .	11
1.3	MRI scans showing axial and sagittal views of the brain . . . . .	11
1.4	Computed Tomography of heart for pneumonia . . . . .	12
1.5	Fundamental algorithms generally used in US imaging analysis. . . . .	12
1.6	Semantic Image Segmentation . . . . .	13
1.7	Input and output of information from neurons . . . . .	14
1.8	A sigmoid function. . . . .	14
1.9	Rectified Linear Unit function . . . . .	15
1.10	A sketch of a biological neuron . . . . .	16
1.11	Hidden units in a Multi-layer Perceptron . . . . .	16
1.12	Convolutional Neural Network . . . . .	17
1.13	Convolutional Layer . . . . .	17
1.14	Illustrating the first 5 steps of convolution operation . . . . .	18
1.15	An example showing some of the initial steps as well as the final output of the Maximum Aggregation process . . . . .	19
1.16	The architecture of Fully Connected Layers . . . . .	19
1.17	LeNet-5 Architecture . . . . .	22
1.18	VGGNet Architecture . . . . .	22
1.19	The architecture of Generative Adversarial Networks . . . . .	23
2.1	General outline for detecting brain tumors. . . . .	26
2.2	Diagram showing the classification of brain tumors by CNN and GAN . .	27
2.3	Examples of Glioma Tumor MRI Images Used in Training . . . . .	29
2.4	Demonstrates how to combine a CNN model and a GAN model. . . . .	31
3.1	Comparison between healthy and tumor MRI images . . . . .	32
3.2	Accuracy and Confusion Matrix of CNN model . . . . .	36
3.3	Loss of CNN model . . . . .	37
3.4	Comparing the quality and accuracy of images of brain tumors for stages A, B, and C. . . . .	37
3.5	Accuracy-loss curve analysis of the three phases . . . . .	38

3.6 Confusion matrices across the three training phases . . . . . 39

# List of Tables

2.1	Table showing the CNN structure in MRI image classification . . . . .	28
2.2	The generator structure shows the output forms and the number of parameters. . . . .	29
2.3	The Discriminator structure shows the output forms and the number of parameters. . . . .	30
3.1	Distribution of MRI images across training, validation, and test sets. . .	33
3.2	Confusion matrix values . . . . .	40
3.3	Summary of final values . . . . .	40

## List of Abbreviations:

<b>DL</b>	Deep Learning
<b>AI</b>	Artificial intelligence
<b>MRI</b>	Magnetic resonance imagin
<b>CT</b>	Computerized tomography
<b>ECT</b>	Emission Computed Tomography
<b>ML</b>	Machine learning
<b>ANN</b>	Artificial Neural Network
<b>CNN</b>	Convolutional Neural Network
<b>US</b>	Ultrasound
<b>MAE</b>	Mean Absolute Error
<b>MSE</b>	Mean Squared Error
<b>GAN</b>	Generative Adversarial Networks
<b>FLOPs</b>	Floating Point Operations

# General Introduction

## 1 Introduction

Artificial intelligence is a branch of computer science capable of analysing complex medical data[1]. Their potential to exploit meaningful relationship within a data set can be used in the diagnosis, treatment and predicting outcome in many clinical scenarios. The expression “Medical Technology” is widely used to address a range of tools that can enable health professionals to provide patients and society with a better quality of life by performing early diagnosis, reducing complications[2]. Among AI methodologies, Machine Learning (ML) and Deep Learning (DL) have emerged as two key subfields that drive innovation in the healthcare sector. ML algorithms are particularly effective in extracting hidden patterns from large-scale healthcare data, while DL techniques - especially deep neural networks (DNN) - have demonstrated superior performance in image-based pattern recognition tasks. These models are able to handle complex data structures, making them highly suitable for applications such as image classification, segmentation, fusion, and registration [3].

The growing availability of electronic health records and the advancement of big data technologies have further enabled the successful deployment of ML and DL techniques in real-world clinical environments. One particularly notable area of application is Magnetic Resonance Imaging (MRI), which offers excellent soft tissue contrast and has benefitted substantially from the application of deep learning methods to improve diagnostic precision and efficiency [4].

There are many aspects of deep learning that could be helpful in health care, such as its superior performance, end-to-end learning scheme with integrated feature learning, capability of handling complex and multi-modality data and so on. To accelerate these efforts, the deep learning research field as a whole must address several challenges relating to the characteristics of health care data (i.e. sparse, noisy, heterogeneous, time-dependent) as need for improved methods and tools that enable deep learning to interface with health care information workflows[5].

## 2 Problematic

Brain-related diseases, particularly tumors, are among the most complex conditions that require an accurate and timely diagnosis due to their direct impact on the patient's life. Magnetic Resonance Imaging (MRI) is one of the most essential tools used to detect these tumors. However, analyzing MRI scans demands advanced medical expertise and may be subject to variability among specialists or delays in urgent cases. With the rapid advancement in artificial intelligence technologies, it has become possible to develop intelligent systems capable of analyzing medical images and extracting indicators of pathological changes or abnormalities. Accordingly, this thesis aims to explore the potential of deep learning techniques to build a model capable of accurately analyzing MRI images and identifying cases that show signs of brain tumors. This approach contributes to supporting diagnostic efforts, enhancing the accuracy of early detection, and reducing reliance on manual interpretation. Nevertheless, one of the key challenges in developing such models is the limited availability of labeled medical images, especially for less common tumor types. To address this, image generation techniques such as Generative Adversarial Networks (GANs) can be employed to artificially expand the training dataset by creating realistic MRI scans. This augmentation process helps improve the model's generalization capability, reduces overfitting, and ultimately enhances detection performance, particularly when data is scarce or imbalanced.

## 3 Motivation Behind the Thesis

The motivation behind this research lies in the urgent need to improve diagnostic accuracy and efficiency in healthcare systems, especially in the early detection of critical diseases like brain tumors. Traditional diagnostic methods often rely on manual interpretation of medical images, which can be time-consuming, subject to human error, and limited by the experience of the radiologist.

With the growing availability of MRI imaging data and the rapid advancements in deep learning, particularly in Convolutional Neural Networks (CNNs), there is a strong opportunity to enhance medical image analysis by developing automated, accurate, and reliable tools.

By applying deep learning techniques to brain MRI images, this study aims to:

- Support early and precise tumor detection, which is crucial for effective treatment and patient survival.
- Reduce the workload on medical professionals and minimize diagnostic errors.
- Contribute to the ongoing development of AI-powered healthcare solutions, making healthcare systems smarter, faster, and more accessible.

- Ultimately, this research seeks to bridge the gap between advanced AI technology and real-world medical practice, helping to build more efficient and intelligent healthcare infrastructures.

## 4 Contributions

Our contributions to the field of medical image classification, particularly in brain tumor detection from MRI scans , can be summarized as follows:

- We introduce a deep learning-based diagnostic system that combines Convolutional Neural Networks (CNNs) with Generative Adversarial Networks (GANs) to improve the detection of brain tumors in high-resolution MRI scans . The CNN serves as a robust image classifier capable of distinguishing between healthy and tumor-affected brain images with high accuracy.

- To enhance the diversity and quality of training data, we employ GANs to generate realistic synthetic MRI scans that simulate both healthy and abnormal cases. This augmentation addresses the challenge of data scarcity and helps improve the generalization ability of the CNN model, especially when dealing with rare or complex tumor patterns.

- We conduct a comparative evaluation between our CNN-GAN-based approach and traditional CNN models trained without synthetic data, demonstrating the superior performance of the hybrid system in terms of classification accuracy, sensitivity, and reliability.

- We validate our model on a curated dataset consisting of 5,000 multimodal MRI images, including 2,000 healthy and 3,000 tumor-labeled scans, sourced from credible medical imaging databases. The model achieves a high classification performance, indicating its potential for real-world deployment as a clinical decision-support tool.

Overall, this work highlights the practical value of integrating CNNs and GANs in medical imaging, offering a scalable and accurate solution to assist radiologists and improve diagnostic workflows in healthcare environments.

## 5 Structure of the Thesis

The rest of this thesis is structured as follows:

- In Chapter 1, explains the general background to the work, which includes general concepts such as medical image and Machine learning, computer vision, CNN, etc.
- In Chapter 2 ,We will begin by presenting a detailed overview of the baseline method, highlighting its limitations. This will be followed by a comprehensive description of our proposed methodology and the specific steps involved.

- In Chapter 3 ,We will describe the experimental setup and procedures, along with the evaluation metrics used. Subsequently, we will analyze and discuss the results obtained.
- Finally, We will draw the general conclusion of the thesis, highlighting the main outcomes and discussing potential future work and research directions.

# Chapter 1

## Work Background

### 1 Introduction

In the field of artificial intelligence, deep learning (DL) is a process that mimics the mechanism of human intelligence in processing big data and creating patterns to make decisions. Deep learning or neural network has been used in many fields such as computer vision, speech recognition and natural language processing, where it has greatly improved the accuracy of diagnosis, treatment planning and disease monitoring. In this chapter we explore the fields of artificial intelligence, computer vision, deep learning and medical image analysis and how they contribute significantly to improving the fields of healthcare systems.

### 2 Definition of Artificial Intelligence

Artificial Intelligence (AI) refers to the capability of machines to mimic human intelligence and is a key branch of computer science. It enables machines to learn from experience and perform tasks typically associated with human cognition, such as problem-solving, reasoning, and natural language understanding [6]. AI developments can be broadly classified into two main categories. The first includes methods and systems that simulate human experience and draw conclusions from a set of rules, such as expert systems.. The second includes systems that model the way the brain works, with artificial neural networks (ANNs) being a prime example [7].

### 3 Definition of Machine learning

Machine learning, an evolving branch of artificial intelligence, plays a vital role in enabling machines to gain human like intelligence without the need for explicit programming. Instead, algorithms learn to recognize patterns and make predictions from examples in the data. Machine learning involves applying statistical models to data using computers

and uses a wider range of statistical methods than those typically used in fields such as medicine. Advanced techniques, such as deep learning, rely on models that make fewer assumptions about the data, allowing them to process and learn from more complex datasets [8].

## 4 Definition of Deep learning

Deep Learning (DL) is a subset of Machine Learning (ML) that provides great flexibility and learning power by representing the world as concepts with a nested hierarchy, where these concepts are defined in simpler terms and a more abstract representation that reflects less abstract concepts. In the case of face recognition, dark or light areas must be identified first before geometric primitives such as lines and shapes.[9] Each node refers to an aspect of the entire network, providing a complete representation of the image when grouped together. Each node has a weight that reflects the strength of its association with the output. The weights are then adjusted as the model is developed. The benefit of DL is that it is supported by massive amounts of data. This opens up more opportunities for innovation[9].

## 5 Types of learning

### 5.1 Learning problems

#### **Supervised learning**

is the search for algorithms that infer from externally supplied instances to produce generalized hypotheses, which then make predictions about future instances. The goal is to build a summary model of the distribution of class labels in terms of predictive attributes [10].

#### **Unsupervised machine learning**

is a machine learning training method for statistical analysis. Its main goal is to discover hidden properties inherent in a dataset by calculating the commonalities between unlabeled samples. To avoid the labeling issue in supervised learning, it is often used as a powerful tool for label-free analysis or irrelevant applications, and has significant advantages in real-time data analysis [11].

#### **Reinforcement learning**

This type of learning is a set of challenges in which users must learn to utilize feedback to take action in a specific context. Although similar to supervised learning, reinforcement learning has delayed feedback and a noisy system because it looks for difficult responses

and models to link causality. Examples of reinforcement learning algorithms include temporal contrast, deep reinforcement [9].

## 6 The difference between ML and DL

Machine learning and deep learning are both branches of artificial intelligence. Machine learning focuses on developing algorithms that can adapt and improve automatically without human intervention, by using structured data to produce the desired outcomes. In contrast, deep learning is a specialized subset of machine learning that leverages neural networks to learn autonomously from unstructured or unlabeled data. Machine learning applies algorithms to analyze data, extract patterns, and make informed decisions based on what it has learned. Deep learning, however, structures these algorithms into multiple layers, forming artificial neural networks capable of learning and making decisions independently. These layered models process data step by step, with each layer contributing a more refined level of analysis [3].

This image shows the difference between Machine learning and Deep learning [9].

