الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة عين تموشنت بلحاج بوشعيب
Belhadj Bouchaib University of Ain Temouchent
Faculty of Science and Technology
Department of Mathematics and Computer Science

Final Year Project

To obtain a Master's degree in :**CYSIA**

Field: **Mathematics and Computer Science**

Sector**: Computer Science**

Specialty: Cybersecurity and Artificial Intelligence

Theme:

Segmentation of Brain Tumors from Medical Images: An Artificial Intelligence Approach

Presented by:
Mrs. Taibi Fatima
Miss. Doukali Fatima Zahra

Before the jury composed of:

Dr Saidi Samira	UAT.B.B (Ain Témouchent) President
Dr Benomar Mohamed Lamine	UAT.B.B (Ain Témouchent) Examiner
Dr Bendiabdallah Mohammed Hakim	UAT.B.B (Ain Témouchent) Supervisor

Academic Year 2025/2026

Contents

List of Figures

List of Tables

Listings

Dedication

First and foremost, I express my deepest gratitude to God for granting me the strength, patience, and perseverance to successfully complete this work. Without His guidance and blessings, this achievement would not have been possible.

I lovingly dedicate this work to my beloved parents and my dear sister. To my parents, words cannot fully express my gratitude for your endless love, unwavering support, and countless sacrifices. You have always been my source of motivation and inspiration, encouraging me to pursue my dreams and never give up. To my sister, thank you for your constant support, understanding, and for always standing by my side through every challenge.

I also dedicate this work to my dear friend, whose encouragement, kindness, and presence have meant so much to me throughout this journey. Your support has made this path easier and more meaningful.

To all of you, I am deeply thankful. This achievement is as much yours as it is mine.

Fatima Zahra

Dedication

All praise is due to Allah, whose grace and guidance granted me the strength and perseverance to complete this work. Without His blessings, this achievement would not have been possible.

I extend my deepest gratitude to my beloved parents, whose unwavering support, prayers, and silent sacrifices have been my greatest source of strength. To my dear husband, thank you for your endless patience, trust, and encouragement; you were my pillar whenever I faltered and my peace when the journey grew heavy. Your steadfast support made this dream a reality

I also thank my dear friends, especially my companion throughout this research, who stood by me as both a sister and a mentor. Her wisdom and unwavering presence brought light to every challenge

Finally, to everyone who supported, prayed for, or encouraged me along this path: this achievement is as much yours as it is mine. Every word written here bears the mark of your kindness

TAIBI FATIMA

Acknowledgments

In the name of Allah, the Most Gracious, the Most Merciful.

All praise is due to Allah, by whose grace righteous deeds are accomplished, and peace and blessings be upon the most honorable of prophets and messengers, our Master Muhammad, as well as upon his family and all his companions.

At the conclusion of this humble work, we are pleased to express our sincere gratitude and deep appreciation to everyone who contributed to the success of this project and enabled us to reach this stage.

First, we extend our heartfelt thanks and profound gratitude to the administration of Ain Temouchent University for their efforts in providing a stimulating academic environment and for the valuable opportunities made available to us to learn and acquire knowledge, which have brought us to this level of scientific and professional maturity.

We express our deepest thanks and appreciation to our supervisor, Professor Mohamed Hakim Bendi Abdellah, for his trust, his valuable time, and the extensive knowledge he shared with us. Thank you, Professor, for being a pillar of support and encouragement from the very beginning, and for the enriching experience that enhanced our knowledge and contributed significantly to the completion of this work.

Our sincere thanks and appreciation go to all professors of the Mathematics and Computer Science Department, who contributed to our academic training and never hesitated to share their knowledge despite their busy schedules. They were always by our side, providing assistance and guidance whenever we needed it. May Allah reward you with the best of rewards.

Our heartfelt thanks also go to everyone who stood by our side, encouraged us, and supported us morally, even with a kind word or a sincere prayer. Every good word was a source of motivation for us to continue and give our best.

“And whoever does not thank people has not thanked Allah.”

All praise is due to Allah, Lord of all worlds, first and last.

Résumé

La segmentation automatique des tumeurs cérébrales à partir d'images par résonance magnétique (IRM) constitue un enjeu majeur en imagerie médicale, essentiel pour le diagnostic précoce, la planification thérapeutique et le suivi clinique. Face à la complexité et à la variabilité des structures tumorales, les méthodes manuelles s'avèrent chronophages et sujettes à la variabilité inter-observateur. Dans ce mémoire, nous proposons une approche basée sur l'apprentissage profond pour segmenter automatiquement les tumeurs cérébrales. Plusieurs architectures de pointe ont été implémentées et comparées : U-Net, Attention U-Net, DeepLabV3+, TransUNet et Vision Transformer (ViT).

Les expériences ont été menées sur un jeu de données public d'IRM cérébrales, prétraité et augmenté pour améliorer la généralisation. L'entraînement a utilisé une fonction de perte combinée (Dice + Entropie croisée binaire) et l'optimiseur Adam, avec des stratégies de régularisation (Early Stopping, réduction du taux d'apprentissage). Les résultats quantitatifs, évalués principalement via le coefficient Dice, montrent la supériorité de l'architecture DeepLabV3+ (Dice = 0,7840), suivie de l'U-Net classique (Dice = 0,7639) et du TransUNet (Dice = 0,7587). Le ViT pur a obtenu les performances les plus modestes, soulignant la nécessité de grands jeux de données pour les architectures basées sur l'attention. Ce travail démontre l'efficacité des modèles convolutifs et hybrides pour la segmentation médicale et ouvre la voie à des systèmes d'aide au diagnostic plus robustes.

Mots-clés : Segmentation sémantique, tumeurs cérébrales, IRM, apprentissage profond, U-Net, DeepLabV3+, TransUNet, coefficient Dice.

Abstract

Automatic brain tumor segmentation from Magnetic Resonance Imaging (MRI) scans represents a critical challenge in medical imaging, essential for early diagnosis, treatment planning, and clinical monitoring. Due to the complexity and heterogeneity of tumor structures, manual delineation is time-consuming and prone to inter-observer variability. This project proposes a Deep Learning-based approach for automatic brain tumor segmentation. Several state-of-the-art architectures were implemented and compared: U-Net, Attention U-Net, DeepLabV3+, TransUNet, and Vision Transformer (ViT).

Experiments were conducted on a public MRI dataset, with preprocessing and data augmentation to enhance model generalization. Training employed a combined loss function (Dice + Binary Cross-Entropy) and the Adam optimizer, complemented by regularization strategies such as Early Stopping and learning rate scheduling. Quantitative results, primarily evaluated using the Dice coefficient, demonstrate the superior performance of DeepLabV3+ (Dice = 0.7840), followed by the standard U-Net (Dice = 0.7639) and TransUNet (Dice = 0.7587). The pure ViT model yielded the lowest scores, highlighting its dependency on large-scale datasets. This study confirms the effectiveness of convolutional and hybrid architectures in medical image segmentation and paves the way for more robust computer-aided diagnosis systems.

Keywords: Semantic segmentation, brain tumors, MRI, deep learning, U-Net, DeepLabV3+, TransUNet, Dice coefficient.

الملخص

تُعدّ عملية التجزئة الآلية لأورام الدماغ من صور التصوير بالرنين المغناطيسي (MRI) من القضايا المهمة في مجال التصوير الطبي، حيث تلعب دورًا أساسيًا في التشخيص المبكر، وتخطيط العلاج، والمتابعة السريرية. ونظرًا للتعقيد الكبير والتنوع في أشكال وأحجام الأورام الدماغية، فإن الطرق اليدوية تُعتبر مستهلكة للوقت ومعرضة لاختلافات التقييم بين المختصين. لذلك، يهدف هذا العمل إلى تطوير منهجية تعتمد على تقنيات التعلم العميق من أجل تجزئة أورام الدماغ بشكل آلي ودقيق. ولتحقيق ذلك، تم تنفيذ ومقارنة عدة معماريات حديثة، وهي: U-Net، Attention وU-Net، وDeepLabV3+، وTransUNet، وTransformer Vision (ViT).

أُجريت التجارب باستخدام مجموعة بيانات عامة لصور الرنين المغناطيسي للدماغ، حيث خضعت البيانات لعمليات المعالجة المسبقة والتكبير بهدف تحسين قدرة النماذج على التعميم. كما استُخدمت دالة خسارة مركبة تجمع بين Dice Loss وBinary Cross-Entropy، بالإضافة إلى مُحسِّن Adam، مع تطبيق تقنيات تحسين التدريب مثل الإيقاف المبكر وتقليل معدل التعلم. وقد أظهرت النتائج الكمية، التي تم تقييمها أساسًا باستخدام معامل Dice، أن نموذج DeepLabV3+ حقق أفضل أداء بقيمة Dice بلغت 0.7840، يليه نموذج U-Net بقيمة 0.7639، ثم نموذج TransUNet بقيمة 0.7587. في المقابل، حقق نموذج Transformer Vision أداءً أقل مقارنةً بالنماذج الأخرى، مما يؤكد حاجة النماذج المعتمدة بالكامل على آليات الانتباه إلى كميات أكبر من البيانات لتحقيق نتائج فعالة.

تُبرز هذه الدراسة فعالية النماذج التلافيفية والهجينة في مجال تجزئة الصور الطبية، كما تفتح المجال أمام تطوير أنظمة مساعدة للتشخيص الطبي تتميز بالدقة والموثوقية.

الكلمات المفتاحية: التجزئة الدلالية، أورام الدماغ، التصوير بالرنين المغناطيسي، التعلم العميق، U-Net، DeepLabV3+، TransUNet، معامل Dice.

List of Abbreviations

2D: Two-Dimensional
3D: Three-Dimensional
AI: Artificial Intelligence
API: Application Programming Interface
ASPP: Atrous Spatial Pyramid Pooling
AUC: Area Under the Curve
BN: Batch Normalization
CNNs: Convolutional Neural Networks
COCO: Common Objects in Context
CT: Computed Tomography
DL: Deep Learning
DNN: Deep Neural Network
DSC: Dice Similarity Coefficient
ET: Enhancing Tumor
FCNs: Fully Convolutional Networks
FLAIR: Fluid-Attenuated Inversion Recovery
GANs: Generative Adversarial Networks
GELU: Gaussian Error Linear Unit
GPUs: Graphics Processing Units
HTML: HyperText Markup Language
IOU: Intersection over Union
KNN: K-Nearest Neighbors
LC-UNETR: Local Context U-Net with Transformers
ML: Machine Learning
MRI: Magnetic Resonance Imaging
NDT: Necrotic and Non-Enhancing Tumor
NLP: Natural Language Processing
NumPy: Numerical Python
PCA: Principal Component Analysis
ReLU: Rectified Linear Unit
RF: Random Forest
RMSprop: Root Mean Square Propagation

RNNs: Recurrent Neural Networks
SciPy: Scientific Python
SNR: Signal-to-Noise Ratio
SVM: Support Vector Machine
TC: Tumor Core
TPU: Tensor Processing Unit
ViT: Vision Transformer
WT: Whole Tumor

General Introduction

Brain tumors constitute a major challenge in modern medicine due to their complexity and significant impact on patient health. According to the World Health Organization (WHO), tumors of the central nervous system represent a heterogeneous group of neoplasms for which early diagnosis and accurate characterization are essential to improve prognosis and guide appropriate therapeutic strategies.

Medical imaging plays a crucial role in the detection and analysis of brain tumors. Among available modalities, Magnetic Resonance Imaging (MRI) is considered the reference technique due to its ability to provide high-resolution images of soft tissues without the use of ionizing radiation. It enables clinicians to precisely visualize brain structures and identify abnormalities. However, the manual interpretation of MRI scans remains a complex and time-consuming process, often subject to inter-observer variability.

In this context, automatic brain tumor segmentation has emerged as a key task in medical image analysis. It consists of accurately delineating tumor regions and their subcomponents, which is essential for diagnosis, treatment planning, and monitoring disease progression. Traditional image segmentation methods, such as thresholding, region-based techniques, and edge detection, have shown limitations when dealing with the variability and complexity of tumor structures.

Recent advances in Artificial Intelligence, particularly Deep Learning, have significantly improved the performance of image segmentation techniques. Convolutional Neural Networks (CNNs) and specialized architectures such as U-Net and its variants have demonstrated remarkable effectiveness in extracting complex features and achieving high segmentation accuracy.

This work is part of this research direction and aims to develop an automatic brain tumor segmentation system based on Deep Learning techniques. The main objective is to design, implement, and evaluate different convolutional neural network architectures for MRI image segmentation, and to compare their performance using standard evaluation metrics.

To achieve this objective, the following contributions are proposed:

- Study the fundamentals of medical imaging and brain tumor characteristics
- Explore classical and modern image segmentation methods
- Implement deep learning architectures for tumor segmentation
- Train and evaluate models on annotated MRI datasets
- Compare performance using metrics such as Dice coefficient and IoU
- Develop a visualization interface for segmentation results

The experiments were conducted using a publicly available MRI dataset, and the models are implemented using Python with deep learning frameworks.

This manuscript is organized into four complementary chapters that cover both the theoretical foundations and the practical implementation of the proposed work.

The first chapter presents the fundamental concepts of medical imaging and brain tumors. It begins with an introduction to digital image processing, including image representation, resolution, and basic characteristics. The chapter then provides an overview of the main medical imaging modalities such as X-ray imaging, computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound, highlighting their principles, advantages, and limitations. Particular emphasis is placed on MRI, given its importance in brain tumor diagnosis. In addition, this chapter introduces brain tumors, discussing their types, classification (benign and malignant), causes, and clinical significance.