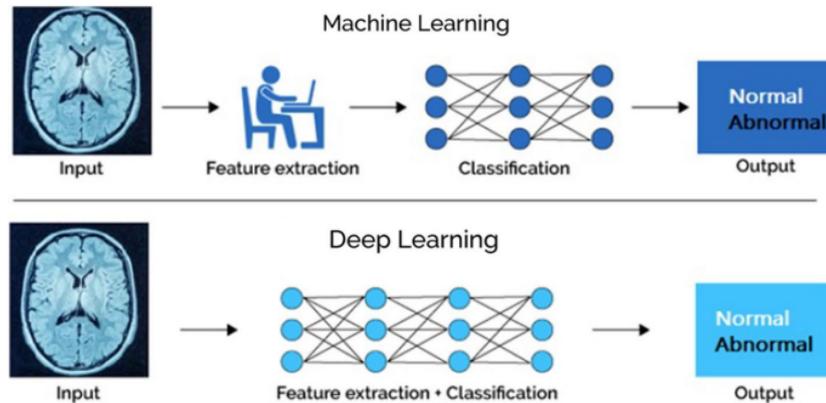


Figure 1.1: The difference between DL and ML[9]

## 7 Computer Vision

Computer Vision is a field focused on understanding and describing the world through visual data by interpreting, reconstructing, and extracting meaningful features from images. It involves the development of artificial systems capable of addressing visual tasks of interest, utilizing various techniques from image processing and analysis to achieve this goal [12].

## **8 Applications of Computer Vision**

### **8.1 Computer Vision for Fabric Defect Detection**

Computer vision-based models are increasingly utilized across various applied domains, particularly in solving industrial problems. In the textile industry, traditional inspection methods that rely on manual human effort are no longer sufficient or efficient to meet current production demands. To improve product quality and reduce production costs, there is a growing need to adopt industrial automation in the inspection process. Computer vision offers a powerful foundation to bridge this gap by enabling accurate, fast, and automated inspection systems, which significantly enhance both efficiency and reliability in textile manufacturing [13].

### **8.2 Computer Vision for Healthcare research**

CV has many potential uses in the healthcare sector; scientific research and technological advancements indicate this. This will have a significant positive impact on both patients and medical institutions. The amount of data about healthcare is growing exponentially. It has excellent potential to be mined and analysed to provide better, individualised treatment, lower medical mistakes, and allow early illness diagnosis. Medical image analysis, predictive analysis, health monitoring, and many more CV applications are used in healthcare to help medical professionals diagnose patients more accurately. CV application in the healthcare industry helps advance medical research, enhance patient care, and enhance medical procedures and treatments. It helps medical practitioners make better choices about patient care [14].

### **8.3 Computer Vision for Dynamic Intrusion Surveillance and Threat Analysis**

Computer vision techniques play a pivotal role in the development of intelligent surveillance systems by analyzing live video feeds in real-time, enabling rapid and accurate detection of unauthorized individuals or activities. These techniques rely on a range of advanced algorithms such as motion detection and tracking, as well as face recognition, to enhance the accuracy and efficiency of detection processes and distinguish between different elements within security scenes. The application of these algorithms demonstrates a high capability to handle diverse security scenarios, thereby improving the reliability and performance of dynamic surveillance and threat analysis systems [15].

### **8.4 Computer Vision for Analyse medical images**

Computer vision (CV) plays a pivotal role in medical image analysis, particularly in the detection and classification of tumors. Initially applied to brain tumor identification,

recent advancements have expanded its capabilities to detect lung tumors and other malignancies with improved sensitivity and accuracy. By integrating machine learning (ML) algorithms, CV systems can analyze CT, MRI, and biopsy images more quickly and precisely than human clinicians, significantly enhancing cancer screening and early diagnosis.

CV mimics the human brain's visual recognition ability by training on large datasets using pattern recognition and deep learning techniques. These models have proven effective in dermatology, identifying malignant skin developments through the examination of sequential photos or videos. In ophthalmology, CV is increasingly used to detect clinically relevant features for both diagnostic and predictive purposes.

Moreover, healthcare CV helps reduce the diagnostic workload by enabling even less-experienced practitioners to objectively and efficiently interpret medical images. Deep convolutional neural networks (CNNs) are widely applied in radiology, particularly for classifying benign and malignant nodules in lung CT scans. Thanks to these capabilities, CV has become a critical tool in modern medicine, enhancing diagnostic accuracy, accelerating treatment decisions, and supporting more personalized patient care [14].

## **8.5 Computer Vision for Remote health monitoring**

Remote health monitoring scenarios, self-service accessibility, and other benefits of medical automation for patients are made possible by the numerous CV medical applications. The most important thing to remember is that CV applications in medical contexts save lives and lessen medical procedures' severity, trauma, and expense. Using CV in treatment planning helps clinicians better comprehend the impact of different treatment modalities by comparing photos taken before and after the therapy on patients. Doctors can watch patients remotely and promptly treat patients, particularly the elderly, using computer vision. Home-based patient monitoring and rehabilitation are examples of CV applications in healthcare, where clinicians digitally watch patients' vitals and follow their progress [14].

## **8.6 Computer Vision for Reduce treatment time**

This system reads images in real time while assisting in image processing to detect particular sickness symptoms. When a CV is appropriately used in medicine, it may help reduce the time wasted on pointless diagnostic tests and provide medical professionals with the tools they need to order more precise diagnoses and successful treatments. CV has developed to the point where it can now assist with various medical duties and other functions. As a result, an increasing number of professionals may now benefit from a range of benefits that these breakthroughs have brought about [14].

## 8.7 Computer Vision for Lowering diagnostic mistakes

Using CV in healthcare applications benefits both clinicians and patients. By offering an opinion on diagnostic findings and spotting the minor irregularities and deviations from the norm that doctors may miss during manual observations, CV mainly assists clinicians in lowering the number of diagnostic mistakes and false positives. To preserve a patient's life, physicians must diagnose certain illnesses, including cancer, in the early stages. Both false-negative and false-positive findings have the potential to be very damaging. In such circumstances, a patient may either delay initiating therapy or base essential choices on their understanding of a supposedly nonexistent ailment. In specific ways, CV eliminates the risk of human mistakes and helps radiologists. It may assist clinicians in the early detection of a variety of diseases. Healthcare practitioners may visualise data in 3D, which is more informative than 2D using computer vision, and they can decide on patient treatment with knowledge. CV is also a practical answer to the shortage of radiologists and MRI technicians in the healthcare industry [14].

## 9 Medical image

Medical imaging is an essential part of disease diagnosis and treatment and represents various imaging modalities. These include X-rays, computed tomography (CT), and magnetic resonance imaging (MRI). These modalities play a vital role in revealing anatomical and functional information about various body organs for diagnostic and research purposes. Medical imaging is also an essential aid in modern healthcare systems. Machine learning plays a vital role in CADx with its applications in tumor segmentation, cancer detection, classification, and image processing [16].

### 9.1 X-rays

X-rays are a form of electromagnetic radiation, A definite value of wavelength, amplitude, frequency and velocity are the main advantage of electromagnetic waves. X-rays are a medical imaging technique, used to identify hard tissues and bone structures. In an X-Ray system the patient is exposed to X-ray radiation (which have frequency in the range of  $10^{16}$ -  $10^{19}$  Hz and wavelength in the range of nanometers  $10^9$ ) and the reflection is made to be produced on 2-dimensional projection radiographs. The X-ray images are a 2-D projection of the internal body organs in different shades of black and white owing to the varied amount of absorption of radiations by bones and tissues [17].



Figure 1.2: spine and Feet X-Ray (left and right) [17]

## 9.2 Magnetic resonance imaging (MRI)

MRI is basically a non-invasive technology which produces detailed anatomical images without the damaging radiation as is the case in X-Ray radiography. In magnetic resonance imaging technology, the subject under study is subjected to strong magnetic field align the protons in the body according to that field. Basically, the protons in the body spin out of equilibrium and begin to strain under the pull of magnetic force when radio frequency current is pulsed through the human body [17].

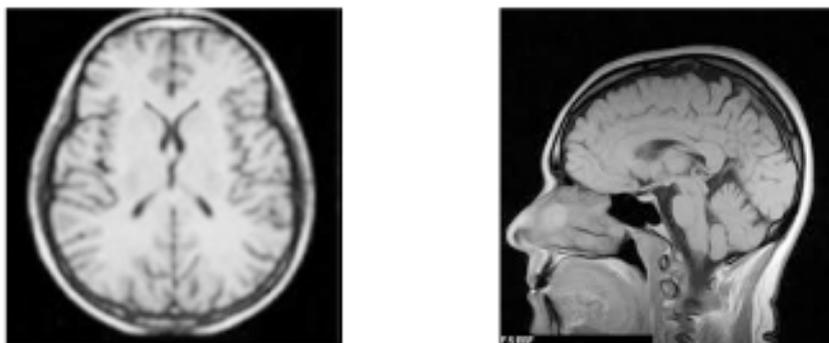


Figure 1.3: MRI scans showing axial and sagittal views of the brain (left and right) [17].

## 9.3 Computerized tomography (CT scan)

Computed tomography, also known as computed tomography, was introduced as a clinical application in 1971 as a development of X-ray imaging technology that was limited to axial images of the body. CT has become an essential tool in patient treatment and follow-up programs and is widely used in radiation therapy procedures [17].

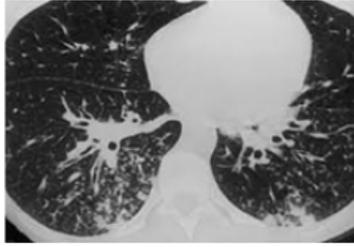


Figure 1.4: Computed Tomography of heart for pneumonia [17]

## 9.4 Ultrasound (US)

A typical medical ultrasound system consists of a transmitting probe, which emits a pulsed signal with frequencies ranging from 20 kilohertz to several gigahertz, and a receiving display. This facilitates the production of real-time images of living tissues within the body. Sound waves propagate through the body and reach soft tissue, fluid, or bone. These waves are reflected off flat surfaces within the body and recorded and displayed on a computer. Ultrasound is used diagnostically to image internal body structures, such as joints, tendons, ovaries, uterus, tendons, and muscles, and most importantly in obstetrics. It is also used to visualize temporal changes in tissue, blood location, cyst presence, and tissue stiffness and movement by exploiting the acoustic impedance of tissue [17].

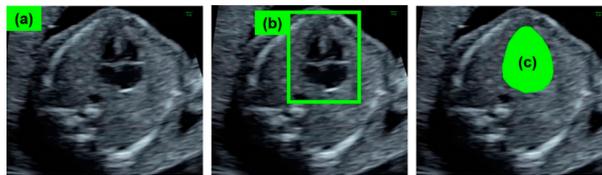


Figure 1.5: Fundamental algorithms generally used in US imaging analysis. (a) Image classification of whether the fetal US image contains a diagnostically useful cross-section such . (b) Detection of the fetal heart for evaluation of fetal heart structure. (c) Segmentation of the boundaries or regions of the fetal heart to measure the fetal cardiac index such as cardiothoracic area ratio (CTAR) [18]

## 9.5 Emission Computed Tomography

SPECT (Single Photon Emission Computed Tomography) and PET (Positron Emission Tomography) are two medical imaging techniques that fall under the umbrella of Emission Computed Tomography (ECT). These imaging methods utilize radioactive substances to examine various physiological properties and patterns within the body. In SPECT, the term "single" refers to the emission of a single photon resulting from radioactive decay, whereas in PET, the decay results in the emission of a single positron. ECT operates on the tracer principle, which posits that the physiological processes of the body can be assessed by tracking the gamma rays emitted from these radioactive materials. PET, as a form of nuclear medicine imaging, allows for the visualization of various functional and metabolic processes occurring within the body [17].

## 10 Image segmentation

It is a specific image processing technique used to divide an image into two or more meaningful regions. Image segmentation can also be considered the process of identifying boundaries between separate semantic entities in an image. From a technical perspective, image segmentation is the process of assigning a label to each pixel in an image, such that pixels with the same label are associated with specific visual or semantic properties [19].



Figure 1.6: Semantic Image Segmentation [19]

## 11 Artificial Neural Network (ANN)

ANNs are artificial adaptive systems that are inspired by the functioning processes of the human brain. They are systems that are able to modify their internal structure in relation to a function objective. They are particularly suited for solving problems of the nonlinear type, being able to reconstruct the fuzzy rules that govern the optimal solution for these problems.