The second chapter focuses on image segmentation techniques, which are essential for extracting meaningful information from medical images. It introduces the concept of segmentation and its role in medical image analysis. It then presents traditional segmentation approaches, including thresholding methods, edge-based techniques, region-based methods, clustering algorithms, and graph-based approaches, along with their advantages and limitations. The chapter also explores classical machine learning techniques applied to segmentation and introduces semantic segmentation as a key approach for accurately identifying tumor regions.

The third chapter is dedicated to Deep Learning and advanced segmentation architectures. It provides an overview of artificial intelligence and neural networks, followed by a detailed explanation of Convolutional Neural Networks (CNNs). The chapter presents state-of-the-art architectures such as U-Net, Fully Convolutional Networks (FCN), and hybrid models combining CNNs with Transformers. It also discusses training strategies, loss functions, optimization methods, and evaluation metrics such as Dice coefficient, Intersection over Union (IoU), precision, and recall.

The fourth chapter presents the experimental implementation of the project. It describes the dataset used, the preprocessing steps applied to MRI images, and the implementation of different segmentation models. The chapter details the training process, experimental setup, and performance evaluation. It also provides a comparative analysis of the results obtained, discusses the strengths and limitations of the proposed approaches, and highlights challenges encountered during experimentation, along with possible future improvements.

Through this structured approach, this work demonstrates the effectiveness of Deep Learning techniques in improving the accuracy and efficiency of brain tumor segmentation, contributing to the advancement of computer-aided diagnosis systems in medical imaging.

Chapter 1
Medical Imaging Fundamentals

Chapter 1

Medical Imaging Fundamentals

1.1 Introduction

Medical imaging is an important aspect of modern medicine. Physicians use diagnostic imaging in conjunction with clinical assessment in the diagnosis of diseases or medical conditions or to determine the progress of disease or the results of treatments. Although clinical information remains a critical factor in making medical decisions, extra diagnostic information garnered from diagnostic imaging is needed and has become indispensable. Diagnostic imaging has enabled examining the human body in numerous ways, providing much more detailed information that may not be visible or determinable from physical exams alone. Not only have advancements provided unparalleled images of the human body, but they also provide invaluable information about diseases themselves. Many non-invasive imaging methods now exist to investigate the interior of the human body. Such methods include the use of ultrasound, endoscopy, and medical imaging. In this way, medical imaging supplements other medical tests and physical examinations. Medical imaging is a needed tool in clinical practice and in medical research to diagnose disease. [1].

1.2 Digital Image Essentials for Segmentation

1.2.1 Definition

The physical reconstruction of data stored within a computer system is what defines the concept of 'digital image'. A digital image is represented as a data structure, with each pixel representing varying elements of the picture. The numerical value assigned to each pixel denotes its intensity or color, effectively transforming the image into an assembly of individual samples that approximate a continuous reality.

Different techniques can be used to obtain digital images. Digital cameras, scanners

for document scanning, 3D rendering software to produce artificial imagery, and graphic design or drawing tools are some examples of sources that are commonly used. Visual data is transformed into numerical values that can be stored, manipulated, and analyzed by computational systems using these technologies. [2]

1.2.2 Representation of images as numerical matrices

Bitmap images, or raster images for short, are used to represent the actual pixels of an image. The graphical representation of an image is divided into numerous elementary points in this widely used encoding system. The spatial coordinates of each point within the grid and a color or intensity are used to define them. The computer merges all these pixels to create a new image.[3]

Figure 1.1: bitmap image[4]

1.3 Basics of Digital Images

1.3.1 Sampling and Quantization

Spatial sampling

The value of $f(x, y)$ in image sampling is measured at discrete spatial intervals. This is the function of an image. The image's small square area, known as a pixel, is represented by each sample. A digital image can be represented as a 2D array of pixels. These pixels are identified by (x, y) spatial coordinates where x and y contain integer values that correspond to their position in the image grid. [5]

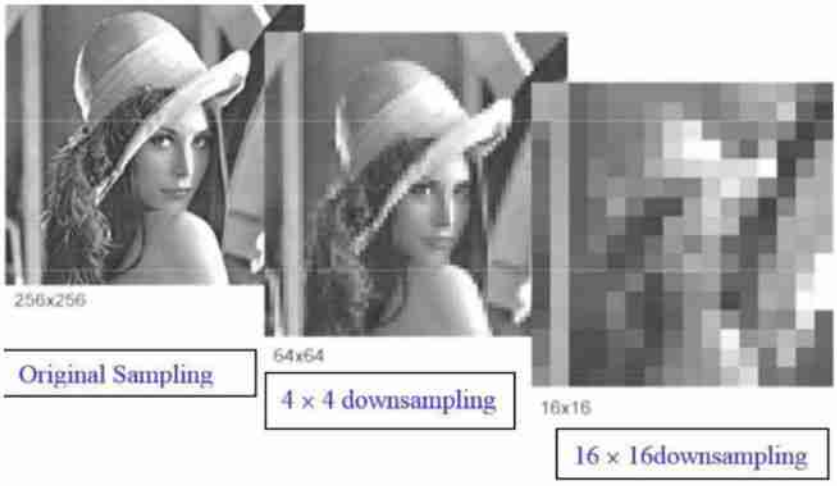


Figure 1.2: Examples: Image Sampling[5]

Quantization

To quantify images, one can use an integer to represent the brightness or intensity of individual pixels. A finite set of discrete levels must be used to convert continuous intensity values into quantized form, as digital computers are capable of processing numerical data. Through this conversion, the measurement range is broken down into discrete units that can be expressed and processed as integer numbers.[5]

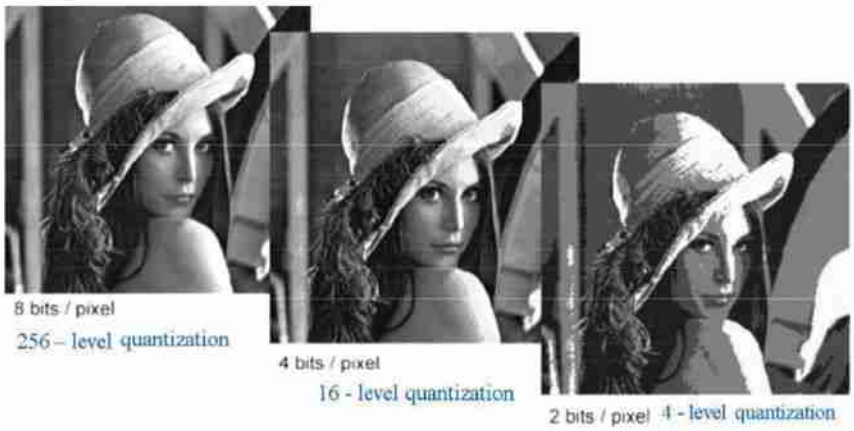


Figure 1.3: Quantization of an image[5]

Pixel and Voxel Size

An image's smallest two-dimensional component that was sampled is known as a pixel. The dimensions of the image are defined along two spatial axes, usually expressed in millimeters, and determine the in-plane spatial resolution. This definition is known as "dimension space." Clinical MRI utilizes a range of pixel sizes, from millimeter (e.g., $1 \times 1\,\mathrm{mm}^2$) to submillimeter.

Alternatively, the word "voxel" refers to any element of volume in three dimensions that can be visualized. A. It is defined by the size of the pixels in the plane of an image, as well as their thickness along the third axis. Slice thickness in 2D multislice imaging is typically around 5 mm, which can increase to submillimeter depth when using more sophisticated 3D acquisition techniques.[6]

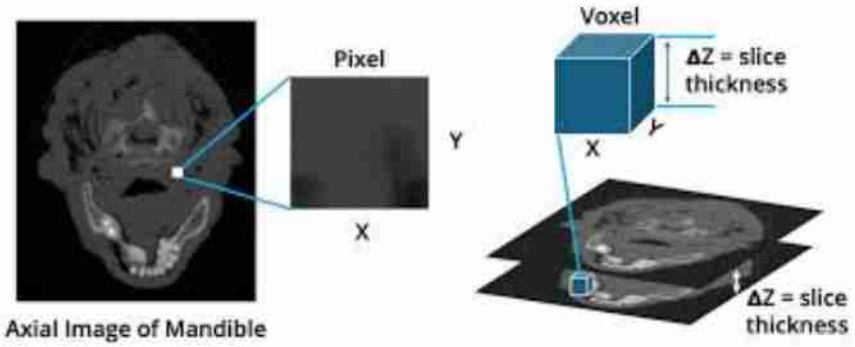


Figure 1.4: Pixel and Voxel size [7]

1.3.2 Intensity and Gray Levels

The term "grayscale" is used to describe an image that has a range of shades of gray, from black to white. Each pixel in a grayscale image is exclusively data on intensity and not color in digital imaging. Typically, the intensity value is represented numerically in an 8 bit representation, with values between 0 (black) and 255 (white), and intermediate values are expressed as having different levels of gray.[8]

Importance of Gray-Level Contrast in Tumor Boundary Delineation

The primary intensity difference between tumor tissues and surrounding areas is mainly represented by gray level contrast, which is crucial for detecting tumor boundaries in medical images. The importance of this contrast in edge detection algorithms is lost when the contrast is not enough to identify meaningful boundaries, resulting in reduced performance. Consequently, subtle variations in pixel intensity may become indistinguishable from noise or gradual transitions between tissues.

This is done by using fuzzy logic combined with a hesitation constant, which effectively controls the uncertainty in low contrast areas. However, the effectiveness of this technique still depends on measurable intensity differences between the true tumor boundary and background tissues. Thus, correct segmentation of tumors is still necessary for accurate diagnosis and effective treatment planning.[9]

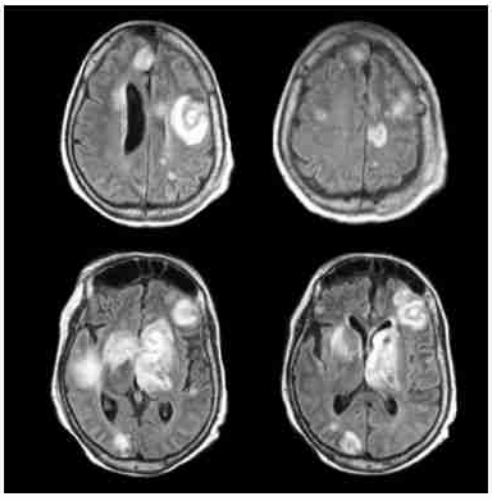


Figure 1.5: Tumor Boundary [10]

1.3.3 Noise and Blur

Different types of noise in medical images

- **Gaussian Noise**: The origin of this noise can be attributed to thermal fluctuations and electronic circuit interference during data acquisition. Common usage is found in almost all medical imaging methods, such as CT (as opposed to polysomnographic testing), resulting in potentially flawed image quality and a high diagnostic threshold.[11]

Figure 1.6: Gaussian Noise [12]

- **Speckle Noise**: Speckle noise is a type of multiplicative granular noise that is natural in coherent imaging systems like ultrasound and MRI. Due to interference of the coherent waves, it generally gives a granular appearance to the images, so it might obscure small anatomical structures and reduce the overall image quality, making the diagnosis challenging.[11]

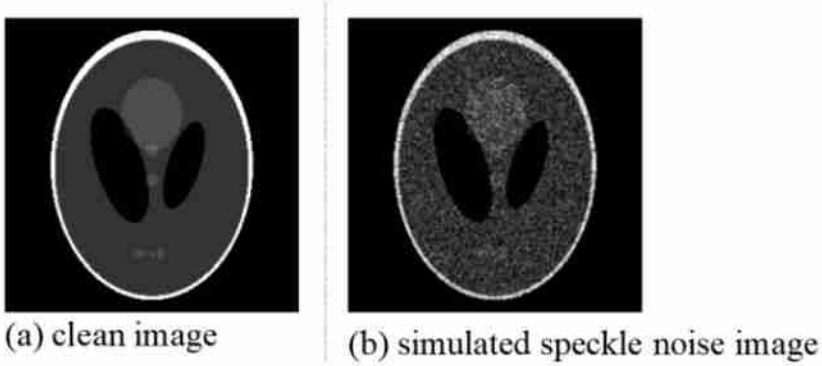


Figure 1.7: Speckle Noise [13]

- **Poisson (Shot) Noise**: This noise arises from the random arrival of photons detected over time. It is commonly observed in X-ray imaging, CT, and low-light imaging scenarios where photon-starved data are present.[11]

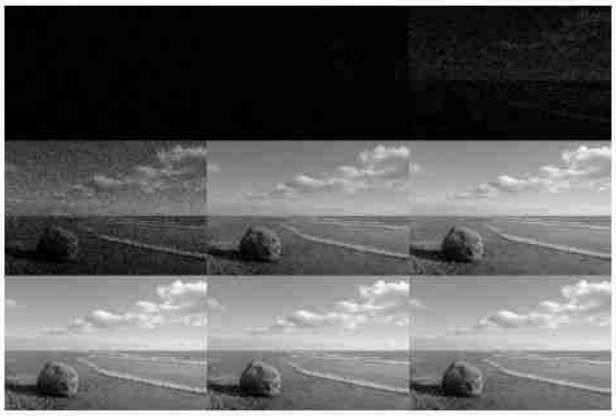


Figure 1.8: Shot Noise [14]

- **Salt-and-Pepper Noise**: This noise appears as randomly distributed white and black pixels, as suggested by its name. It is typically caused by transmission errors, faulty sensor elements, or bit mismatches during digital data storage and transmission.[11]