- The basic elements of an artificial neural network are the nodes, called processing elements (PE), and the connections. Each node has its own inputs, from which it receives communications from other nodes, and each node has a function that converts its global inputs into outputs [20].
- In mathematical terms, the function of a neuron is expressed as follows:

$$v_k = \sum_{j=1}^m w_{kj} x_j \quad (1.1)$$

$$y_k = \varphi(v_k + b_k) \quad (1.2)$$

- A function that receives an input signal and produces an output signal after a certain threshold value is called an activation function. However, artificial neural

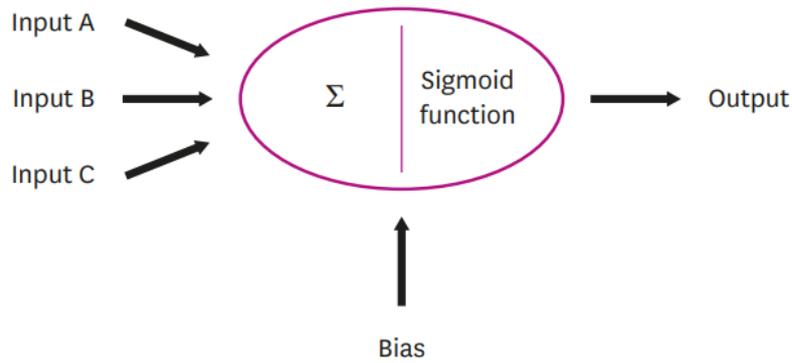


Figure 1.7: Input and output of information from neurons [21]

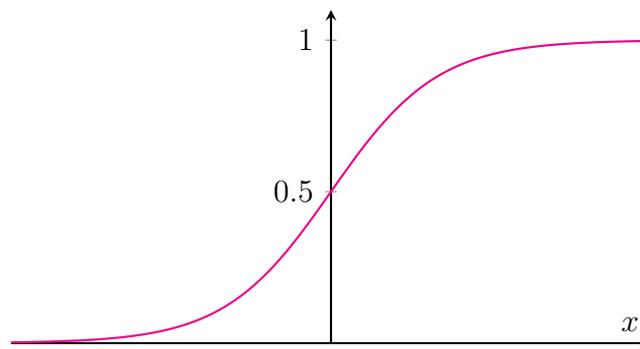


Figure 1.8: A sigmoid function.

networks use different functions other than activation functions, and most of them use sigmoid functions, which are also called logistic functions. The sigmoid function is easy to calculate compared to other functions and is expressed by the following equation.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1.3)$$

- The hyperbolic tangent (tanh) activation function is another commonly used nonlinear activation function in neural networks [22].

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1.4)$$

- Rectified linear unit function (ReLU) provides a very simple nonlinear transformation [22]:

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (1.5)$$

- We also have loss functions, used to evaluate the performance of the model between

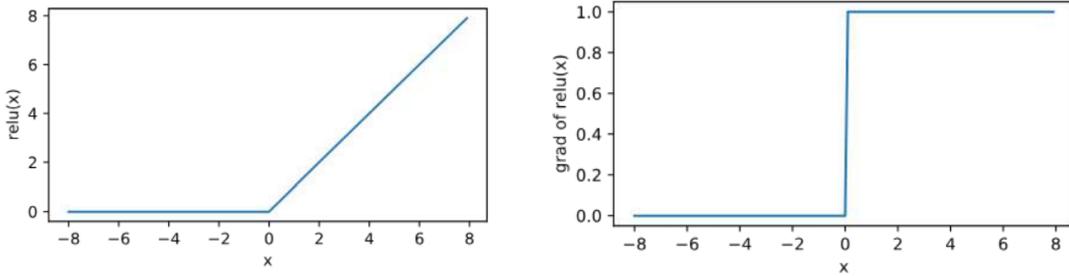


Figure 1.9: Rectified Linear Unit function [22]

expected values and actual values, best known as MSE and MAE:

- **Mean Absolute Error:** Absolute Error, also known as L1 loss, is the absolute difference between a predicted value and the actual value.

$$L1 = |y_{\text{actual}} - y_{\text{predicted}}| \quad (1.6)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (1.7)$$

Here, N is the number of data samples,  $y_i$  is the true value, and  $\hat{y}_i$  is the predicted value. Mean Absolute Error is a simple yet robust loss function used in regression models. [23]

- **Mean Squared Error:** Squared Error Loss, also known as L2 loss, is the squared difference between the prediction and actual values.

$$L2 = (y_{\text{actual}} - y_{\text{predicted}})^2 \quad (1.8)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (1.9)$$

Here, N is the number of data samples,  $y_i$  is the true value, and  $\hat{y}_i$  is the predicted value. MSE is also referred to as a quadratic loss because the penalty is squared rather than directly proportional to the error. The outliers are given more weight when the error is squared, creating a smooth gradient for minor errors[23].

## 11.1 The biological inspiration for ANN

A neuron (or nerve cell) is a special biological cell that processes information . It is composed of a cell body, or soma, and two types of out-reaching tree-like branches: the axon and the dendrites. The cell body has a nucleus that contains information

about hereditary traits and a plasma that holds the molecular equipment for producing material needed by the neuron. A neuron receives signals (impulses) from other neurons through its dendrites (receivers) and transmits signals generated by its cell body along the axon (transmitter), which eventually branches into strands and substrands. At the terminals of these strands are the synapses. A synapse is an elementary structure and functional unit between two neurons (an axon strand of one neuron and a dendrite of another). When the impulse reaches the synapse's terminal, certain chemicals called neurotransmitters are released. The neurotransmitters diffuse across the synaptic gap, to enhance or inhibit, depending on the type of the synapse, the receptor neuron's own tendency to emit electrical impulses [24].

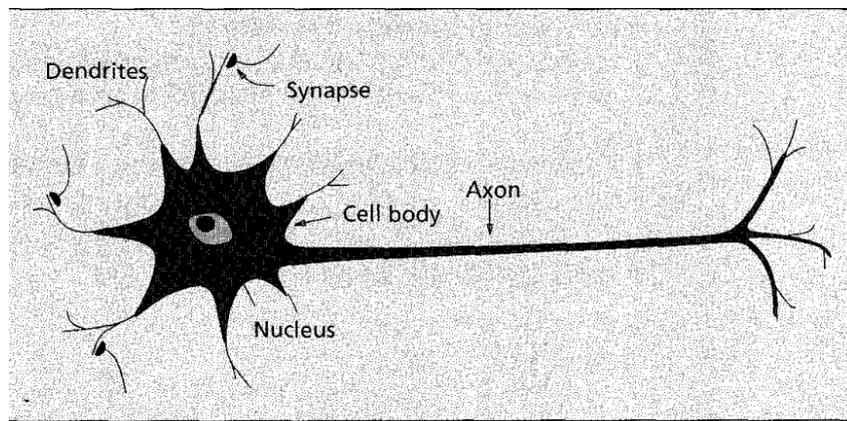


Figure 1.10: A sketch of a biological neuron [24]

## 11.2 Architecture of Artificial Neural Network

The addition of hidden units allows the network to develop complex feature detectors:

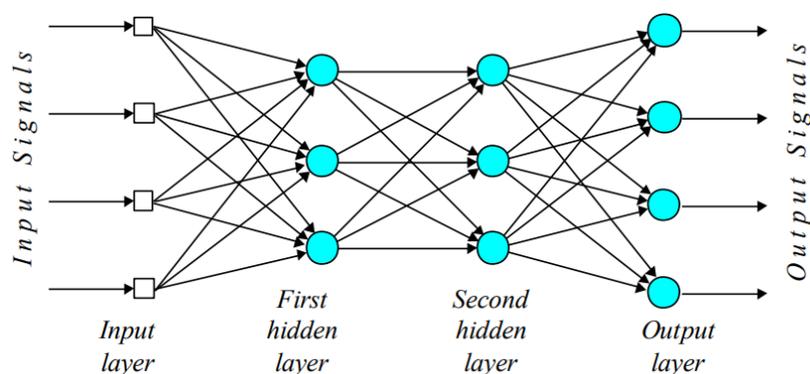


Figure 1.11: Hidden units in a Multi-layer Perceptron [22]

## 12 Convolutional Neural Network (CNN)

### 12.1 Definition of Convolutional Neural Network

In deep learning, Convolutional Neural Networks (CNNs), also known as ConvNets, represent a prominent class of neural networks that are frequently employed for analyzing visual data. CNNs are widely used in the field of computer vision due to their ability to automatically and efficiently extract features from images. The name 'convolutional' is derived from the structure of their hidden layers, which typically include convolutional layers, pooling layers, and fully connected layers. CNNs are applied to a variety of tasks, including image classification, visual data processing, and regression-based predictions.

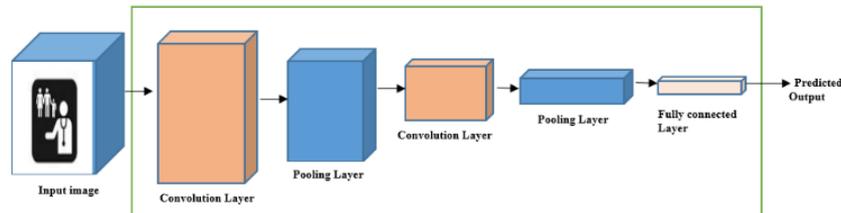


Figure 1.12: Convolutional Neural Network [25]

### 12.2 Convolutional layer

The first convolutional layer is a key component in any CNN architecture. It includes a set of convolutional filters (also known as kernels) that are applied to the input image (represented as N-dimensional matrices) through a convolution operation to produce an output feature map [26].

Activation map = Input \* Filter

$$\text{Output}(p, q) = \sum_{y=0}^{\text{columns}-1} \sum_{x=0}^{\text{rows}-1} \text{Input}(p-x, q-y) \cdot \text{Filter}(x, y) \quad (1.10)$$

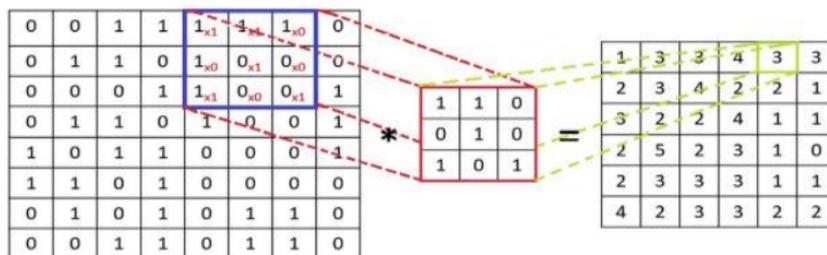


Figure 1.13: Convolutional Layer [27]

## Convolution Operation

To understand the convolution operation, consider a grayscale image with dimensions  $4 \times 4$  and a  $2 \times 2$  kernel initialized with random weights. During the convolution process, the  $2 \times 2$  kernel is slid across the entire  $4 \times 4$  image both horizontally and vertically. At each position, the corresponding values of the kernel and the image patch are multiplied element-wise, and the results are summed to produce a single scalar value in the output feature map. This operation is repeated until the kernel has moved over all valid positions of the image and can no longer slide further [26].

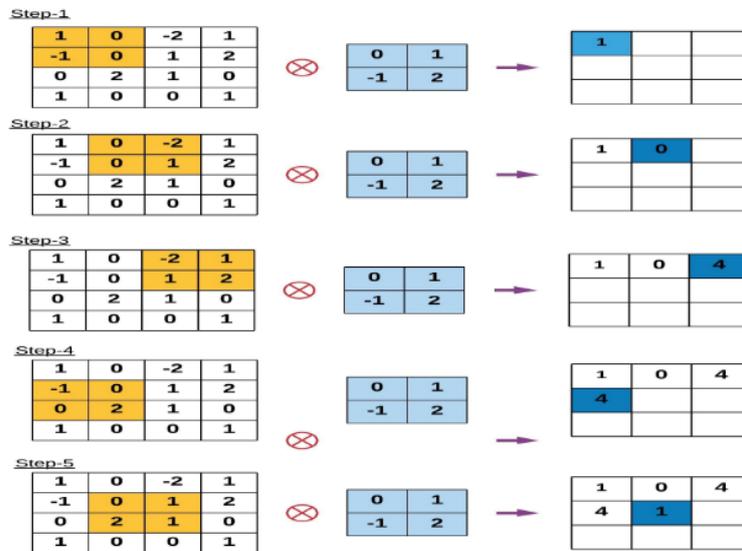


Figure 1.14: Illustrating the first 5 steps of convolution operation [26]

## 12.3 Pooling layer

The pooling layer plays a crucial role in reducing the spatial dimensions of the convolved features, thereby decreasing the computational complexity required for training. This dimensionality reduction not only enhances efficiency but also facilitates the extraction of dominant features that are invariant to position and rotation, thus supporting robust and efficient model training [25]. The pooling layer processes each activation map in the input and reduces its dimensionality by applying the "MAX" function. This technique is commonly used in most convolutional neural networks (CNNs) [28].