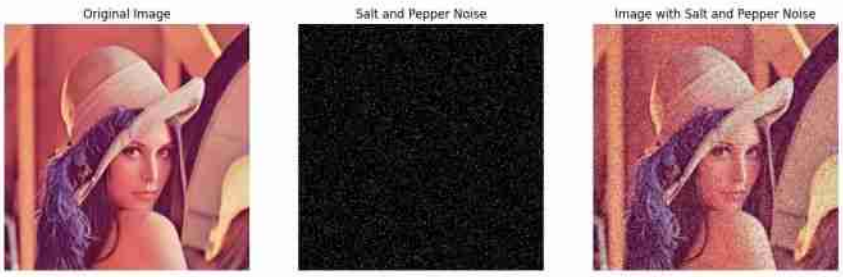


Figure 1.9: Salt-and-Pepper Noise [12]

- **Rician Noise**: Rician noise is commonly observed in MRI magnitude images. It arises when the magnitude of complex MR data is computed, transforming the

underlying Gaussian noise in the real and imaginary components into a Rician distribution. This type of noise particularly affects images acquired with low SNR and can significantly influence the accuracy of subsequent image analysis and processing tasks[15]

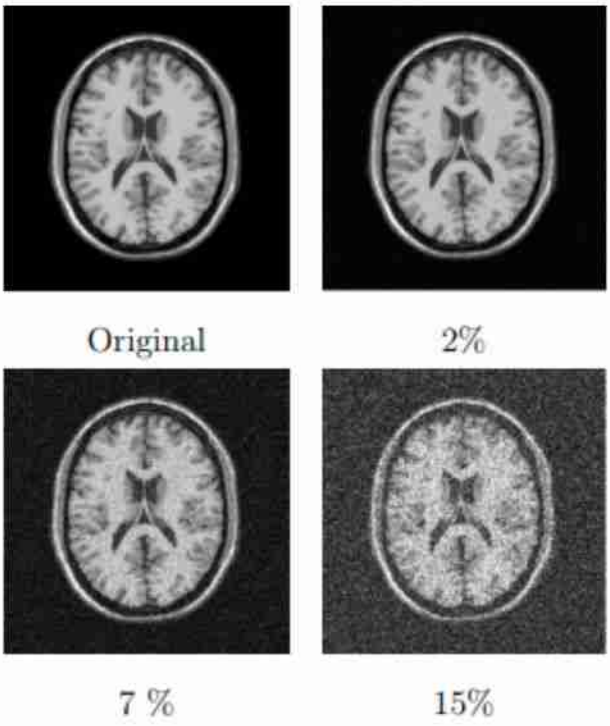


Figure 1.10: Simulation of Rician noise for different levels [15]

- **Periodic Noise**:This type of noise appears as regular patterns or ripples across the image. It is usually caused by electrical interference during image acquisition and often manifests as repetitive structures that produce distinct peaks in the frequency domain.[16]

Figure 1.11: Periodic Noise [17]

- **Motion Artifacts**: Although often categorized as a separate type of imaging artifact, patient motion during acquisition such as breathing, cardiac activity, or invol-

untary movementscan introduce blurring and ghosting effects in MRI images. These effects behave similarly to noise and can significantly degrade image quality, thereby affecting the reliability of subsequent analysis and diagnostic interpretation.[18]

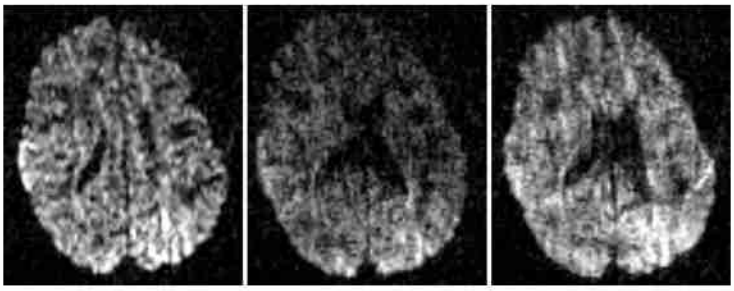


Figure 1.12: Motion Artifacts[19]

Impact of Noise on Segmentation Accuracy

Noisy data can lower segmentation accuracy since it introduces inconsistencies and inaccuracies into the labels, leading the network astray by learning these errors rather than the real structures. As it is trained on labels that have suffered from erosion, dilation or elastic transformation it learns the noise and does not learn the actual structure boundary, resulting in poor generalization on clean images. This noise corrupts the training signal leading the network towards a local optimum. This leads to errors in the accuracy of the segmentations and lower values of metrics and so lower the clinical reliability of the network.[20]

Image blur and resolution loss

Blurring images can help smooth out jarring pixel changes while keeping the important edges intact. The effect of blurring can be that the fine details of the images are smoothed out to put a focus on the overall structure of the images. It's useful in image processing for reducing the noise and small speckles in images by averaging the pixel value of neighboring pixels. It can be used to help algorithms focus on the biggest objects in a scene, rather than the fine details. This can be useful for object recognition and edge detection. The blurring effect can also be used to help draw attention to a specific region in an image by softening the surrounding area. This is also known as a "mask".[21]

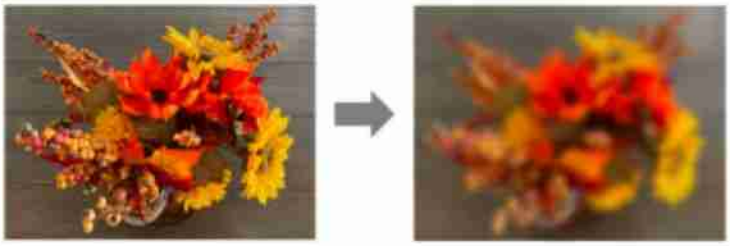


Figure 1.13: An original image (left) and a blurred version (right)[21]

1.4 Overview of Medical Imaging Modalities

1.4.1 X-ray Radiography

The use of radiography is a crucial imaging technique that finds broad application in healthcare and industry. The use of ionizing radiation, such as X-rays or gamma rays, allows for the creation of detailed images of internal structures within the human body or manufactured objects. The primary use of this non-invasive technique is for medical diagnosis, including the detection of fractures or pathological abnormalities, but it also serves as an essential tool for industrial NDT, material inspection, and quality control, thereby improving both clinical decision-making and safety in industrial settings. Additionally, it has applications across multiple fields.[22]

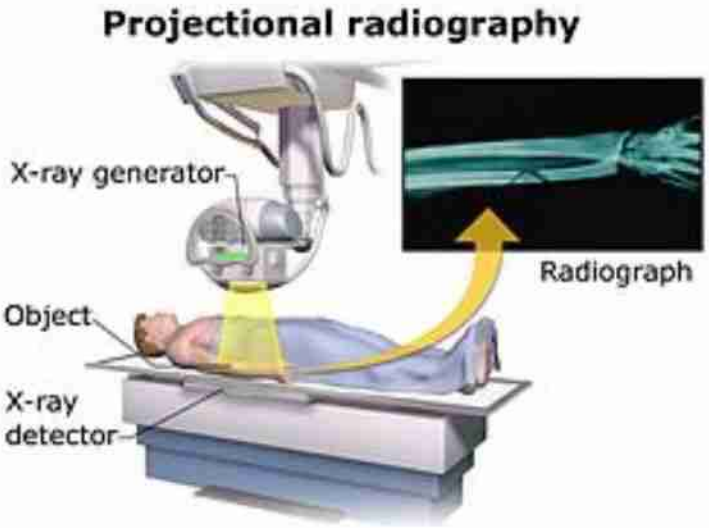


Figure 1.14: Acquisition of projectional radiography, with an X-ray generator[23]

1.4.2 Computed Tomography (CT)

A sophisticated X-ray imaging technique, a Computed Tomography scan, uses alternating X-ray beams that rotate around internal body structures to produce cross-sectional images. CT's technology enables the transmission of detailed cross-sectional information, eliminating the need for image superposition in traditional X-rays. The ability to visualize anatomical structures with precision enables better clinicopathological correlation and aids in the accurate diagnosis and evaluation of suspected diseases.[24]

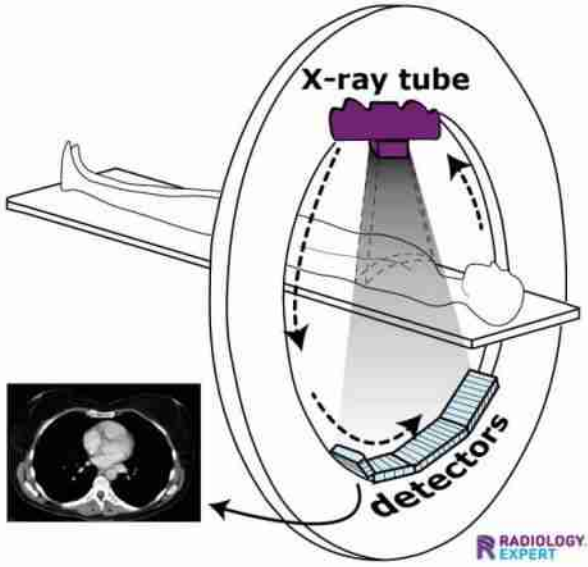


Figure 1.15: Computed Tomography

1.4.3 Magnetic Resonance Imaging (MRI)

The MRI is an imaging technique that allows for the visualization of internal structures and functional aspects of the human body without any intrusion. The application of non-ionizing electromagnetic radiation eliminates any known risks associated with exposure. By applying carefully controlled magnetic fields and RF pulses, the hydrogen nuclei of the body align with the external magnetic field and generate energy upon excitation by radiating a radiofrequency wave. By detecting and processing the emitted signal, a computer produces high-resolution cross-sectional images in any plane to provide comprehensive anatomical and functional information for clinical evaluation and diagnosis. [25]

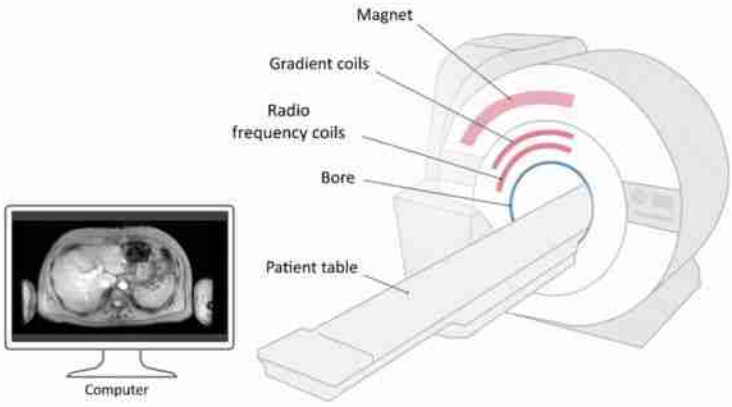


Figure 1.16: MRI scanner [26]

1.4.4 Ultrasound Imaging

An ultrasound scan is a non-invasive medical imaging method that uses high-frequency sound waves to produce real-time images of internal body structures. This device uses ultrasound to send sound waves into the body, which are then directed towards organs (lungs), muscles, blood vessels, and other tissues such as the heart, abdomen, or pelvis. The machine captures the echoes that bounce back from these structures and uses them to generate detailed images that can be displayed on a monitor, allowing clinicians to evaluate anatomical features, monitor physiological function, and aid in accurate diagnosis without exposure to ionizing radiation.[27]

Figure 1.17: Ultrasound Imaging machine [28]

1.4.5 Nuclear Medicine (PET / SPECT)

Medicinal imaging that involves administering radioactive substances through injection, oral ingestion or breathing inhalation is known as nuclear medicine. Following the administration of materials, a special camera is utilized to capture close-up images of organs and tissues within the body.

By using a specific camera, the radiotracer measures the radiation produced by the tracer. The heart, lungs, liver and gallbladder, as well as bones and other body parts, can be visualized by combining the data with other information. Nuclear medicine has the potential to be employed for both diagnostic and therapeutic purposes.

Some of the most commonly used nuclear medicine imaging techniques include:

- **PET scans (Positron Emission Tomography)**
- **SPECT scans (Single Photon Emission Computed Tomography)**[29]

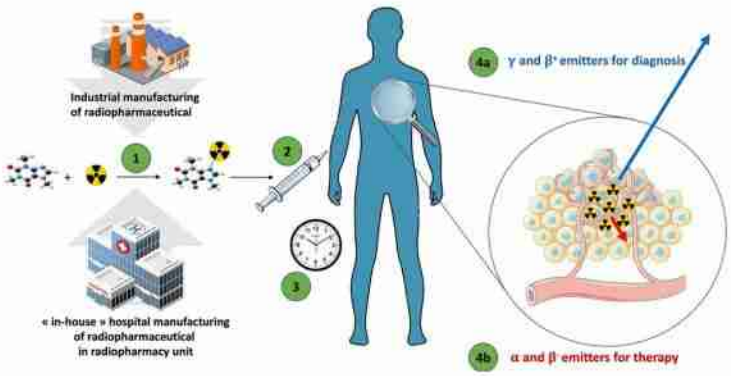


Figure 1.18: Nuclear Medicine[30]

1.5 MRI as the Preferred Modality for Brain Tumor Imaging

Magnetic resonance imaging (MRI) is the preferred modality for brain tumor diagnosis and analysis because it provides superior soft tissue contrast non-invasively, allowing clear visualization of tumor boundaries and sub-compartments. Unlike other imaging techniques, MRI can generate multiple sequences such as T1-weighted, T1-weighted with contrast, T2-weighted, and T2-FLAIR—each highlighting different tissue characteristics, which together enable precise delineation of active tumor, necrosis, and surrounding edema. This comprehensive imaging capability, combined with its wide clinical availability and compatibility with advanced analysis methods for segmentation and monitoring, makes MRI the standard for diagnosis, treatment planning, and longitudinal assessment of brain tumors.[31]

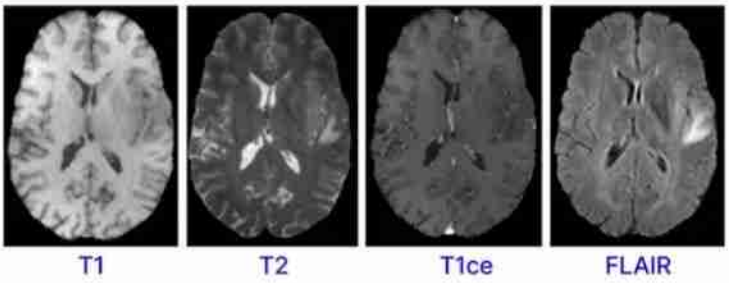


Figure 1.19: MRI sequences for brain[32]

1.6 Conclusions

Accurate diagnosis is an important factor for the effective treatment, which has been enhanced by the recent advances in medical imaging. Among all medical imaging modalities, MRI plays a critical role in the diagnosis of brain tumors due to its high resolution

and superior soft tissue contrast, the first topic of this chapter. The second part is dedicated to the necessary background on brain anatomy and the basic principles of medical imaging in order to set a foundation for the advanced diagnostic and analysis techniques discussed in the next chapter.