### Max Pooling

Max Pooling is a technique that selects the maximum value from each submatrix within the activation map and constructs a new matrix from these values. This process helps reduce the number of learnable features, thereby simplifying the model, while still preserving the most important and distinctive features of the original image. Max Pooling

is usually done using a 2x2 filter [29].

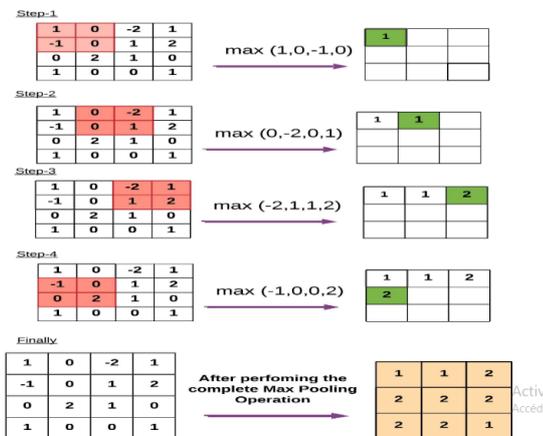


Figure 1.15: An example showing some of the initial steps as well as the final output of the Maximum Aggregation process[26]

## 12.4 Fully connected layer

The final part or layers of any Convolutional Neural Network (CNN) designed for classification typically consists of fully connected (FC) layers. In these layers, each neuron is connected to every neuron in the previous layer. The last fully connected layer functions as the output layer, serving as the classifier for the CNN. Fully connected layers are a type of feed forward artificial neural network (ANN) and are based on the principles of the traditional multi-layer perceptron (MLP) architecture. These layers receive their input from the last convolutional or pooling layer in the form of feature maps. These feature maps are flattened into a one dimensional vector, which is then passed through the fully connected layers to produce the final output of the CNN [26].

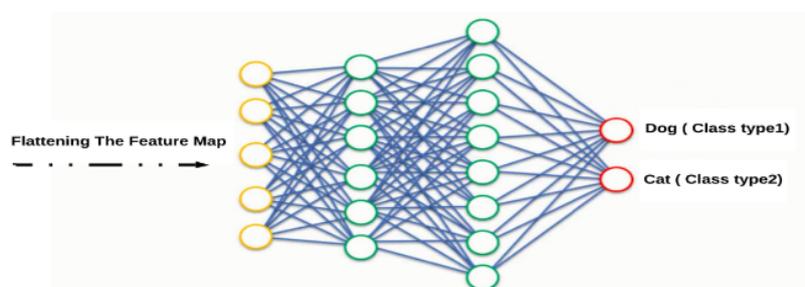


Figure 1.16: The architecture of Fully Connected Layers [27]

## 13 CNNs Work

### 13.1 Input (The Training Data)

The input layer in a neural network is typically represented in three dimensions: width, height, and depth. This is often written as  $\text{width} \times \text{height} \times \text{depth}$ , where the input is a matrix of pixel values. For instance, an input of  $(64 \times 64 \times 3)$  means the image has:

- **Width:** 64 pixels.
- **Height:** 64 pixels.
- **Depth:** 3, which refers to the color channels (Red, Green, Blue) in an RGB image.

Usually, the input layer dimensions are even numbers, allowing them to be repeatedly divided by 2 during operations like pooling or convolution without loss of structure.

### 13.2 Filter

Also called kernels or feature detectors, filters are small matrices used for feature detection. Each kernel is a grid of discrete values, where each value represents a weight. At the start of CNN training, these weights are usually initialized randomly, though other initialization methods may be applied [26]. During training, over multiple epochs, the weights are gradually updated, enabling the kernel to learn and extract meaningful features from the input data [29].

### 13.3 Convolved Feature

The Convolved Feature, also known as the Feature Map or Activation Map, is the output generated when a filter moves across the input image. At each position, the filter computes the dot product with the corresponding submatrix of the image, resulting in an output volume that captures the presence of specific features [29].

### 13.4 Receptive Field

The Receptive Field refers to the specific region of the input matrix that matches the size of the filter. This is the portion of the input that the filter interacts with at each step during the convolution process [29].

### 13.5 Depth

The number of filters [29].

### 13.6 Depth Column

Also known as a fiber, it is a cluster of neurons that all respond to the same receptive field [29].

## 13.7 Stride

Stride determines how far the filter moves after each convolution step. A larger stride results in a smaller output volume. For example, with a stride of 2, the filter shifts by 2 columns after each convolution. It is important to choose the stride so that the output volume remains an integer. Smaller stride values are typically used to achieve more precise results [29].

## 13.8 Zero Padding

Zero padding involves adding zeros around the edges of the input volume to preserve the spatial dimensions of both the input and output volumes. Without zero padding, information at the borders of the image may be lost, and the dimensions of the volume will shrink, which can reduce performance [29].

## 13.9 Parameter Sharing

When a specific feature appears frequently across many images, it is efficient to use parameter sharing to detect that feature in all images. However, this practice is not very common. For example, in face detection, if the face always appears in the same position in each image, there may be no need to repeatedly search for facial features[29].

## 13.10 Dilation

Dilation refers to filters that contain gaps between their cells. For example, consider a one-dimensional filter  $W$  of size 3 and an input  $X$ :

- With a dilation rate of 0:

$$w[0] \times x[0] + w[1] \times x[1] + w[2] \times x[2]$$

- With a dilation rate of 1:

$$w[0] \times x[0] + w[1] \times x[2] + w[2] \times x[4]$$

This spacing allows the filter to cover a wider area without increasing its size [29].

## 13.11 ReLU Layer

The ReLU layer applies an activation function element-wise that transforms all negative values to zero, effectively setting the threshold at zero. This operation does not change the volume or affect any hyperparameters [29].

## 14 Popular CNN Architectures

### 14.1 LeNet

Currently, the development of LeNet has reached the LeNet-5 version. This version is a gradientbased CNN learning structure and was first introduced for digital handwriting character recognition [27].

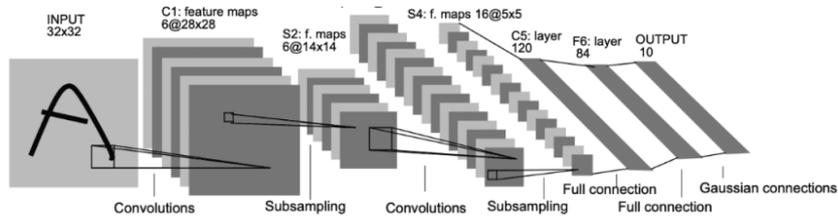


Figure 1.17: LeNet-5 Architecture [27]

### 14.2 AlexNet

This architecture is one of CNN architectures with a basic, simple, yet effective layer design. AlexNet has five convolutional layers, followed by a pooling layer on its fourth layer and three layers of a fully connected layer on its fifth. In AlexNet architecture, the convolutional kernels are extracted during the back-propagation optimization procedure by optimizing with the stochastic gradient function [29].

### 14.3 VGGNet

The latest version of VGGNet is VGGnet-16. This architecture employs 13 convolutional layers and 3 fully connected layers. The size of the convolutional layer in VGG16 is  $3 \times 3$  with an aggregation and padding layer of size 1. Meanwhile, the size of the aggregation layer is  $2 \times 2$  with a step size of 2. The resolution of the input image in VGG-16 is  $224 \times 224$  [27].

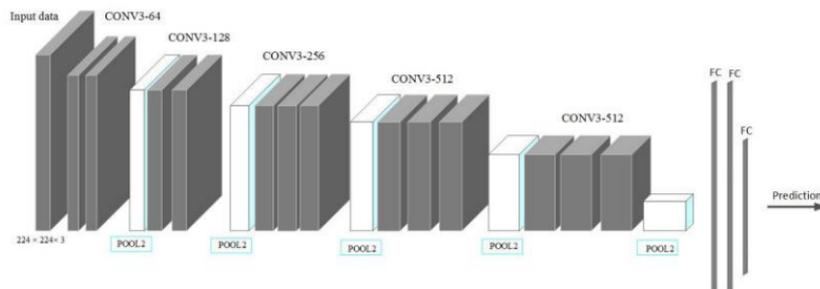


Figure 1.18: VGGNet Architecture [27]

## 15 Generative Adversarial Networks (GANs)

### 15.1 What Is The Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a type of artificial intelligence algorithm developed to address the problem of generative modeling. The primary objective of a generative model is to analyze a set of training data and learn the underlying probability distribution that produced those examples. Once trained, GANs can generate new, similar examples by sampling from the learned distribution [30]. While deep learning-based generative models are widely used, GANs stand out as some of the most effective, particularly for generating highly realistic, high-resolution images. They have been applied to a broad range of tasks, primarily in research contexts. However, GANs still pose unique challenges and open research questions, as they rely on game-theoretic principles, in contrast to most other generative models that are based on standard optimization techniques [30].

### 15.2 How Does The GAN Work

GANs are inspired by game theory, where the generator and the discriminator compete with each other during the training process in an attempt to reach a Nash equilibrium. The principle of generator  $G$  is to generate fake data as much as possible to fit the

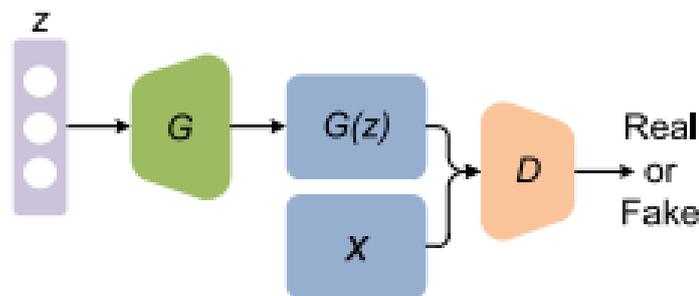


Figure 1.19: The architecture of Generative Adversarial Networks[31]

potential distribution of real data, while the principle of discriminator  $D$  is to correctly distinguish real data from fake data. The input of the generator is a random noise vector  $z$  (usually a uniform or normal distribution). The noise is mapped to a new data space via generator  $G$  to obtain a fake sample,  $G(z)$ , which is a multi-dimensional vector. And, the discriminator  $D$  is a binary classifier, it takes both the real sample from dataset, and the fake sample generated by generator  $G$  as the input, and the output of discriminator  $D$  represents the probability that the sample is a real rather than a fake. When the discriminator  $D$  cannot determine whether the data comes from the real dataset or the generator, the optimal state is reached. At this point, we obtain a generator model  $G$ ,

which has learned the distribution of real data [31]. The generator and the discriminator can be viewed as two competing players, each with its own objective function, denoted as  $J(G)$  and  $J(D)$ , respectively. The discriminator  $D$ , acting as a binary classifier, is trained using a cross-entropy loss function defined as:

$$J(D) = -\frac{1}{2} \mathbb{E}_{x \sim p_{data}} [\log D(x)] - \frac{1}{2} \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (1.11)$$

Here,  $x$  represents a real data sample drawn from the true data distribution  $p_{data}$ , and  $z$  is a noise vector sampled from a prior distribution  $p_z$  (typically uniform or normal). The generator maps  $z$  to a synthetic data sample  $G(z)$ , which is then evaluated by the discriminator.

The discriminator's objective is to maximize the probability of correctly identifying real samples and minimizing the probability of misclassifying fake samples generated by  $G$ . Hence, it seeks to make  $D(x) \rightarrow 1$  and  $D(G(z)) \rightarrow 0$ . Conversely, the generator aims to deceive the discriminator by generating samples that are indistinguishable from real data, thus driving  $D(G(z)) \rightarrow 1$ .

This adversarial relationship between the generator and discriminator forms a **zero-sum game**, where the improvement of one network leads to a loss for the other. Consequently, the generator's loss function can be defined as the negative of the discriminator's loss [31]:

$$J(G) = -J(D) \quad (1.12)$$

## 16 Conclusion

In conclusion, this chapter has explored an overview of some basic and important concepts, including deep learning, neural networks, and computer vision techniques in the field of medical image analysis and classification to diagnose medical conditions, and we have highlighted the importance of these techniques in improving healthcare systems. Our goal was to provide some details about the background of the work, making it easy to understand and clearly contextualized.

# Chapter 2

## Proposed Method

### 1 Introduction

The increasing global burden of brain tumors has intensified the need for accurate and efficient diagnostic systems. Magnetic Resonance Imaging (MRI) plays a crucial role in brain tumor detection due to its high-resolution imaging capabilities. However, manual interpretation of MRI scans remains time-consuming and prone to human error, especially in regions with limited access to experienced radiologists. To address these challenges, researchers have turned to artificial intelligence, particularly deep learning, to automate and improve the diagnostic process.

This chapter presents a comprehensive approach to brain tumor classification using convolutional neural networks (CNNs) and Generative Adversarial Networks (GANs). It begins by describing the structure and distribution of a multimodal MRI dataset used for training, validation, and testing. It then outlines the proposed classification pipeline, including preprocessing techniques and the architecture of a lightweight CNN designed for binary classification (tumor vs. no tumor). Furthermore, the chapter introduces the integration of GANs for synthetic data generation, enhancing the dataset and mitigating overfitting challenges. This combination of CNN and GAN aims to increase the generalizability of the model and improve the classification accuracy on limited medical datasets.

### 2 An Overview on The Proposed Method

Brain tumors cause a high mortality rate around the world. Therefore, the automatic classification of brain tumors has attracted a lot of attention from researchers. Designing a system capable of accurately and effectively identifying tumors without human intervention is a major challenge. Figure 2.1 provides a general description of the classification of brain tumors. Figure 2.1 illustrates the main steps, which are the pre-processing and segmentation of medical MRI images, the extraction and selection of the most relevant

features, and finally the application of machine learning algorithms.

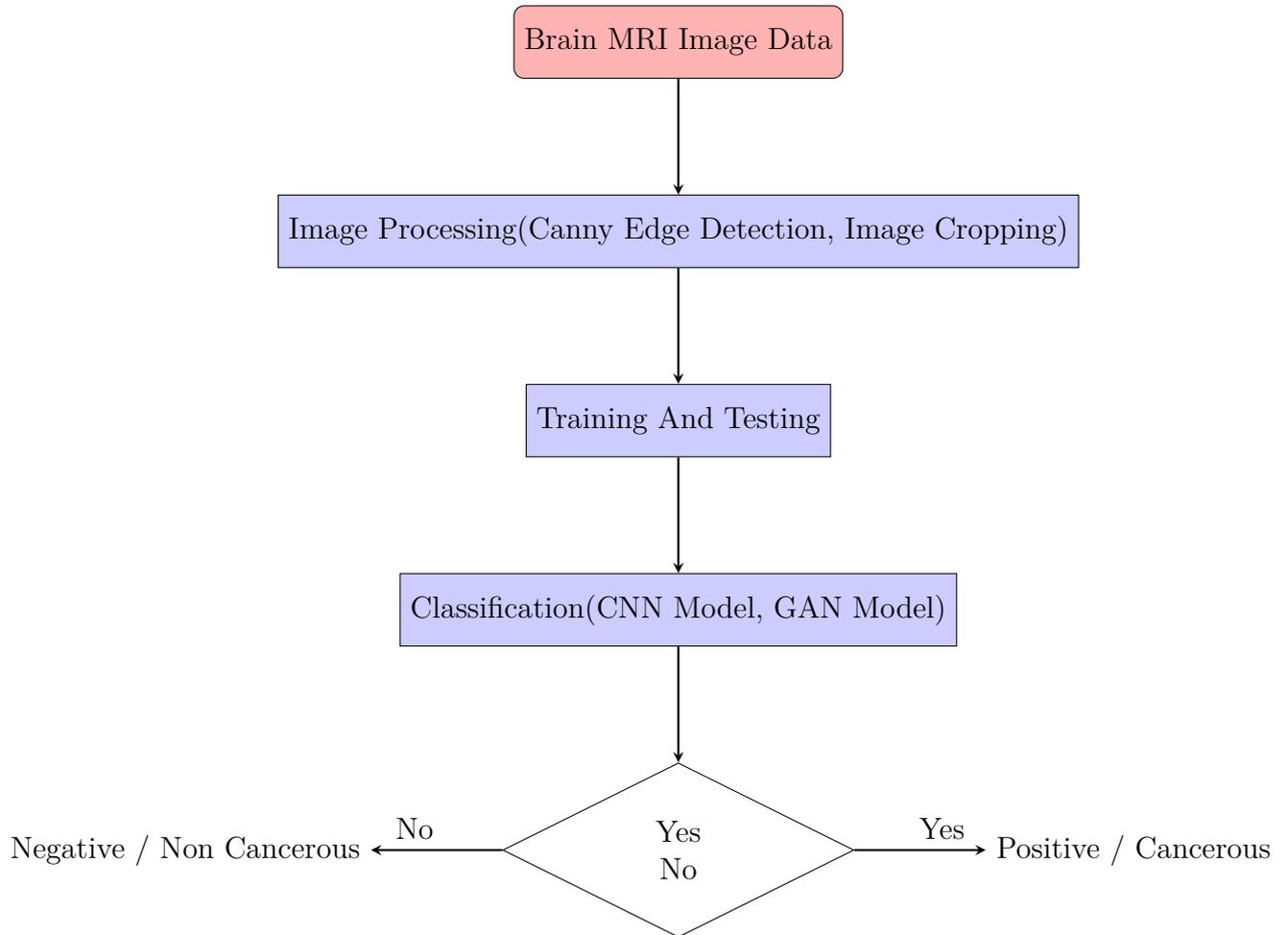


Figure 2.1: General outline for detecting brain tumors.

The proposed work includes an automatic detection process and a diagnosis model to predict brain tumors on MRI images. The first step involves acquiring data in the form of brain MRI images from the Kaggle dataset. Each image is then processed individually using various pre-processing techniques, namely resizing, cropping, Subsequently, the pre-processed images were processed with a training and testing phase with pre-trained CNN models and GAN to predict brain disease.

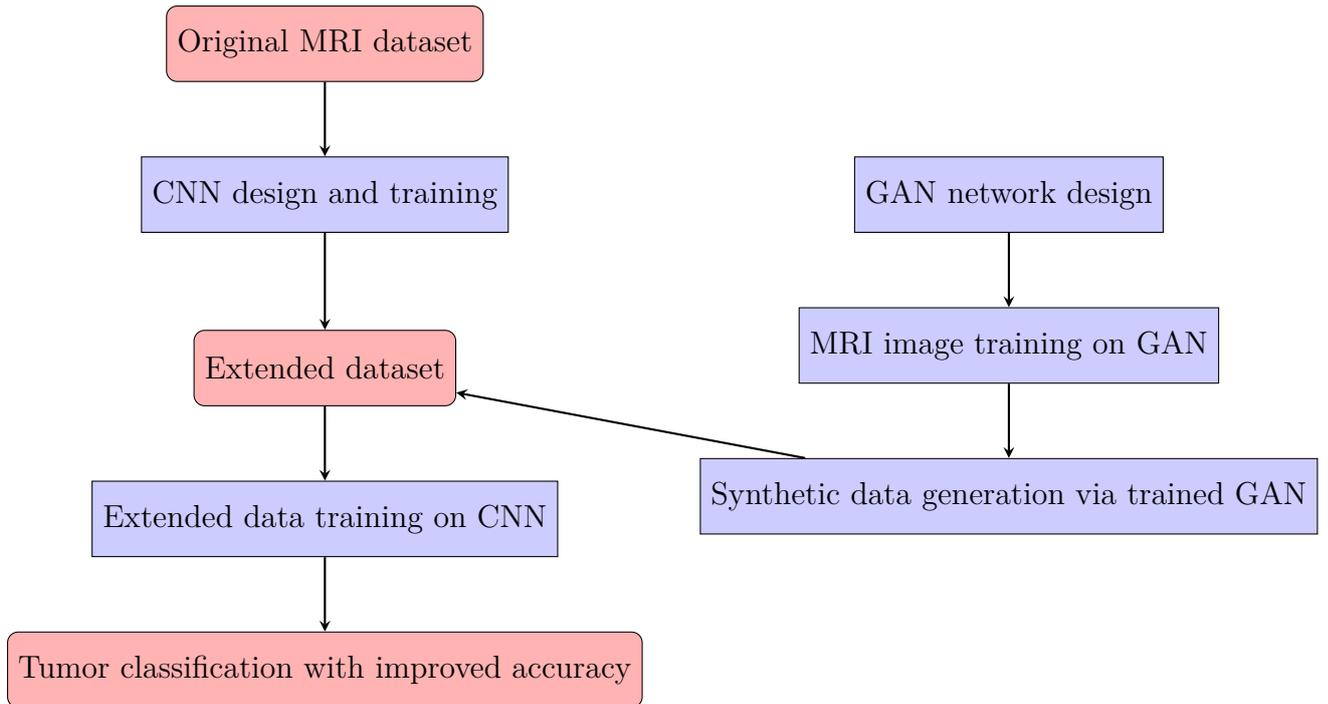


Figure 2.2: Diagram showing the classification of brain tumors by CNN and GAN

### 3 The CNN Model

The proposed method starts with the design of a lightweight convolutional neural network (CNN) to classify MRI images: tumor or no tumor. The network architecture consists of three convolutional layers followed by maximally dense clustering layers, providing a balance between performance and computational efficiency. Some of the basic structures used in CNNs are as follows.

- Convolutional layers (Conva 2D): Three layers with each layer containing 32 filters of size  $(3 \times 3)$  with a ReLU activation function were used to add nonlinearity.
  - Layers MaxPooling2D: Used to reduce the dimensions of an image to reduce the number of coefficients and improve efficiency by taking maximum values from small areas  $(2 \times 2)$ .
  - Layer Flatten: To convert the data from the convolutional layers into a flat vector form that can be fed into the layers Dense.
  - Two layers Dense: The Fully Connected Layer uses two layers, the first layer contains 32 neural units that receive the input from the previous layer, while the second layer contains two units dedicated to binary classification, such as “tumor” and “no tumor”.
- Activation function: The ReLU activation function encourages the model to learn quickly and avoids the problem of vanishing gradients. The output layer does not

Layer (type)	Output Shape	Param #
conv2d_15 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_10 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_16 (Conv2D)	(None, 32, 32, 32)	9,248
max_pooling2d_11 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_17 (Conv2D)	(None, 16, 16, 32)	9,248
flatten_5 (Flatten)	(None, 8192)	0
dense_10 (Dense)	(None, 32)	262,176
dense_11 (Dense)	(None, 2)	66

Table 2.1: Table showing the CNN structure in MRI image classification

contain activations (logits) because it is used with SparseCategoricalCrossentropy(from\_logits=True).

- Kernel size: A 3×3 pixel filter kernel in each convolutional layer.
- Stride: The default step when moving is 1 in each direction.
- Padding 'same': Maintains image dimensions after wrapping by appropriate padding.
- Loss function: Suitable for multi-class classification (here 2 classes), without activating softmax in the output layer.
- Learning rate: Default update rate for the Adam algorithm.
- Epochs: We used 10 epochs to train the model.

- This miniaturized design was chosen to minimize overlearning and reduce training time, while maintaining a robust feature extraction capability, especially when dealing with small medical datasets.
- In this project, we designed a new CNN model to achieve a balance between accuracy, computational efficiency, and training speed, rather than using existing CNN networks for several reasons:
  - Lightweight: The designed model contains far fewer layers and parameters.
  - Simplicity of the task: Since the classification task is binary (tumor/healthy) and the images are small in size (64×64), there is no need for deep and complex models to achieve good performance.
  - Designing the model from scratch gives us complete control over the architecture, making it easy to modify layers and filters to suit the nature of the data.
 Pre-trained models are usually huge and can lead to overfitting, especially when dealing with limited data.

## 4 The GAN Model

We used GAN to generate new images that simulate real MRI images, the model consists of two parts, the Generator and the Discriminator, we trained the generator to

produce images with dimensions of  $128 \times 128$  pixels (gray images) based on the real image data, the generator is randomly oriented from (latent space) and passes through a chain (Conv2DTranspose) that gradually scales the dimensions from  $256 \times 8 \times 8$  to the final image of  $128 \times 8 \times 1$ .