Chapter 2
Brain Tumor

Chapter 2

Brain Tumor

2.1 Introduction

Brain tumors constitute a complex group of abnormalities involving the growth of brain or brain-adjacent cells. The function of the brain can be severely or totally impaired depending on their location and the time that passes before their diagnosis and treatment. In hundreds of thousands of cases, the outcome is often severe or lethal. The most common forms of these tumors are glioblastoma, a fast-growing tumor that originates locally, meningioma a slow-growing tumor and tumors that are secondary. Some brain tumors appear to be the result of a genetic, environmental or metastatic disease.

2.2 Definition of Brain Tumor

2.2.1 General definition

A brain tumor is a mass of abnormal cells that is found inside the brain or skull. The tumor can be benign or malignant. Primary brain Tumors come from the brain tissue. Cancer cells can also come from other parts of the body. The treatment is based on the type, size and where the tumor is found. It can focus on taking out the whole tumor or managing symptoms based on the condition. Tumors are a bit different so to diagnose and make decisions you need to be able to classify them.[33]

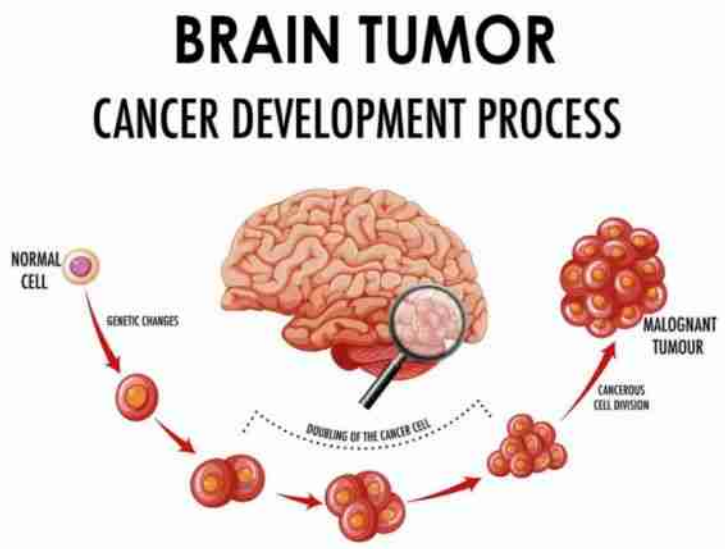


Figure 2.1: Brain tumor cancer development process[34]

2.2.2 Distinction between primary and secondary (metastatic) tumors

Definition and Origin

Primary Tumor: This is the original cancer, named after the organ or tissue type where it first began (e.g., breast cancer, lung cancer, colon cancer).[35]

Secondary (Metastatic) Tumor: This tumor forms when cancer cells travel through the bloodstream or lymphatic system to a new location, establishing a new growth. For example, breast cancer that has spread to the bones is considered metastatic breast cancer, not bone cancer.[35]

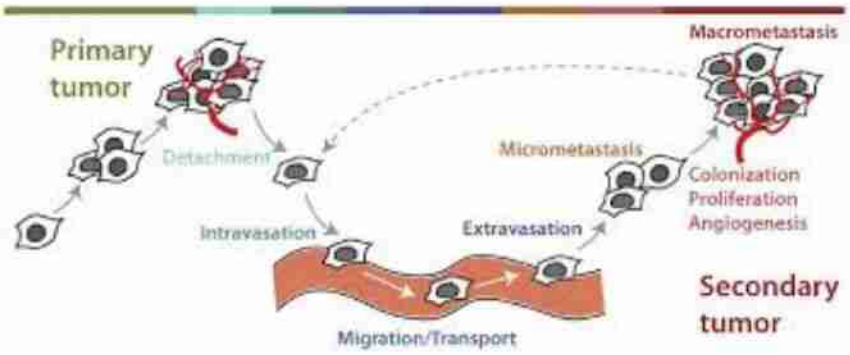


Figure 2.2: Primary tumor and Secondary tumors[36]

Characteristics and Behavior

- **Cell Type:**Secondary tumors are made of the same type of cells as the primary tumor.[37]
- **Aggressiveness:**Metastatic tumors are generally more aggressive, having developed the ability to break away,travel and colonize new hostile environments.[37]
- **Genetic makeup:**While similar to the primary tumor, secondary tumors may possess additional genetic mutations that make them more resistant to treatment.[37]
- **Growth Patterns:**Primary tumors often exhibit more predictable growth patterns, while metastases can sometimes show different and non-infiltrative growth patterns.[37]

2.3 Types of Brain Tumors

2.3.1 Benign vs malignant tumors

Benign Tumors: are non-cancerous that remain localized, do not invade surrounding tissues or spread distantly and typically have slow growth and clear borders. while often not problematic, they can cause complications such as pain or difficulty breathing by compressing nearby structures, which may require surgical removal. Once excised they are unlikely to recur. However, certain types like polyps, can become malignant and are therefore monitored or removed preemptively.[38]

Malignant Tumors: are cancerous growths characterized by the uncontrolled division of cells that invade surrounding tissues.They can spread to distant sites in the body through the bloodstream or lymphatic system, a process called metastasis. Common sites for metastasis include the liver, lungs, brain and bones. Treatment depends on the stage:early-stage cancers are often treated with surgical removal, potentially combined with chemotherapy or radiotherapy. If the cancer has already metastasized, systemic treatments like chemotherapy or immunotherapy are typically required.[38]

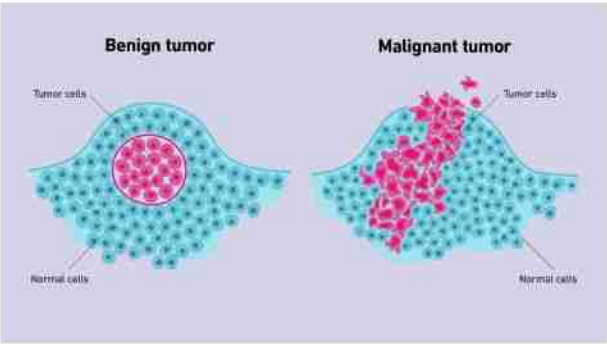


Figure 2.3: Benign vs malignant tumors[39]

2.3.2 Common types of brain tumors

Brain tumors are categorized into different types, such as:

- **Gliomas:** are tumors that develop from glial cells, which support and protect nerve cells in the brain. They include astrocytomas, glioblastomas, oligodendrogliomas, and ependymomas. Most gliomas are malignant, with glioblastoma being the most aggressive and common type.[40]

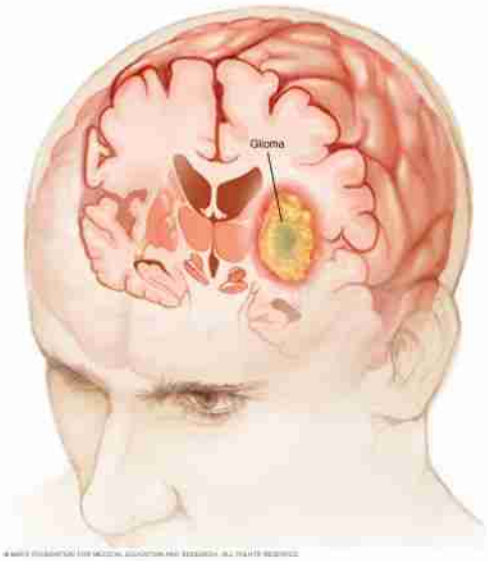


Figure 2.4: Gliomas tumors[41]

- **Choroid plexus tumors:** arise from cells that produce cerebrospinal fluid in the brain's ventricles. They can be either benign or malignant. The malignant form, called choroid plexus carcinoma, occurs more frequently in children.[40]

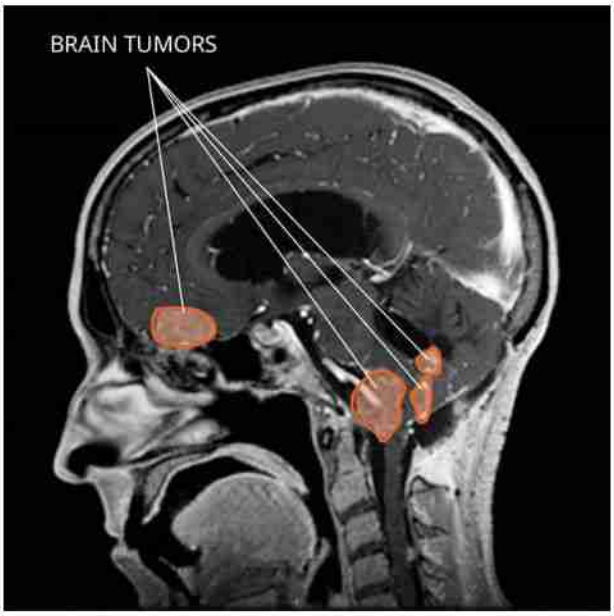


Figure 2.5: Choroid plexus tumors[42]

- **Meningiomas:** arise from the meninges, the protective membranes surrounding the brain and spinal cord. They are usually benign but can sometimes be malignant. Meningiomas are the most common type of benign brain tumor.[40]

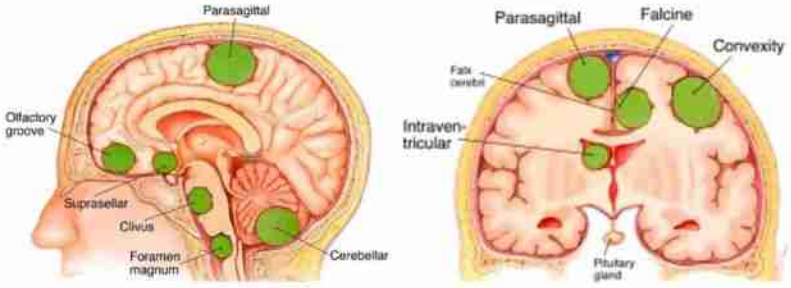


Figure 2.6: Meningiomas tumors[43]

- **Nerve tumors:** develop in or around nerves connected to the brain. The most common type is acoustic neuroma (schwannoma), which affects the nerve related to hearing and balance. These tumors are typically benign and slow-growing.[40]

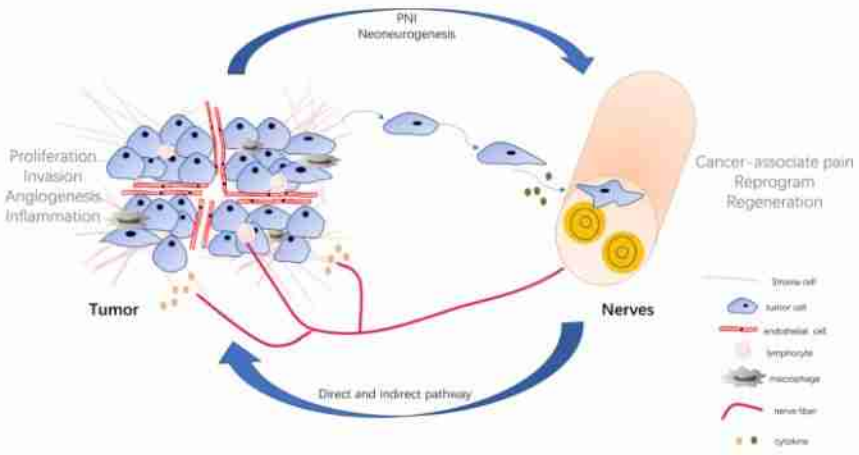


Figure 2.7: Nerve tumors[44]

- **Pituitary tumors:** occur in or near the pituitary gland, which controls hormone production. Most pituitary tumors are benign and may affect hormone balance. Craniopharyngioma is a related tumor that forms near the pituitary gland.[40]

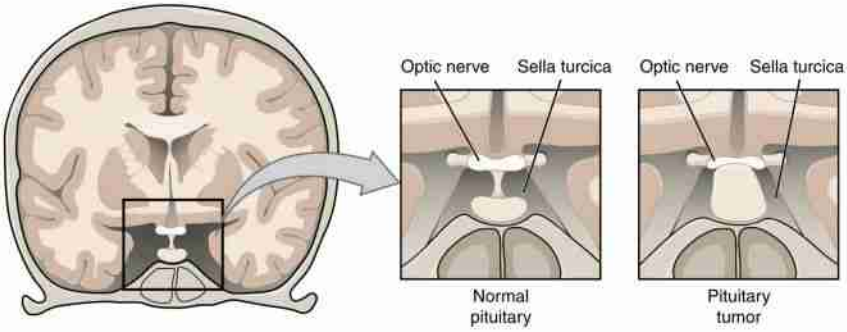


Figure 2.8: Pituitary tumors[45]