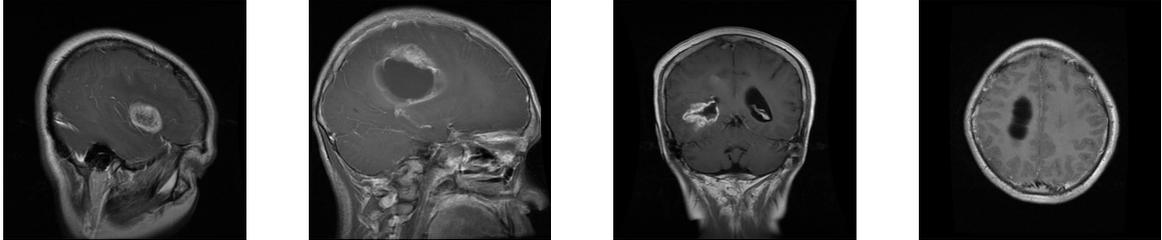


Figure 2.3: Examples of Glioma Tumor MRI Images Used in Training

The generator is randomized from latent space and passes through the Conv2DTranspose chain, which gradually enlarges the dimensions from  $256 \times 8 \times 8$  to the final  $128 \times 128 \times 1$  image, while the discriminator receives real or fake images to compare them, so that the generator can produce high-resolution images.

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 16384)	1,638,400
batch_normalization_30 (BatchNormalization)	(None, 16384)	65,536
leaky_re_lu_42 (LeakyReLU)	(None, 16384)	0
reshape_6 (Reshape)	(None, 8, 8, 256)	0
conv2d_transpose_24 (Conv2DTranspose)	(None, 16, 16, 128)	819,200
batch_normalization_31 (BatchNormalization)	(None, 16, 16, 128)	512
leaky_re_lu_43 (LeakyReLU)	(None, 16, 16, 128)	0
conv2d_transpose_25 (Conv2DTranspose)	(None, 32, 32, 64)	204,800
batch_normalization_32 (BatchNormalization)	(None, 32, 32, 64)	256
leaky_re_lu_44 (LeakyReLU)	(None, 32, 32, 64)	0
conv2d_transpose_26 (Conv2DTranspose)	(None, 64, 64, 32)	51,200
batch_normalization_33 (BatchNormalization)	(None, 64, 64, 32)	128
leaky_re_lu_45 (LeakyReLU)	(None, 64, 64, 32)	0
conv2d_transpose_27 (Conv2DTranspose)	(None, 128, 128, 16)	12,800
batch_normalization_34 (BatchNormalization)	(None, 128, 128, 16)	64
leaky_re_lu_46 (LeakyReLU)	(None, 128, 128, 16)	0
conv2d_18 (Conv2D)	(None, 128, 128, 1)	401

Table 2.2: The generator structure shows the output forms and the number of parameters.

- TheTable 2.3 shows a summary of the GAN discriminator architecture created using the Keras library. We note from the results that the highest value of the coefficients is in the last layer of Conva2D (819,456), due to the increase in the number of filters. The model also contains a total of 1,092,865 parameters, which is an appropriate number for this type of task.

Layer (type)	Output Shape	Param #
conv2d_19 (Conv2D)	(None, 64, 64, 32)	832
leaky_re_lu_47 (LeakyReLU)	(None, 64, 64, 32)	0
dropout_12 (Dropout)	(None, 64, 64, 32)	0
conv2d_20 (Conv2D)	(None, 32, 32, 64)	51 264
leaky_re_lu_48 (LeakyReLU)	(None, 32, 32, 64)	0
dropout_13 (Dropout)	(None, 32, 32, 64)	0
conv2d_21 (Conv2D)	(None, 16, 16, 128)	204 928
leaky_re_lu_49 (LeakyReLU)	(None, 16, 16, 128)	0
dropout_14 (Dropout)	(None, 16, 16, 128)	0
conv2d_22 (Conv2D)	(None, 8, 8, 256)	819 456
leaky_re_lu_50 (LeakyReLU)	(None, 8, 8, 256)	0
dropout_15 (Dropout)	(None, 8, 8, 256)	0
flatten_3 (Flatten)	(None, 16384)	0
dense_10 (Dense)	(None, 1)	16 385

Table 2.3: The Discriminator structure shows the output forms and the number of parameters.

## 5 Measure The Size of The Model using FLOPs

The number of floating operations represents the amount of calculations required to process an image captured during forward pass within the neural network. It was calculated manually using the following equation:

$$\text{FLOPs} = H_{\text{out}} \times W_{\text{out}} \times K_h \times K_w \times C_{\text{in}} \times C_{\text{out}} \times 2 \quad (2.1)$$

- Manual FLOPs calculation: 321,159,168.

This formula was applied to all Conv2D and Conv2DTranspose layers of the generator, ignoring the effects of Batche and LeakyReLU. where the calculation result for the generator FLOPs is approximately 2.7 billion calculations. The number of operations also expresses the number of learnable weights within the model, which is a direct measure of the model size in memory. These values were extracted using the `( )` model. `summray` function from the Keras library.

## 6 CNN-GAN hybrid scheme

First, we trained the GAN on yes images to generate similar synthetic images. Then, the generated images were used as part of the CNN training data by feeding them through a dynamic generator during training. Finally, the CNN was trained using the original data and the synthetic images generated by the GAN.

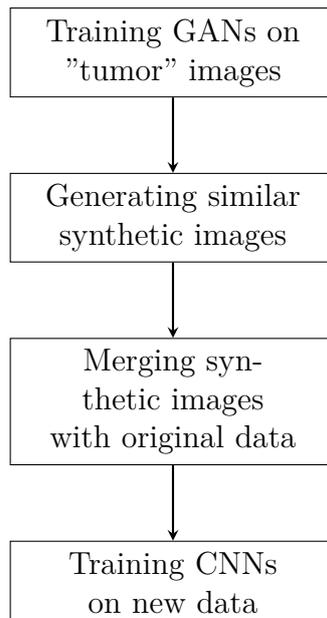


Figure 2.4: Demonstrates how to combine a CNN model and a GAN model.

## 7 Conclusion

In conclusion, this chapter outlined the dataset preparation, preprocessing techniques, and the architecture of the proposed models for brain tumor classification using MRI images. A structured approach was employed, beginning with data acquisition and distribution, followed by the development of a lightweight CNN for classification and the integration of a GAN model for data augmentation. These methods collectively aim to improve detection accuracy and generalization, especially when dealing with limited medical data.

# Chapter 3

## Experimental Results

### 1 Introduction

Building on the detailed presentation of our methodology and its implementation in the previous chapter, this chapter focuses on the experimental results. It delves into the key aspects of our approach and outlines the steps taken to obtain and analyze these results.

### 2 Experimental Setup

This section describes the details of the experimental environment, including the data set, development tools, and metrics used to evaluate performance.

#### 2.1 Dataset Description

This dataset contains a collection of multimodal medical images, specifically MRI for detecting brain tumors. It also includes high-resolution MRI images with 5000 images classified as healthy (no) 2000 images and tumor (yes) 3000 images, collected from different sources and optimized for maximum accuracy.

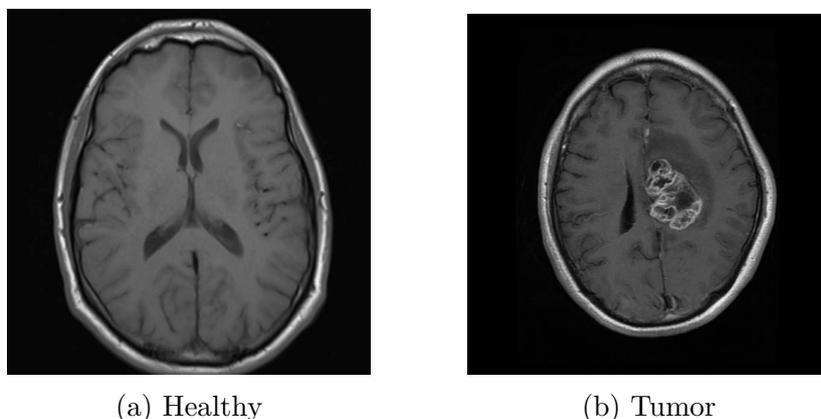


Figure 3.1: Comparison between healthy and tumor MRI images

- First, we divided the data consisting of two categories (yes, no) into three folders: Training, Test, and Valid, as shown in the table below:

<b>Class</b>	<b>Train</b>	<b>Validation</b>	<b>Test</b>
healthy (no)	1400	200	400
tumor (yes)	2400	300	600

Table 3.1: Distribution of MRI images across training, validation, and test sets.

- Second, we categorized the images into yes and no in each file in different proportions as follows:  
70% training,  
20% testing,  
10% validation.

## 2.2 Development Frameworks

The model was implemented using Python 3.10. We used [Keras/TensorFlow/PyTorch...etc] to build neural networks, along with libraries such as NumPy and Matplotlib...etc. for preprocessing, visualization, and evaluation. Experiments were conducted on Google Colab with GPU acceleration enabled.

### Google Colab

Google Colaboratory, often known as “Google Colab” or simply “Colab”, is a research-focused platform designed for building and testing machine learning models using powerful hardware like GPUs and TPUs. It offers a cloud-based, serverless Jupyter notebook environment that supports interactive coding and experimentation.

Just like other G Suite tools, Colab is free to use, making it a popular choice for students, researchers, and developers working on AI and data science projects[32].

### Python

Python is a modern and versatile programming language that supports multiple paradigms, including object-oriented, functional, and imperative styles. Its clear syntax and user-friendly design make it an excellent choice for beginners. Although primarily known as a scripting language, it also offers the ability to be compiled into machine-readable binaries. One of its biggest advantages is that it allows developers to write programs using significantly fewer lines of code compared to languages like C++, Java, or C [33].

### TensorFlow

is a library for high-performance numerical computations . It’s used across various scientific fields. TensorFlow is a framework for defining and running computations that

involve tensors, which are partially defined computational objects that eventually produce a value. Key features of TensorFlow are Better computational graph visualizations, reduces error by 50 to 60 percent in neural machine learning, , TensorFlow is particularly useful for Speech and image recognition, Text-based applications, Time-series analysis, and Video detection[34].

### **NumPy**

is the fundamental package for numerical computation in Python; it contains a powerful N-dimensional array object. NumPy also addresses the slowness problem partly by providing these multidimensional arrays as well as providing functions and operators that operate efficiently on these arrays. Key features of NumPy are Provides fast, precompiled functions for numerical routines, Array-oriented computing for better efficiency, Supports an object-oriented approach, and Compact and faster computations with vectorization. NumPy is particularly useful for Extensively used in data analysis, Creates a powerful N-dimensional array[34].

### **Pandas**

is a must in the data science life cycle. It is the most popular and widely used Python library for data science, along with NumPy in matplotlib. , it is heavily used for data analysis and cleaning. Pandas provide fast, flexible data structures, such as data frame CDs, which are designed to work with structured data very quickly and intuitively. Key features of Pandas are Eloquent syntax and rich functionalities that gives you the freedom to deal with missing data [34].

### **Matplotlib**

Matplotlib has powerful yet beautiful visualizations. It's a plotting library for Python , it's extensively used for data visualization. It also provides an object-oriented API, which can be used to embed those plots into applications. Key features of Matplotlib are Usable as a MATLAB replacement, with the advantage of being free and open-source, Supports dozens of backends and output types, which means you can use it regardless of which operating system you're using or which output format you wish to use [34].

### **Keras**

The Keras [35]library provides a high-level programming interface for building neural networks. It can work on top of TensorFlow, Microsoft Cognitive Toolkit (CNTK) or Theano [36] . The library is written entirely in Python and is distributed under the MIT license. Current version 2.3.1 The library is based on the following principles: ease of use, modularity,extensibility [37].

## PyTorch

PyTorch was launched in 2016 and revolves around the idea of keeping the API simple, allowing it to be quickly modified and kept up to date with the latest trends in the field of AI [38]. It does so by adhering to three main principles, of which the first is the method through which its functions are defined. The framework defines all its components in a pythonic way, the intent being to make it easily usable to users who are already familiar with the Python programming language. On the same topic of simplicity, the fact that components are defined through interfaces is the second principle of PyTorch. Through this, the complexity of defining and using components is masked by simple initializations and function calls, leading to a clearer identification of core neural networking concepts and therefore an easier learning curve[39].

## 2.3 Evaluation metrics

Evaluation metrics are essential tools for measuring the performance of machine learning models across various applications such as classification detection. Here, we discuss the main metrics used in classification tasks.

### Accuracy

It is a measure of the performance of models in classifying and segmenting data and is expressed mathematically as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (3.1)$$

It shows how well the model performs in predicting correctly and is used to compare models and indicates the model's ability to predict new data.

### Confusion Matrix

The confusion matrix [40] is typically used in machine learning to evaluate or to visualize the behavior of models in supervised classification contexts [41]. It is a square matrix in which the rows represent the actual class of the instances and the columns their predicted class.[42]

is divided into four values:

- True Positive (TP): The test result is positive in the presence of the clinical abnormality.
- False Positive (FP): The test result is positive in the absence of the clinical abnormality.

- True Negative (TN): The test result is negative in the absence of the clinical abnormality.
- False Negative (FN): The test result is negative in the presence of the clinical abnormality.

### Confusion Matrix-derived metrics

- Sensitivity (recall): Measures the model's ability to detect true positives.

$$\text{Sensitivity (Recall)} = \frac{TP}{TP + FN} \quad (3.2)$$

- Precision: Measures the accuracy of the model in predicting only positive cases.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.3)$$

## 3 Comparison And Results

### 3.1 Experiment 1: CNN Model

The Figure 3.2 shows the results of the CNN model after training it in the database and 10 epochs. Looking closely at the results shown in the graph, we notice that the training accuracy and validation accuracy increase significantly from 1-3 epochs, where the value approaches 0.99, while it stabilizes at 1.

Thus, the values that appeared to us indicate that the model results are excellent and highly accurate.

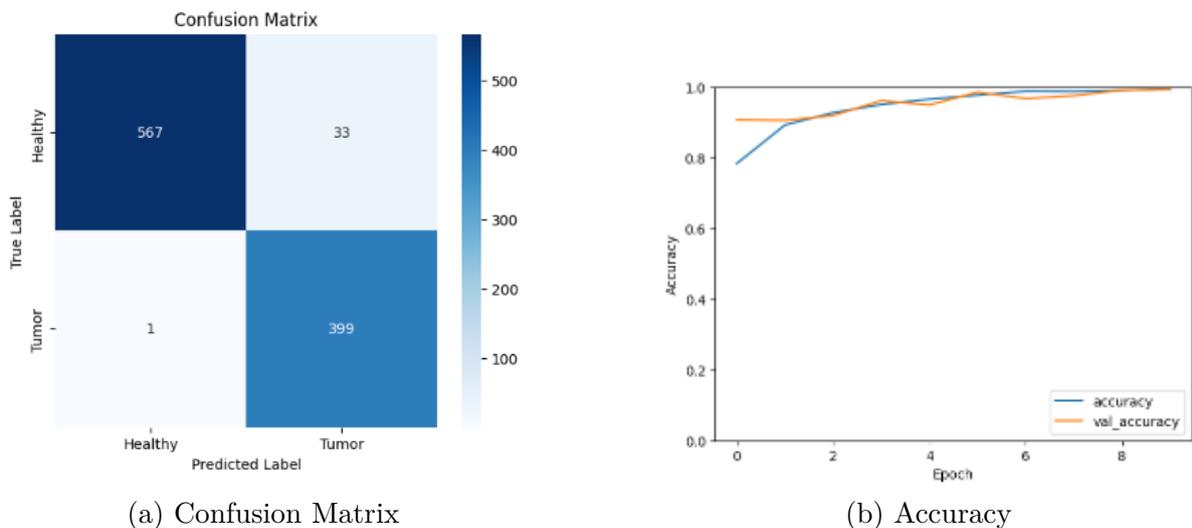


Figure 3.2: Accuracy and Confusion Matrix of CNN model

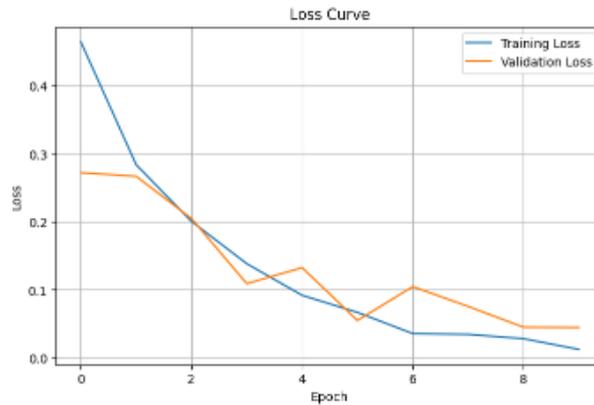
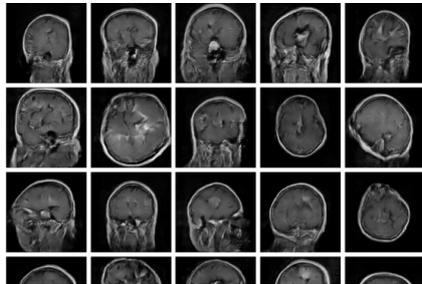


Figure 3.3: Loss of CNN model

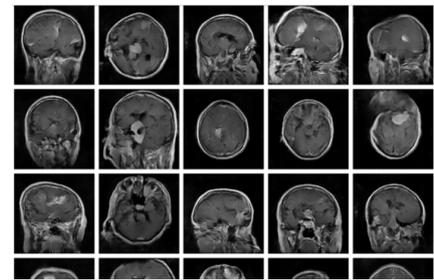
- The figure 3.3 represents the loss curve for the CNN model. We note that the blue line is below the orange line, which shows us that the model is learning well, as the loss clearly decreases across all data from the training and validation data.

### 3.2 Experiment 2: Generating images using results

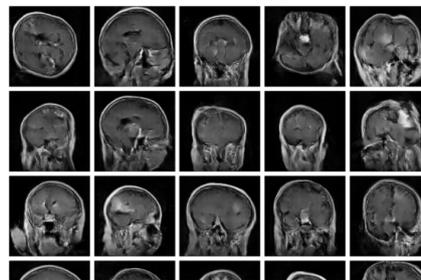
After we divided the GAN model into three stages (200 images, 500 images, 1000 images) and trained it for 200 epochs, we obtained the following results.



(a) 200 Images



(b) 500 Images



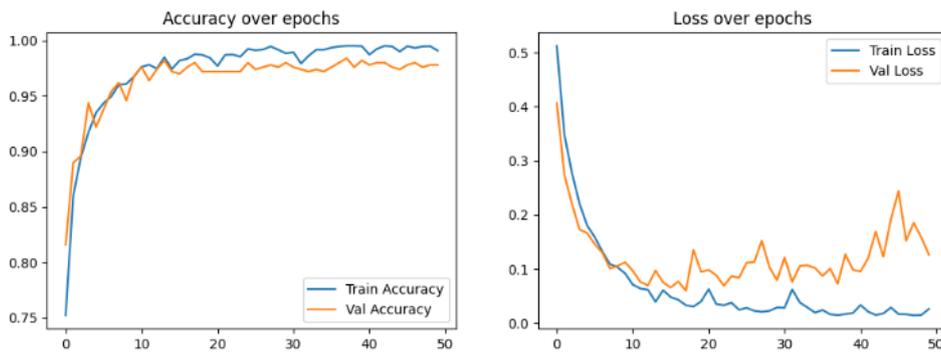
(c) 1000 Images

Figure 3.4: Comparing the quality and accuracy of images of brain tumors for stages A, B, and C.

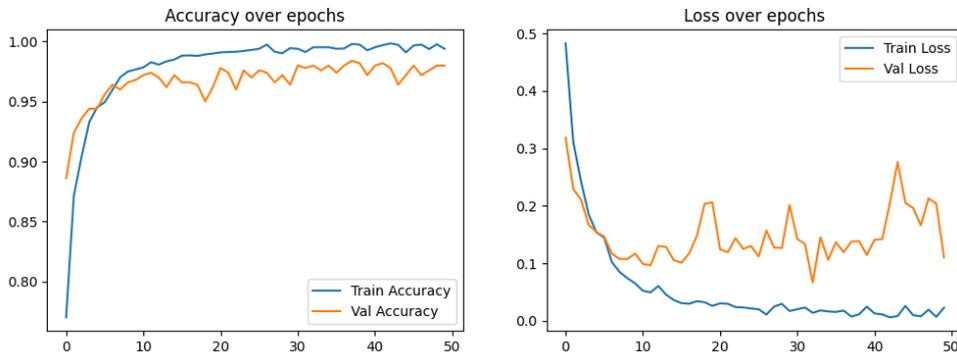
- The Figure 3.4 shows brain images for diagnosis in the event of a tumor using magnetic resonance imaging for the three stages. It shows us that image (a) is less

accurate and clear for the tumor, while image (b) is clearer and more accurate, and image (c) appears more widespread or less specific in some areas, Shown slides.

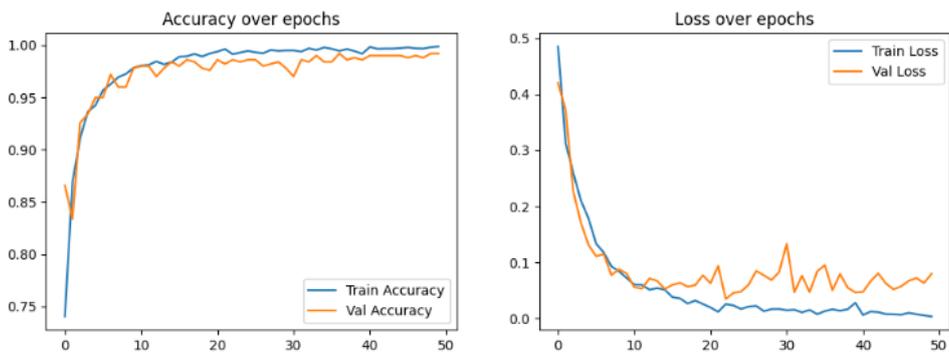
- The Figure 3.5 shows the accuracy and loss curves of a machine learning model or neural network over the course of 50 epochs for each stage. When comparing the three images, it becomes clear to us that there is a slight difference between the stages, and we notice that the Train Accuracy line, which shows the accuracy of the model on the training set, is increasing rapidly, unlike the Val Accuracy line, which appears to be slightly less than the accuracy of the training. Therefore, this graph indicates that the model is learning successfully. Its accuracy improves over time.