- **Other brain tumors:** Other rare brain tumors can develop in muscles, blood vessels, connective tissues, or skull bones. Some malignant tumors originate from immune system cells in the brain. This condition is known as primary central nervous system lymphoma.[40]

2.4 Clinical Importance of Segmentation

The medical imaging segmentation method is important for the clinical workflow to determine the appropriate medical operation for the appropriate individual based on the appropriate test by separating the anatomical structure from the imaging examination, such as MRI or CT, to get an accurate diagnosis and appropriate treatment.[46]

Role in Diagnosis

Segmentation accurately identifies tumors, organs and tissues, enabling the early detection of abnormalities and allowing quantitative tissue characterization, significantly improving the accuracy of diagnosis compared to manual methods.[47]

Role in Treatment Planning

Segmentation provides detailed boundaries to support radiotherapy dose calculation and implant design, as well as support personalized treatment strategies, such as tailoring chemotherapy to the size and location of the tumor.[47]

Role in Monitoring

Segmentation tracks changes in the size of lesions or organs over time to assess how well the treatment is working and the progression of the disease, enabling timely adjustments to the therapy and improving the treatment outcome through serial comparisons.[47]

Tumor Applications

Segmentation accurately measures tumor size in pixels or volume, important for staging and response evaluation. The correct localization of the tumor is given via region of interest extraction. The exclusion of irrelevant regions for better visualization is important. While surgical planning, the precise outline in the contour guide the resection, reduce risk and improve accuracy.[47]

2.5 Challenges in Brain Tumor Segmentation

Brain tumor segmentation in magnetic resonance imaging (MRI) is a crucial yet highly challenging task in medical image analysis. Accurately distinguishing tumor tissue from healthy brain parenchyma, surrounding edema, and necrotic regions is complicated by a combination of biological variability and technical limitations. Several key challenges contribute to the complexity of this task, including:

Variability in Tumor Shape and Size

Brain tumors are highly heterogeneous in terms of appearance, location, and shape. They can be multifocal (existing in multiple places) or diffuse, making it difficult for automated models to detect and delineate them uniformly. Size ranges from small, difficult-to-detect anomalies to large masses that deform surrounding brain structures.[48]

Image Noise and Artifacts

MRI scans are often subject to noise (e.g., Rician noise) and artifacts such as motion blur, intensity non-uniformity (bias field), or sensitivity loss, which degrade image quality and obscure edge details. These artifacts can interfere with segmentation algorithms, causing them to misclassify healthy tissue as tumor, or vice versa.[48]

Ambiguous or Unclear Tumor Boundaries

Tumors, especially malignant ones, often lack clear boundaries and blend into surrounding healthy tissue due to infiltration. The ambiguous, fuzzy, or low-contrast edges make it difficult for algorithms to determine exactly where the tumor ends, leading to imprecise volume estimation.[48]

Differences Between Imaging Modalities (Multi-modal MRI)

Brain tumors are analyzed using multiple, complementary MRI sequences—T1, T1-contrast enhanced (T1-CE), T2, and Fluid Attenuated Inversion Recovery (FLAIR). Each modality highlights different characteristics (e.g., necrosis, edema). A significant challenge is ensuring the consistency of image quality, alignment, and intensity normalization across these modalities to prevent poor segmentation results.[48]

2.6 Conclusions

The purpose of this chapter is to give a general look at brain tumors, beginning with a basic description and the differences between primary and secondary tumors. Different kinds of brain tumors, both benign and malignant, are shown to explain how they differ in source, how they act, and how they influence a patient. This chapter also discusses why tumor segmentation is medically important, mostly for diagnosis, treatment planning, and monitoring disease progression. Lastly, This chapter discusses the main challenges with spotting brain tumors in MRI scans, like how different tumors can be, how noisy the images are, and how unclear the borders can look. This chapter gives the medical and clinical background needed to grasp the segmentation ways and computational approaches that will be given in the next chapter.

Chapter 3
Automatic Medical Image Segmentation

Chapter 3

Automatic Medical Image Segmentation

3.1 Introduction

Manual segmentation is often a labor-intensive, inconsistent process, so automated techniques are important for to improve clinical workflows' accuracy, reproducibility, and efficiency in the evaluation of brain magnetic resonance imaging (MRI). In this chapter, we provide a description of some basic aspects of automated medical image segmentation, as well as ways to implement them.

In this chapter, traditional segmentation techniques, machine learning methods, and deep learning architectures (CNNs and Transformers) are reviewed. Additionally, the chapter describes the primary metrics that are used to evaluate segmentation performance, and it explores recent state-of-the-art techniques for analyzing brain images.

3.2 Fundamentals of Image Segmentation

3.2.1 Definition of Image Segmentation

Image segmentation is the process of dividing a digital image into a set of connected and non-overlapping regions that together cover the entire image. Each region consists of pixels that are homogeneous with respect to one or more characteristics, such as gray-level intensity, color, texture, or spectral properties, while neighboring regions are significantly different. The objective of image segmentation is to transform the image into a representation that is more meaningful and easier to analyze, enabling the identification and extraction of relevant structures or features for further processing. [49]

Figure 3.1: Image Segmentation [50]

3.2.2 Medical Image Segmentation

Medical image segmentation is a fundamental task in medical imaging, enabling accurate diagnosis, treatment planning, and quantitative analysis by delineating anatomical structures and pathological regions. Traditional segmentation techniques, including thresholding, edge-based, region-based, clustering, and graph-based methods, have been widely used due to their simplicity and interpretability, but they often struggle with noise, intensity variations, and ambiguous boundaries. Recent advances in deep learning have significantly improved segmentation performance through architectures such as CNNs, FCNs, U-Net, RNNs, GANs, and Autoencoders, which enable automatic feature learning from complex medical data. However, challenges related to data availability and computational cost remain. To overcome these limitations, hybrid approaches that combine traditional methods with deep learning have emerged, offering enhanced robustness and accuracy. This review provides a comprehensive overview of these segmentation strategies and highlights their strengths, limitations, and applications in medical imaging.[51]

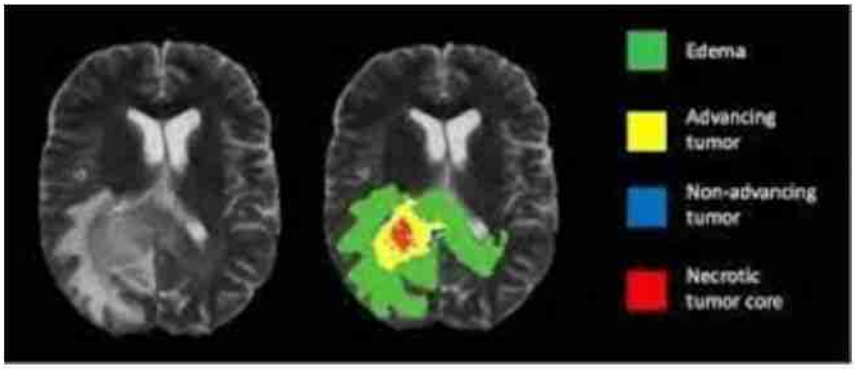


Figure 3.2: Medical Image Segmentation [52]

3.3 Traditional Segmentation Methods

Traditional methods rely on predefined rules and low-level image features such as intensity, edges, and spatial relationships. Common techniques include:

- **Thresholding based methods:** Segment regions based on pixel intensity values.[53]

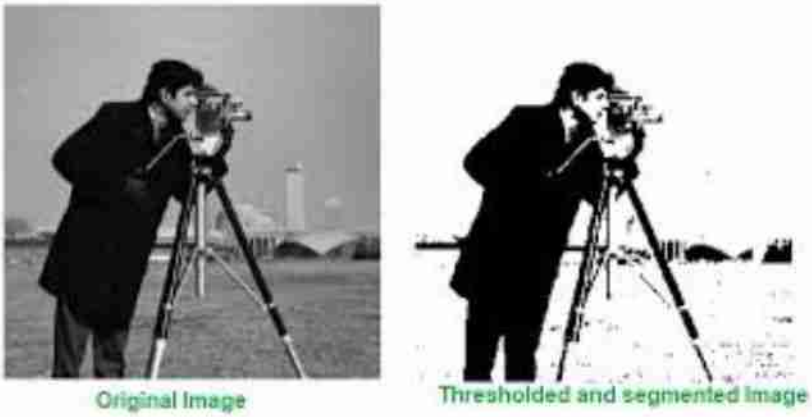


Figure 3.3: Thresholding based methods [54]

- **Edge based methods:** Detect object boundaries using gradient information.[53]

Figure 3.4: Edge based methods [55]

- **Region based approaches:** Group pixels into homogeneous regions based on similarity criteria.[53]

Figure 3.5: Region based approaches [56]

- **Clustering based methods:** Classify pixels into clusters using unsupervised learning techniques.[53]

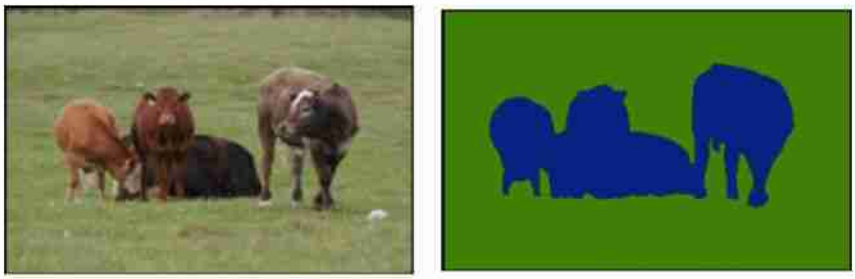


Figure 3.6: Clustering based methods [57]

- **Graph based segmentation:** Represent images as graphs and partition them using optimization strategies.[58]

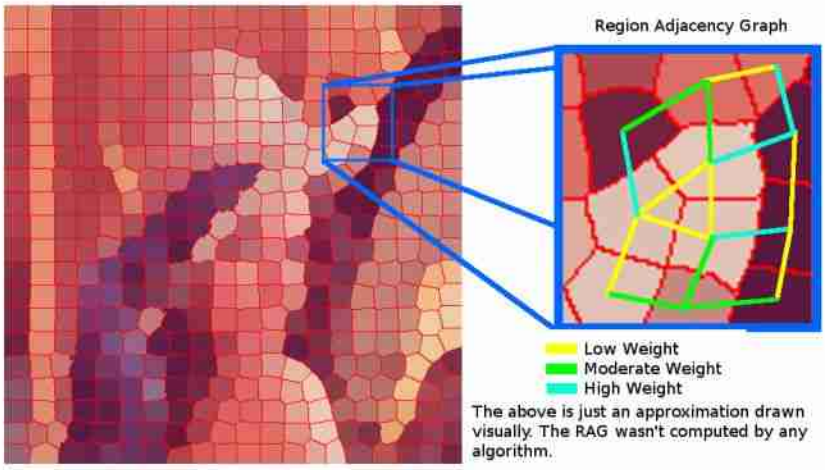


Figure 3.7: Graph based segmentation [59]

Although these methods are computationally efficient and interpretable, their performance degrades in the presence of noise, intensity inhomogeneity, and unclear boundaries.[53]

3.4 Machine Learning-Based Segmentation

3.4.1 Artificial Intelligence Overview

Artificial Intelligence (AI), a part of computer science, aims to create systems that can handle jobs that usually need human smarts, like thinking, learning, seeing, and deciding.

Instead of just copying how people act, AI tries to make smart agents that can act well and logically in various unclear situations.

The idea of smarts in AI can be viewed ways, like acting like a human (as said in the Turing Test), thinking like a human using models, thinking logically, and acting logically to get the best results.

These views give the base for judging machine smarts.

AI systems usually have traits like seeing, thinking, learning, changing, planning, talking, and freedom.

The field has a few key sub-areas, like ML, DL, NLP, Computer Vision, and Reinforcement Learning. Each helps grow smart technologies.

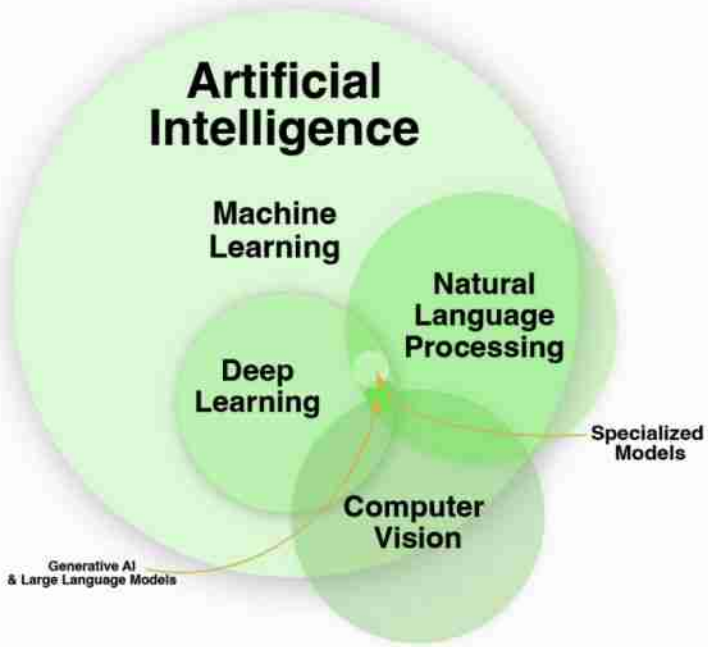


Figure 3.8: Subfields of Artificial Intelligence [60]

In short, Artificial Intelligence is a changing area that puts together computer models and data methods to copy and improve smart behavior in many uses.[61]

3.4.2 Machine Learning Techniques for Segmentation

Machine learning has helped medical image segmentation move forward through data-driven methods that can show complicated patterns in images. Different from standard segmentation ways that are based on rules, machine learning learns from data. This betters how it can change and work in different clinical situations.