(a) 200 Images



(b) 500 Images



(c) 1000 Images

Figure 3.5: Accuracy-loss curve analysis of the three phases

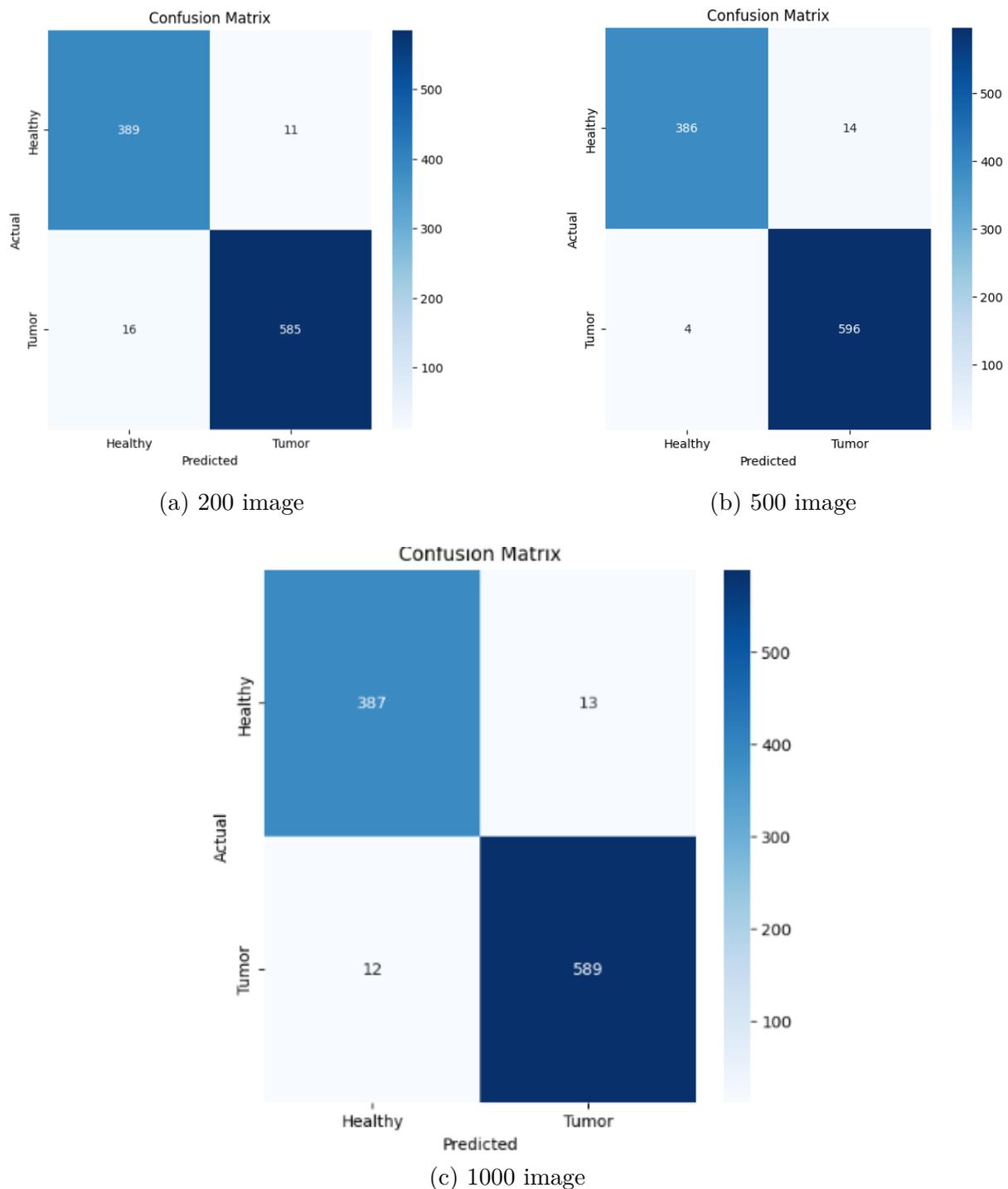


Figure 3.6: Confusion matrices across the three training phases

- The Figure 3.6 shows the confusion matrix for the binary classification model, classifying cases into "healthy" and "tumor." Based on the table 3.1, we note that the B image shows the least number of errors and accurately diagnoses the presence of a tumor well, even though the values are close between the stages.

Phases	TN	TP	FN	FP	Sensitivity	Precision
200 images	389	585	16	11	0.973	0.981
500 images	386	596	4	14	0.993	0.977
1000 images	387	589	12	11	0.980	0.981

Table 3.2: Confusion matrix values

### 3.3 Experiment 3: The CNN+GAN Model

The images generated in the previous experiment were used to enhance the “tumor” category in the training data. The CNN model was then retrained on the expanded data. The results showed a significant improvement compared to the baseline model.

Phases	Precision	Recall	F1-score	Accuracy	Loss
CNN	0.96	0.94	0.97	0.966	0.109
CNN+GAN	0.977	0.993	0.985	0.984	0.140

Table 3.3: Summary of final values

The table 3.3 shows a comparison of the performance of two CNN models and the combined CNN+GAN model across different stages, using Precision, Recall, F1-score, Accuracy, and Loss metrics. It is clear from the table 3.3 that most metrics showed significant improvement when the GAN model was added to the CNN model. This means that the combined model is more powerful and effective in the task being evaluated, reflecting a significant improvement in overall classification performance despite a slight increase in loss.

## 4 Conclusion

In this chapter, we present the implementation details of the proposed method, including the programming language and libraries used. We then report the experimental results obtained from testing on the dataset, accompanied by interpretations and insights to facilitate a better understanding of these results.

# Conclusion General

This thesis focused on developing a deep learning system for detecting brain tumors from MRI scans, aiming to enhance diagnostic accuracy beyond traditional methods. After an extensive review of existing techniques, we proposed a hybrid model that combines Convolutional Neural Networks (CNNs) with Generative Adversarial Networks (GANs) to improve the quality and

We detailed our methodology comprehensively, from data preprocessing to evaluation procedures. Testing our approach on a large, multimodal MRI dataset demonstrated that the CNN-GAN hybrid outperforms conventional CNN models in classification accuracy, sensitivity, and reliability.

The findings emphasize the practical benefits of integrating CNNs and GANs for medical image analysis, offering a scalable and efficient solution to support radiologists in early and accurate tumor detection while minimizing diagnostic errors. Overall, this study highlights the transformative potential of AI-driven methods in medical diagnostics and their role in improving patient care.

However, the study does have several limitations. One key limitation is the dependence on the quality and representativeness of the training data. Although the GAN component enhances data diversity, there remains a risk that generated images might introduce unrealistic features or artifacts. Moreover, the model was primarily trained and evaluated on a single dataset, raising concerns about its generalizability across different imaging devices or institutions. Additionally, the training process for GANs is computationally intensive and requires careful tuning, which could pose challenges for real-time or clinical deployment.

To address these limitations, future work could explore several directions. Expanding the dataset to include scans from multiple sources and diverse patient populations would improve model robustness. Further, integrating other data types such as clinical notes, histopathology images, or genomic data could enable a more comprehensive diagnostic framework. Future efforts could also incorporate explainability features to enhance trust and transparency in clinical settings. Finally, optimizing the model for deployment in resource-constrained environments would make the system more accessible and practical for widespread use.

# Bibliography

- [1] J Holmes, L Sacchi, R Bellazzi, et al. Artificial intelligence in medicine. *Ann R Coll Surg Engl*, 86:334–8, 2004.
- [2] Giovanni Briganti and Olivier Le Moine. Artificial intelligence in medicine: today and tomorrow. *Frontiers in medicine*, 7:509744, 2020.
- [3] Rashid Amin, Mohammed A Al Ghamdi, Sultan H Almotiri, Meshrif Alruily, et al. Healthcare techniques through deep learning: issues, challenges and opportunities. *IEEE Access*, 9:98523–98541, 2021.
- [4] Jin Liu, Yi Pan, Min Li, Ziyue Chen, Lu Tang, Chengqian Lu, and Jianxin Wang. Applications of deep learning to mri images: A survey. *Big Data Mining and Analytics*, 1(1):1–18, 2018.
- [5] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.
- [6] Dalvinder Singh Grewal. A critical conceptual analysis of definitions of artificial intelligence as applicable to computer engineering. *IOSR Journal of Computer Engineering*, 16(2):9–13, 2014.
- [7] S Agatonovic-Kustrin and Rosemary Beresford. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5):717–727, 2000.
- [8] Trishan Panch, Peter Szolovits, and Rifat Atun. Artificial intelligence, machine learning and health systems. *Journal of global health*, 8(2):020303, 2018.
- [9] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, Ahmed Shihab Albahri, Bashar Sami Nayyef Al-Dabbagh, Mohammed A Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H Al-Timemy, et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10(1):46, 2023.

- [10] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.
- [11] Bashar Rajoub. Supervised and unsupervised learning. In *Biomedical signal processing and artificial intelligence in healthcare*, pages 51–89. Elsevier, 2020.
- [12] Arthur Francisco Araújo Fernandes, João Ricardo Rebouças Dórea, and Guilherme Jordão de Magalhães Rosa. Image analysis and computer vision applications in animal sciences: an overview. *Frontiers in Veterinary Science*, 7:551269, 2020.
- [13] Aqsa Rasheed, Bushra Zafar, Amina Rasheed, Nouman Ali, Muhammad Sajid, Saadat Hanif Dar, Usman Habib, Tehmina Shehryar, and Muhammad Tariq Mahmood. Fabric defect detection using computer vision techniques: a comprehensive review. *Mathematical Problems in Engineering*, 2020(1):8189403, 2020.
- [14] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, and Mumtaz Ahmed. Computer vision to enhance healthcare domain: An overview of features, implementation, and opportunities. *Intelligent Pharmacy*, 2024.
- [15] Biresh Kumar, Shreya Kumari, Pallab Banerjee, Purushottam Kumar, Mohan Kumar Dehury, and Anita Kumari. Intruderwatch: Ai-powered computer vision for dynamic intrusion surveillance and threat analysis. In *2024 6th International Conference on Energy, Power and Environment (ICEPE)*, pages 1–7. IEEE, 2024.
- [16] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.
- [17] Bhawna Goyal, Sunil Agrawal, and BS Sohi. Noise issues prevailing in various types of medical images. *Biomedical & Pharmacology Journal*, 11(3):1227, 2018.
- [18] Masaaki Komatsu, Akira Sakai, Ai Dozen, Kanto Shozu, Suguru Yasutomi, Hidenori Machino, Ken Asada, Syuzo Kaneko, and Ryuji Hamamoto. Towards clinical application of artificial intelligence in ultrasound imaging. *Biomedicines*, 9(7):720, 2021.
- [19] Swarnendu Ghosh, Nibaran Das, Ishita Das, and Ujjwal Maulik. Understanding deep learning techniques for image segmentation. *ACM computing surveys (CSUR)*, 52(4):1–35, 2019.
- [20] Enzo Grossi and Massimo Buscema. Introduction to artificial neural networks. *European journal of gastroenterology & hepatology*, 19(12):1046–1054, 2007.

- 
- [21] Su-Hyun Han, Ko Woon Kim, SangYun Kim, and Young Chul Youn. Artificial neural network: understanding the basic concepts without mathematics. *Dementia and neurocognitive disorders*, 17(3):83–89, 2018.
- [22] John J Hopfield. Artificial neural networks. *IEEE Circuits and Devices Magazine*, 4(5):3–10, 1988.
- [23] Aryan Jadon, Avinash Patil, and Shruti Jadon. A comprehensive survey of regression-based loss functions for time series forecasting. In *International Conference on Data Management, Analytics & Innovation*, pages 117–147. Springer, 2024.
- [24] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.
- [25] Meenakshi Malik, Dr.Yashpal Singh, Puneet Garg, and Swati Gupta. Deep learning in healthcare system. *International Journal of Grid and Distributed Computing*, 13:469–468, 10 2020.
- [26] Anirudha Ghosh, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, and Debashis De. Fundamental concepts of convolutional neural network. *Recent trends and advances in artificial intelligence and Internet of Things*, pages 519–567, 2020.
- [27] Purwono Purwono, Alfian Ma’arif, Wahyu Rahmaniari, Haris Imam Karim Fathurrahman, Aufaclav Zatu Kusuma Frisky, and Qazi Mazhar ul Haq. Understanding of convolutional neural network (cnn): A review. *International Journal of Robotics and Control Systems*, 2(4):739–748, 2022.
- [28] Keiron O’shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
- [29] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, pages 1–5. IEEE, 2020.
- [30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [31] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent progress on generative adversarial networks (gans): A survey. *IEEE access*, 7:36322–36333, 2019.
- [32] Ekaba Bisong. Google colab. In *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pages 59–64. Springer, 2019.
-

- [33] Sloan Kelly and Sloan Kelly. What is python? *Python, PyGame and Raspberry Pi Game Development*, pages 3–5, 2016.
- [34] AL Sayeth Saabith, T Vinothraj, and M Fareez. Popular python libraries and their application domains. *International Journal of Advance Engineering and Research Development*, 7(11), 2020.
- [35] F Keras Chollet. Available online: <https://keras.io> (accessed on 14 august 2019).© 2019 by the authors. *Licens. MDPI, Basel, Switzerland. This Artic. is an open access Artic. Distrib. under terms Cond. Creat. Commons Attrib.(CC BY) Licens.(https://http//creativecommons.org/licenses/by/4.0/)*, 2015.
- [36] The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.
- [37] Migran N Gevorkyan, Anastasia V Demidova, Tatiana S Demidova, and Anton A Sobolev. Review and comparative analysis of machine learning libraries for machine learning. *Discrete and Continuous Models and Applied Computational Science*, 27(4):305–315, 2019.
- [38] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.
- [39] Ovidiu-Constantin Novac, Mihai Cristian Chirodea, Cornelia Mihaela Novac, Nicu Bizon, Mihai Oproescu, Ovidiu Petru Stan, and Cornelia Emilia Gordan. Analysis of the application efficiency of tensorflow and pytorch in convolutional neural network. *Sensors*, 22(22):8872, 2022.
- [40] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.
- [41] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. *An introduction to statistical learning*, volume 112. Springer, 2013.
- [42] Olivier Caelen. A bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, 81(3):429–450, 2017.