Machine learning has three main types: supervised learning, unsupervised learning, and reinforcement learning.[62]

Supervised Learning

Supervised learning is often used for medical image segmentation. This method trains models using labeled data, where each image has a matching segmentation mask. The aim is to learn how to assign labels (like tumor, edema, or healthy tissue) to each pixel or voxel.

Typical algorithms for this include SVM, k-NN, Decision Trees, Random Forests, Logistic Regression, and Artificial Neural Networks. Before classification, these methods usually extract features by hand, such as intensity, texture, and spatial data. These methods can do well, but how well they do depends on having correct and labeled data

available.[62]

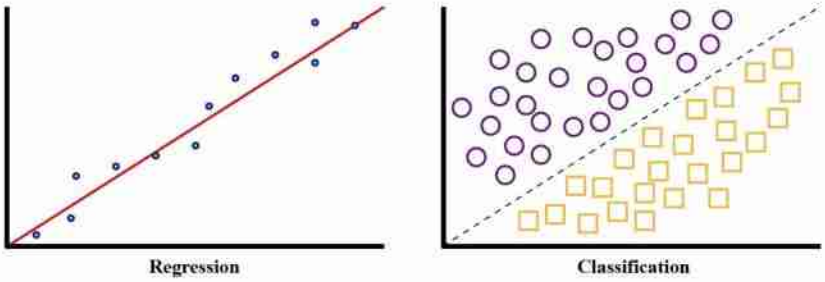


Figure 3.9: Regression vs Classification in Machine Learning [63]

Unsupervised Learning

Unsupervised learning works by finding patterns in data that isn't labeled. In image segmentation, it groups pixels or areas together based on how alike they are.

Common methods include k-means clustering, hierarchical clustering, and ways to reduce the number of variables, like PCA. These are helpful when you don't have much labeled data. But, they might not work as well on complicated medical images because they don't have any prior knowledge to guide them.[62]

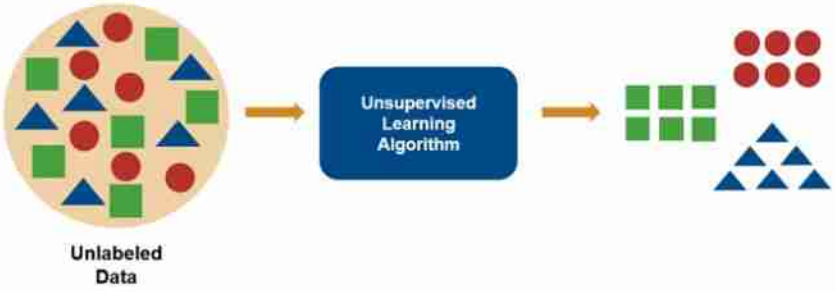


Figure 3.10: Unsupervised Learning [64]

Reinforcement Learning

Reinforcement learning uses a framework where an agent interacts with an environment. The model learns to make decisions by getting feedback in the form of rewards. While not used as often as supervised methods for medical image segmentation, reinforcement learning offers a good structure for adaptive segmentation strategies that happen in sequence.[62]

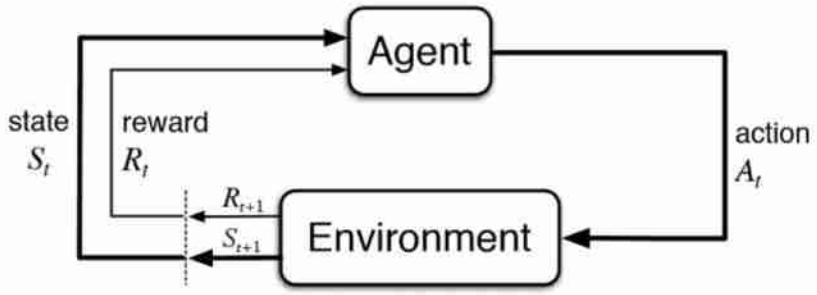


Figure 3.11: Reinforcement Learning [65]

Role of Machine Learning in Segmentation

Machine learning fills a space between classic image work and deep learning. It helps automatically sort image parts while staying understandable and quick. But it depends on features made by hand and can't pull out features in levels well, which can hold it back when dealing with tricky medical images.

Because of these limits, deep learning models for segmentation have come about, learning how to represent features from simple image data on their own.[62]

3.5 Deep Learning-Based Segmentation

3.5.1 Deep Learning for Segmentation

Deep learning is the primary tool used today to accurately delineate tumor boundaries within medical images.Here, we are not talking about customer segmentation or optimizing marketing campaigns, but rather medical semantic segmentation, where each pixel in an MRI image is classified as either part of the tumor or healthy tissue. Modern architectures such as U-Net, V-Net, DeepLab, and attention-based models have proven highly effective for this specific task because they are designed to capture precise spatial context and fine grained details even when labeled data is scarce, a common challenge in medical imaging. Moreover, some of these models integrate attention mechanisms or GANs to refine segmentation boundaries and reduce noise. On the other hand, while machine learning algorithms in commercial settings aim to understand customer behavior and deliver personalized experiences, the same underlying philosophy extracting hidden patterns from data is applied in medicine, but with a life saving purpose: precisely determining tumor size, monitoring its progression, and planning surgery or radiotherapy with exceptional accuracy. [66]

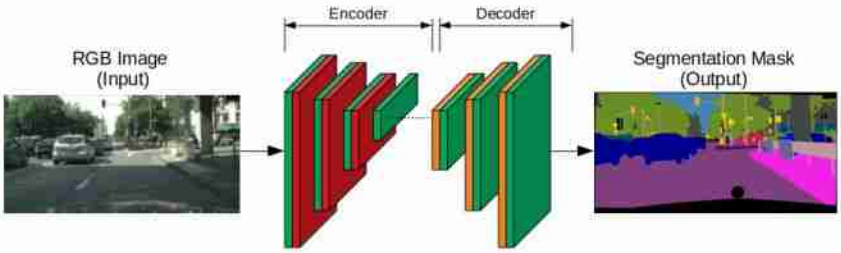


Figure 3.12: Deep Learning for Segmentation [67]

3.5.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the foundation of modern computer vision. They started simply with LeNet-5 for recognizing handwritten digits, then evolved significantly after AlexNet in 2012, which used GPUs and proved that deep networks could outperform humans in image classification.

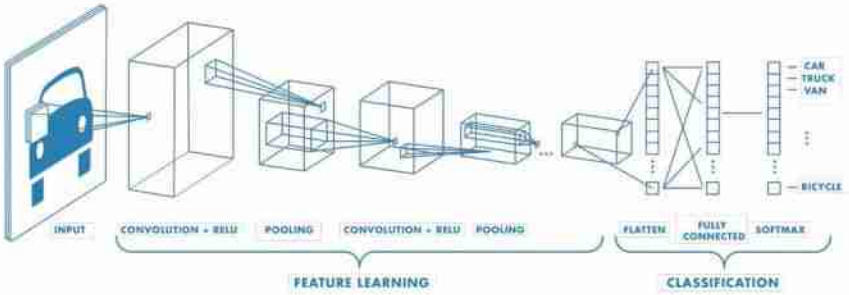


Figure 3.13: Convolutional Neural Networks Architecture [52]

After that, smarter architectures emerged:

- VGG showed that depth improves accuracy.
- GoogLeNet introduced "Inception" modules to capture features at multiple scales within the same layer.
- ResNet solved the vanishing gradient problem using "skip connections" enabling networks with hundreds of layers without performance loss.
- MobileNet and ShuffleNet were designed to be lightweight and fast, ideal for mobile devices with limited resources.

All these advances rely on key components:

- Activation functions (like ReLU),
- Loss functions (such as Cross-Entropy or Dice Loss),
- Optimizers (like Adam or RMSprop).

Experiments have shown that choosing these components directly impacts a model's accuracy and training speed.[68]

3.5.3 Fully Convolutional Networks

In the area of brain tumor segmentation using magnetic resonance imaging (MRI), Fully Convolutional Networks (FCNs) are a base point in the progress of segmentation models that use deep learning. The FCN design brought in a changing idea: adapting standard convolutional neural networks, which were first made for image sorting, to do detailed, pixel-by-pixel prediction work.

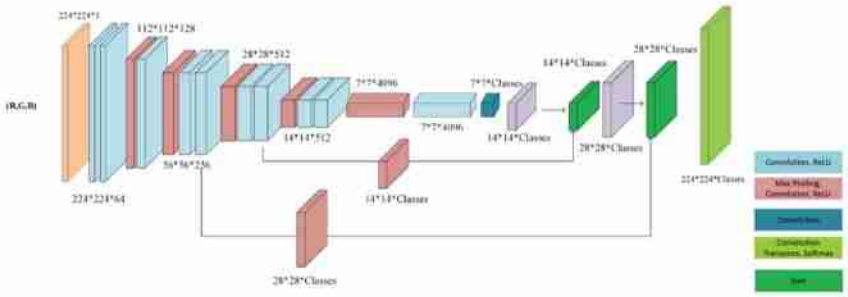


Figure 3.14: Fully Convolutional Networks Architecture [69]

The main creative step of FCNs is changing fully linked layers to convolutional layers. This change lets the network take input images of any size and make segmentation maps that keep the spatial sizes of the first image. Because of this, FCNs permit full training and quick result drawing by handling the whole image in a single go, instead of checking smaller parts on their own.

But, the first FCN-32s model made fairly rough segmentation borders because of the ongoing drop of spatial clarity triggered by pooling actions. To lessen this issue, the creators made skip paths, which mix high level meaning features from deeper layers with exact spatial info from shallower layers. This mixing plan greatly improved border finding.

Later fixes, called FCN-16s and FCN-8s, added multi level feature mixing to get gradually better segmentation results. These upgrades were quite key for medical image study, where right marking of tumor edges is vital for finding and planning treatment.

Even though FCNs have mostly been gone past by more complex designs like U-Net and DeepLab, their core design rules are still key to current segmentation models. The encoder-decoder way, with skip paths for multi-scale feature joining, still makes up the structural base of top medical image segmentation networks.[70]

3.5.4 U-Net Architecture and Variants

Regarding the segmentation of brain tumors from MRI images, U-Net has become the backbone of most modern models. The underlying idea is simple yet powerful: a dual path

network Encoder path: compresses the image through convolutional layers and pooling operations to extract global context (e.g., does this region contain a tumor or not?). Decoder path: reconstructs the pixel wise map at high resolution using upsampling, while integrating information from the encoder path via skip connections to recover fine grained details (e.g., exactly where do the tumor boundaries begin?).

The original U-Net achieved remarkable results even with very few labeled images, thanks to techniques such as elastic deformation, which artificially generate diversity without requiring additional data.

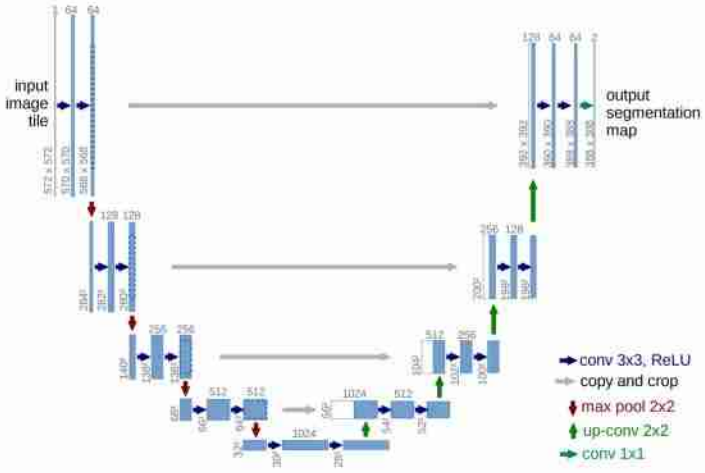


Figure 3.15: U-Net Architecture [71]

However, research hasn't stopped at the basic model. Dozens of variants have emerged, each enhancing U-Net's performance in different medical contexts:

- **3D U-Net:** designed for volumetric image segmentation (such as 3D MRIs), where convolutional operations are applied in three dimensions.
- **Attention U-Net:** incorporates attention gates to guide the network toward important regions (e.g., focusing on the tumor and ignoring healthy tissue).
- **Inception U-Net:** uses filters of varying sizes within the same layer to capture both small and large objects simultaneously.
- **Residual U-Net:** adds short connections within each block to address the vanishing gradient problem in deep networks.
- **Dense U-Net:** connects every layer to all preceding layers, improving information flow and boosting accuracy.
- **U-Net++:** builds an interwoven network of connections between the two paths, reducing the loss of semantic information.

- **Adversarial U-Net:** employs a GAN framework to refine segmentation boundaries through competition between a generator and a discriminator.
- **Cascaded U-Net:** processes the image through two consecutive networks, the first identifies the general region (e.g., the entire brain), and the second segments the tumor within that region.

All these variants weren't developed arbitrarily. Each one addresses a specific challenge in medical imaging: the scarcity of labeled data, the complexity of tumor borders, tissue overlap, or the need to efficiently process 3D images.[72]

3.6 Transformer-Based Segmentation Models

3.6.1 Vision Transformers (ViT)

Vision Transformers (ViT) change computer vision by using the Transformer design, which was first made for language processing, for image analysis. Different from CNNs, which use local areas and step by step feature finding, ViT models treat images as a series of patches and use self-attention to understand overall relationships.

In ViT, an image is broken into patches. Each patch is flattened, then changed into an embedding vector. These embeddings are put with positional encodings to keep spatial data, then sent to a Transformer encoder. This encoder has multi-head self-attention and feed-forward layers. Self-attention helps the model see long-range links across the whole image, allowing overall context to be known from the start.

A main benefit of ViTs is that they can model overall interactions without biases found in convolutional operations. Yet, this needs large datasets for good training, since Transformers don't have the spatial knowledge that CNNs do.

In medical image segmentation, Vision Transformers have shown much ability, mostly when added to designs that mix convolutional encoders with Transformer modules. These methods use the good parts of CNNs to get local texture patterns and Transformers to model overall context, which betters segmentation exactness in hard imaging cases.

In general, Vision Transformers are a change in computer vision by testing convolution methods and giving a scalable option for high-dimensional visual tasks.[73]

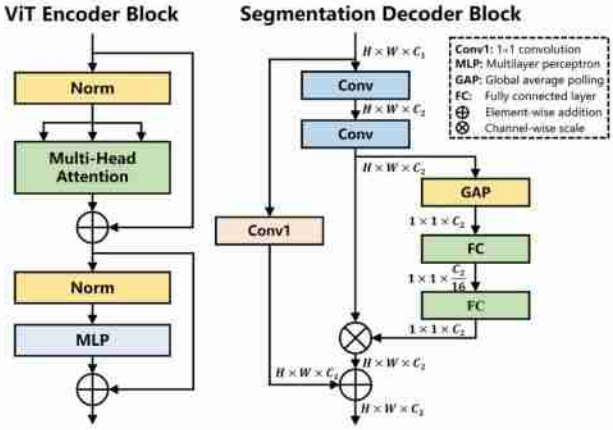


Figure 3.16: ViT Architecture [74]

3.6.2 Motivation for Transformers in Segmentation

Transformers are being used in image segmentation to address the problems that convolutional neural networks (CNNs) have with understanding long-range relationships. CNNs do well with prediction tasks, but they depend on local areas to view images, which limits their capability to understand the whole image at once. Using deeper layers can help, but it also raises costs and may cause information to be lost.

Medical image segmentation needs a good understanding of small texture details and the big picture of how things are arranged. Entities like tumors or organs might have mixed intensity patterns and weird shapes. So, knowing how things are located relative to each other is important for pinpointing their exact boundaries. Standard ways using CNNs might not fully understand context, mostly when there are unclear edges, noise, or complicated backgrounds.

Transformers solve these problems with a self-attention method, which lets direct modeling of relationships between all parts of an image. By figuring out weights across the whole image, Transformers can understand long-range relationships early on. This betters boundary accuracy and strengthens consistency, even when the image changes.

Transformers also let designs be changed and grown. Segmentation models can use pretraining on big sets of data and transfer learning methods. Designs that mix CNN encoders with Transformer modules look good because they combine local feature finding with big-picture context knowledge.

The reason Transformers are added into segmentation setups comes from a need for better learning, improved structure, and greater adaptation to visual patterns, especially in fields like medical image analysis.[73]

3.6.3 Hybrid CNN Transformer Architectures

The use of hybrid CNN-Transformer models is a helpful way to do image segmentation. They mix the strengths of convolutional neural networks (CNNs) and Transformer models. CNNs work well to find local details like edges and textures. Transformers are good at understanding long-range relationships using self-attention. Combining both lets models use local details and overall context.

This mix is helpful for medical image segmentation. To correctly outline body parts or problem areas, you need detailed boundaries and an idea of how things fit together in the image. CNN encoders take out organized feature data, and Transformer parts make these features better by creating overall relationships in the image.

There are different ways to combine these. Some models use CNNs to take out features first. Then, Transformer blocks improve the feature maps using attention. Other methods put Transformer modules inside convolutional layers to improve context understanding bit by bit. Also, encoder-decoder setups often use skip connections to keep fine spatial info, making sure the segmentation maps are put back together correctly.

Hybrid CNN-Transformer models can handle noise, intensity changes, and unclear structures better, all of which are common problems in medical images. By using CNN biases and attention-based modeling, these setups segment images better than if you just used CNNs or Transformers alone.

In short, hybrid CNN-Transformer models are a good, balance design. They give better feature representation and adapt better to complex segmentation jobs, especially with medical imaging data which is often complex and varies a lot.[73]

3.7 Evaluation Metrics

3.7.1 A Multi-Dimensional Evaluation Framework for Medical Image Segmentation

Current evaluation of AI algorithm performance in medical imaging, particularly for lesion detection, relies on single-dimensional geometric metrics such as the Dice coefficient and Jaccard Index (IoU), which measure the overlap between detected and actual regions. While useful, these metrics do not fully capture clinical complexity or ensure diagnostic reliability in practical applications. To address this limitation, this work proposes a three-dimensional evaluation framework that combines:

- **Volumetric overlap** between detected and actual lesions
- **Boundary accuracy and anatomical fidelity** of lesion contours
- **Size proportionality** between algorithmic estimates and clinical measurements

This integrated model aims to bridge computational assessment with clinical judgment, providing an objective tool to guide the development of more accurate and reliable diagnostic algorithms, particularly in electromagnetic and microwave medical imaging.[75]

3.7.2 Classification-Based Metrics

Classification-based metrics evaluate the performance of a classifier by comparing predicted labels with actual class labels using a confusion matrix. These metrics are commonly used in binary and multiclass classification problems during both training and testing stages.

Confusion Matrix

For binary classification, evaluation is derived from the confusion matrix composed of:

- True Positive (TP): correctly predicted positive instances
- True Negative (TN): correctly predicted negative instances
- False Positive (FP): negative instances incorrectly predicted as positive
- False Negative (FN): positive instances incorrectly predicted as negative

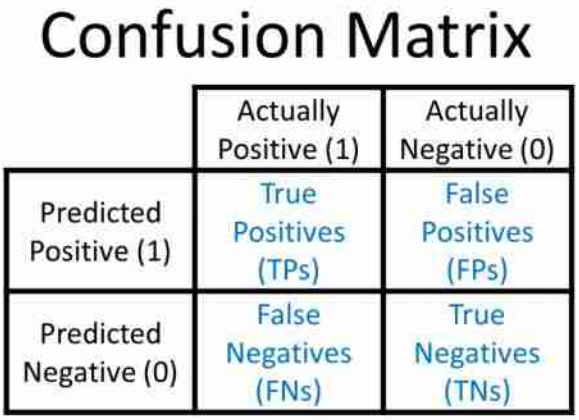


Figure 3.17: Confusion Matrix [76]

These four components form the basis for most classification-based metrics.[77]

Accuracy and Error Rate

Accuracy measures the proportion of correctly classified instances over the total number of instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

The error rate is the complement of accuracy:

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{3.2}$$

Although accuracy is simple and widely used, it may be misleading in the presence of imbalanced datasets.[77]

Sensitivity and Specificity

Sensitivity (also called Recall or True Positive Rate) measures the proportion of correctly classified positive instances[77]:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.3}$$

Specificity (True Negative Rate) measures the proportion of correctly classified negative instances [77]:

$$Specificity = \frac{TN}{TN + FP} \tag{3.4}$$

Precision and Recall

Precision measures how many predicted positive instances are actually correct:

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

Recall is equivalent to Sensitivity.[77]

F-Measure

The F1-score is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.6}$$

It balances the trade-off between false positives and false negatives.[77]

Geometric Mean (G-Mean)

The Geometric Mean balances the performance between positive and negative classes[77]:

$$G\text{-}Mean = \sqrt{Sensitivity \times Specificity} \tag{3.7}$$

Limitations

Many classification-based metrics, particularly accuracy, may lack discriminative power and can be biased toward majority classes. Therefore, relying solely on accuracy may lead to suboptimal model evaluation, especially in imbalanced classification problems.[77]

Dice Coefficient and Intersection over Union

In medical image segmentation, overlap-based metrics are commonly used to evaluate the similarity between the predicted segmentation mask and the ground truth mask.

The Dice Coefficient (Dice Similarity Coefficient, DSC) is defined as:

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{3.8}$$

The Intersection over Union (IoU), also known as the Jaccard Index, is defined as:

$$IoU = \frac{TP}{TP + FP + FN} \tag{3.9}$$

Both Dice and IoU range from 0 to 1, where 1 indicates perfect overlap between predicted and ground truth regions. Dice is generally more sensitive to small segmented structures, while IoU is slightly stricter as it penalizes mismatches more strongly. These metrics are particularly important in medical image segmentation tasks such as tumor or organ delineation.[78]

3.7.3 Metric Selection and Interpretation

There is no single metric that can adequately evaluate segmentation quality, the choice of metric depends on the clinical objective and the characteristics of the anatomical structure. While overlap based metrics like DSC and IoU have been widely used to quantify the spatial agreement between predicted and reference segmentations, they mainly measure the overlap between regions and might not accurately capture boundary inaccuracies. To complement, distance based metrics such as Hausdorff Distance and Mean Surface Distance evaluate the contour discrepancies and boundary errors. When the detection performance is clinically critical, classification metrics such as sensitivity, specificity and precision need to be considered. More importantly, the interpretation of these metrics needs to take into consideration various confounding factors including class imbalance, object size, the potential clinical impacts of false positives and false negatives. Overall, using multiple metrics, rather than a single numerical score will provide a much more accurate evaluation for the algorithm performance.[79]

3.8 State of the Art

This section presents a state-of-the-art review of research on the segmentation and classification of brain MRI images. Several representative studies using different algorithmic approaches for the analysis of brain medical images have been selected, particularly for tumor detection and diagnosis of neurodegenerative diseases.

Selected Works

In this context, [Talukder et al. (2025) [80]] proposed an attention-based convolutional U-Net model, called **ACU-Net**, for brain tumor segmentation in MRI images. The aim was to improve segmentation accuracy and computational efficiency while overcoming the limitations of traditional methods in capturing complex tumor structures. The model was evaluated on the **BraTS 2018** and **BraTS 2020** datasets, which contain multi-modal MRI scans (T1, T1c, T2, and FLAIR) with annotated tumor regions including whole tumor (WT), tumor core (TC), and enhancing tumor (ET).

A preprocessing pipeline was applied, including resizing to 128×128, denoising, intensity normalization, adaptive slicing, and data augmentation (rotation, flipping, and noise injection) to improve robustness. The proposed ACU-Net integrates convolutional layers with an attention mechanism within a U-Net architecture to focus on relevant tumor regions while preserving spatial information.

The experimental results showed excellent performance, achieving high Dice scores on both datasets: on BraTS 2018,99.23% (WT), 99.27% (TC), and 96.99% (ET), and on BraTS 2020, 98.72%, 98.40%, and 97.66%, respectively. These results outperform several state-of-the-art methods, confirming the effectiveness of attention mechanisms in U-Net for accurate brain tumor segmentation.

In this study,[Abrar et al. (2025) [81]] an attention-based U-Net model called **ACU-Net** was proposed for brain tumor segmentation from MRI images. The main objective was to improve the accuracy and robustness of tumor boundary detection by incorporating attention mechanisms into the traditional U-Net architecture.

The method was evaluated on the **BraTS 2018 dataset**, which contains multi-modal MRI scans with labeled tumor regions, including WT, TC, and ET. The data was preprocessed using normalization, resizing, and augmentation techniques such as rotation and flipping to improve model generalization.

The proposed model combines spatial and channel attention within an encoder–decoder structure with skip connections, allowing it to focus more on relevant tumor regions while reducing background noise. It was trained using a combination of Dice loss and cross-entropy loss.

The results showed strong performance, achieving Dice scores of **94.04%** for WT,

98.63% for TC, and **98.77%** for ET, outperforming standard U-Net and other CNN-based methods. Overall, ACU-Net improves segmentation performance and is effective for medical imaging tasks.

[Pourmahboubi et al. (2025) [82]] proposed a transfer learning-based U-Net model for brain tumor segmentation using VGG19 as the encoder backbone to enhance feature extraction. The method was tested on the **TCGA-LGG dataset**, which contains MRI and FLAIR images from 120 patients with lower-grade gliomas. The data was preprocessed through resizing to 256×256, normalization, splitting (70/15/15), and augmentation to improve generalization. The architecture combines the VGG19 encoder with a decoder and skip connections, and uses the Focal Tversky loss to handle class imbalance and improve small tumor segmentation. The model achieved strong results, with an AUC of 0.9957, Dice score of 0.9679, F1-score of 0.9679, precision of 0.9541, recall of 0.9821, and IoU of 0.9378. Overall, it outperformed standard U-Net models, showing that transfer learning with VGG19 improves segmentation performance.

Within this work,[Chen et al. (2024) [83]] proposed a transfer learning-based U-Net model for brain tumor segmentation using a pre-trained VGG19 encoder to enhance feature extraction and capture complex tumor structures.

The method was evaluated on the **TCGA-LGG dataset**, which contains MRI scans of 120 patients with Grade II and III gliomas, mainly using FLAIR images. Preprocessing included resizing, normalization, and dataset splitting (70/15/15), along with data augmentation to improve generalization.

The architecture combines VGG19 as encoder with a U-Net decoder and skip connections, and uses Focal Tversky loss to handle class imbalance and improve small tumor detection.

The model achieved strong results, with an AUC of 0.9957, Dice score of 0.9679, F1-score of 0.9679, precision of 0.9541, recall of 0.9821, and IoU of 0.9378, outperforming standard U-Net and demonstrating the effectiveness of transfer learning.

In this context,[Xing et al. (2025) [84]] proposed a hybrid model named Local–Global UNet Transformer (LG-UNETR) for brain tumor segmentation, aiming to improve accuracy by combining CNNs' local feature extraction with Transformers' global context modeling while reducing their respective limitations.

The approach was evaluated on the **BraTS2023-GLI** and **BraTS2024-GLI** datasets, which include multi-modal MRI scans (T1, T1Gd, T2, and T2-FLAIR) with annotations for tumor subregions such as WT, TC, and ET, presenting significant variability in tumor appearance and imaging conditions.

The model integrates convolutional layers with attention mechanisms and introduces Semantic-Oriented Masked Attention (SMA) to enhance decoder feature learning, along with Network-in-Network (NiN) blocks to improve channel representation and reduce parameter complexity.

The results showed that LG-UNETR outperformed several state-of-the-art methods, achieving a Dice score of **82.51%** and a low HD95 of **8.02 mm** on BraTS2024-GLI, demonstrating improved segmentation accuracy and boundary precision.

Within this context, [Alquran et al. (2024) [85]] proposed an improved U-Net-based model for brain tumor segmentation in MRI images, aiming to increase accuracy while reducing computational cost through architectural optimization.

The study used a public Kaggle brain tumor dataset containing 2D MRI images with annotated tumor regions, split into training (70%), validation (10%), and testing (20%) sets for evaluation.

The proposed method modifies the standard U-Net by tuning several components such as kernel size, number of channels, dropout rate, pooling strategy, and activation function. In particular, Leaky ReLU replaces ReLU to improve gradient flow and model stability, while other adjustments help reduce complexity without sacrificing performance.

The results showed that the enhanced U-Net achieved a Dice score of **90.2%** and an accuracy of **99.4%**, outperforming standard U-Net and nnU-Net. It also significantly reduced model size to 5.5 million parameters compared to 25.71 million in nnU-Net, making it more efficient and lightweight for medical applications.

Comparative Summary

Authors	**Dataset**	**Methodology**	**Results (Dice)**
Talukder et al. (2025)	BraTS 2018, BraTS 2020	ACU-Net with attention mechanism and preprocessing (resizing, normalization, augmentation)	99.23% (WT), 99.27% (TC), 96.99% (ET); 98.72%, 98.40%, 97.66%
Abrar et al. (2025)	BraTS 2018	ACU-Net with spatial and channel attention, Dice + Cross-Entropy loss	94.04% (WT), 98.63% (TC), 98.77% (ET)
Pourmahboubi et al. (2025)	TCGA-LGG	U-Net with VGG19 encoder (transfer learning), Focal Tversky loss	96.79%
Chen et al. (2024)	TCGA-LGG	U-Net with VGG19 encoder, skip connections, Focal Tversky loss	96.79%
Xing et al. (2025)	BraTS2023-GLI, BraTS2024-GLI	CNN + Transformer (LG-UNETR), SMA attention, NiN blocks	82.51%
Alquran et al. (2024)	Kaggle Brain Tumor Dataset	Improved U-Net with hyperparameter tuning and Leaky ReLU activation	90.2%

Table 3.1: Comparative analysis of brain tumor segmentation methods

Discussion

The latest research highlights the swift advancement of deep learning techniques for identifying brain tumors in MRI scans. According to recent research, the prevalence of encoder-decoder architectures like U-Net and its variations persists, with notable advancements in attention mechanisms, transfer learning, and hybrid models being made.

The integration of spatial and channel attention into attention-based models like ACU-Net leads to improved tumor boundaries and targeted areas. This is a significant improvement over previous research. In the same vein, transfer learning methods that incorporate pre-trained backbones like VGG19 improve feature extraction and performance, particularly on scarce datasets.

Recent hybrid architectures, such as CNN-Transformer models like LG-UNETR, demonstrate promising results through the use of local feature extraction and global context modeling. The use of these techniques can enhance resilience in intricate and diverse tumor types while also introducing additional computational complexity.

Dice score and related metrics are commonly used to evaluate performance across all reviewed methods, indicating consistent improvements over traditional baseline U-Net models. Nonetheless, issues such as class imbalance, variation in tumor appearance, limited annotated medical data, and computational cost persist.

Brain tumor segmentation is experiencing a shift towards efficient, hybrid, and attention-driven architectures that provide an optimum balance between accuracy, robustness (which may increase over time), and computation efficiency.

3.9 Conclusion

This chapter gave a short look at automatic medical image segmentation, covering its theory, how it has changed over time, and how it is judged. The shift from older methods to deep learning and combined CNN Transformer designs has greatly boosted how well segmentation works in medical imaging.

Even with these improvements, issues like differences in data, unequal class sizes, and how well it applies to unseen data are still present. The ideas introduced here create a base for the implementation and testing talked about in the next chapter.

Chapter 4
Experiments and Results

Chapter 4

Experiments and Results

4.1 Introduction

This chapter discusses the experimental protocol and performance analysis of the brain tumor segmentation approach proposed.The document provides a detailed account of the development environment and tools utilized, as well as an overview of datasets and preprocessing techniques. The chapters provide information on the models used and the training approach implemented. Finally, a comprehensive analysis and discussion of the obtained results are provided to evaluate the effectiveness of proposed methods.

4.2 Tools Used

Our system was built on two platforms. Kaggle was utilized to download and access the dataset, leveraging its extensive public data collection. Google Colab was utilized to run experiments and code execution, utilizing Google Drive as the foundation for a user friendly cloud environment for model training and evaluation. Additionally, Google Colab provides access to hardware accelerators like GPUs and TPUs, which can significantly enhance computational efficiency and shorten the time required to train deep learning models.

4.2.1 Google Colab (Colaboratory)

Google Colab is a free cloud-based platform where you can write and execute Python code in a web browser without any local setup. It provides free usage of computing resources like GPUs, which is quite handy for a student, researcher, or anybody who would like to perform basic machine learning/data science tasks.[86]
Similar to Jupyter Notebook, it provides the user the ability to create notebooks, which can contain code, text, and multimedia files such as HTML, LaTeX, and images. It provides collaboration features where the person with whom the user shares the link to

the notebook can view it as well as comment and edit the notebook. Finally, it has a built-in saving functionality using Google Drive.[86]

Figure 4.1: Logo of Google Colab [87]

4.2.2 Kaggle

Kaggle is an online platform and community for data science and machine learning that allows users to find and publish datasets, develop models, and run notebooks using GPU support. Founded by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017, Kaggle is widely used for hosting and participating in competitions, where users of all levels can solve real-world problems individually or in teams. The platform also provides an integrated Jupyter Notebook environment that runs directly in the browser, eliminating the need for local hardware or setup. In addition, Kaggle hosts a large collection of datasets and allows users to upload and share their own, making it a comprehensive environment for experimentation and collaboration.[88]

Figure 4.2: Logo of Kaggle [89]

4.2.3 Programming Language (Python)

The use of Python is widespread in various fields, including web development and data science, ML, and AI . Open source, easy to learn, cross platform compatible, with a large library community and strong community backing. The language was initially developed by Guido van Rossum in 1989 and has been updated multiple times, with Python 3 becoming the most popular variant available today.[90]

Figure 4.3: Logo of Python [91]

4.2.4 Libraries

- **TensorFlow :**
  TensorFlow, a ML and numerical computation library, was officially released under the Apache 2.0 license by Google Brain in 2015 and is open source. It finds extensive usage in both DNN and predictive analytics. By making the training process less complex and more efficient, developers can quickly develop and deploy models for various applications such as classification, image recognition, and NLP.

  TensorFlow relies on data flow graphs, which exhibit computations as nodes performing mathematical operations and contain a collection of tensitories. This approach is supported by the concept of "data flow".[92]

Figure 4.4: TensorFlow's Logo [93]

- **Keras :**
  An open source Python DL API called Keras is designed to facilitate rapid experimentation with neural networks, and it is both high level and user friendly. As a tool for TensorFlow, JAX, or PyTorch integration it is designed to debug quickly and make code easy to deploy. Keras facilitates the development, training, and deployment of computer vision and natural language processing models. Moreover, The ONEIROS project included the development of Keras, which was released in 2015 and was developed by François Chollet. TensorFlow has been extensively integrated with it.[94]

Figure 4.5: Logo of Keras [95]

- **Matplotlib :**
  Matplotlib is a comprehensive, open source library for the Python programming language used to create static, animated, and interactive visualizations. It serves as a foundational data visualization tool, designed for 2D plotting of arrays and integrating with NumPy and SciPy, making it ideal for data science and scientific computing.[96]

Figure 4.6: Logo of Matplotlib [97]

- **NumPy :**
  NumPy is the primary open source Python library for scientific computing, designed to handle large, multi-dimensional arrays and matriceS. This object is essential for high performance numerical computation and is used in data science, ML (e.g. Scikit-learn/tensorFlow), and visualization.[98]

Figure 4.7: Logo of NumPy [99]

- **Pycocotools :**
  pycocotools (Python COCO API) is a Python library designed to interact with, parse, and visualize annotations in the Microsoft Common Objects in Context (COCO) dataset format. It is the standard tool used by computer vision researchers and engineers to load, evaluate, and analyze data for object detection, segmentation, and captioning tasks.[100]

4.3 Dataset Used

Description

The dataset used in this work is a Brain Tumor MRI dataset for semantic segmentation, available on Kaggle [101]. It contains medical brain images annotated at the pixel level to identify tumor regions. The dataset is designed for semantic segmentation tasks, where each pixel is classified into one of two classes: **tumor (Class 1)** or **non-tumor (Class 0)**.

The images were resized to **640 × 640 pixels**, and the annotations are provided in COCO format, allowing the generation of binary segmentation masks. The dataset includes variations in tumor size, shape, and location, making it suitable for evaluating deep learning models in medical image segmentation.

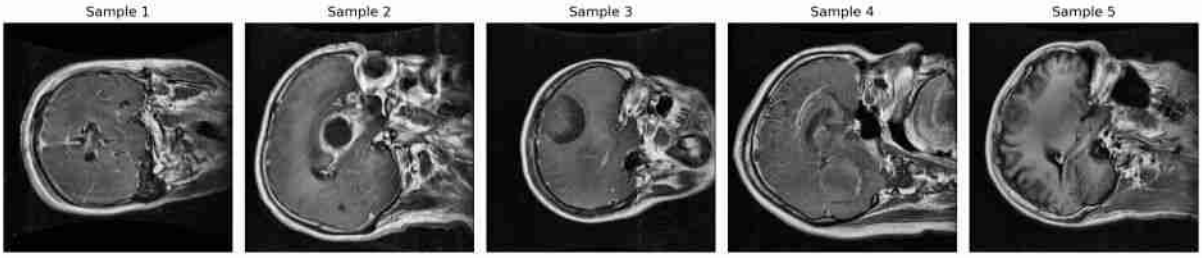


Figure 4.8: Visualization of Sample Images from the Brain Tumor Dataset

4.4 Proposed Methods

4.4.1 Vision Transformer (ViT)

The original ViT architecture was primarily designed for image classification tasks. In this work, the model was adapted for medical image segmentation by replacing the final classification head with a segmentation decoder. The Transformer encoder extracts global contextual features from MRI images using self-attention mechanisms, while the decoder progressively restores spatial information to generate the final brain tumor segmentation mask.

The architecture is composed of the following primary components:

Patch Embedding

The input image is divided into fixed-size patches using a convolutional layer, where both the kernel size and stride are equal to the patch size. Each patch is then projected into a high-dimensional embedding space of dimension `embed_dim`. The spatial structure of the image is thus converted into a sequence of patch tokens.

Main function: Transform the 2D image into a sequence representation suitable for transformer processing.

Transformer Encoder

The transformer encoder consists of multiple stacked transformer blocks. Each block includes:

- **Multi-Head Self-Attention (MHSA):** Captures global dependencies between image patches.

- **Layer Normalization:** Stabilizes and accelerates training.
- **Dropout:** Helps reduce overfitting.
- **MLP (Feed-Forward Network):** Composed of two fully connected layers with GELU activation.

Residual connections are applied after both the attention and MLP sublayers to improve gradient flow and model performance.

Main function: Learn rich global representations and long-range dependencies within the image.

Reshaping Layer

The sequence of encoded patch representations is reshaped back into a 2D feature map. The spatial dimensions are reconstructed according to the original image size and patch configuration.

Main function: Prepare the encoded features for the decoding (upsampling) stage.

Decoder (Upsampling Path)

The decoder is composed of several upsampling blocks. Each block includes:

- Bilinear upsampling to increase spatial resolution.
- Two convolutional layers followed by Batch Normalization and ReLU activation.

The number of filters decreases progressively throughout the decoder.

Main function: Reconstruct the segmentation map while recovering spatial resolution.

Output Layer

The final layer is a convolutional layer followed by a sigmoid activation function. It produces a pixel wise segmentation mask, where each pixel is assigned a probability indicating the presence of the target class.

Main function: Generate the final segmentation output.

Stage	Layer Type	Input Size	Activation
Input	Input Layer	$256 \times 256 \times 3$	–
Patch Embedding	Conv2D (patch_size×patch_size, stride=patch_size) + Reshape	$N \times embed\_dim$	–
Position Encoding	Add positional embeddings	$N \times embed\_dim$	–
Dropout	Dropout	$N \times embed\_dim$	–
Transformer - Block 1	MHA + MLP + Add & Norm	$N \times embed\_dim$	GELU
Transformer - Block 2	MHA + MLP + Add & Norm	$N \times embed\_dim$	GELU
...	(Repeated *num_blocks* times)	$N \times embed\_dim$	GELU
Normalization	LayerNormalization	$N \times embed\_dim$	–
Reconstruction	Reshape (tokens → feature map)	$H \times W \times embed\_dim$	–
Decoder - Block 1	UpSampling + Conv2D(f_1) ×2 + BN	$2H \times 2W \times f_1$	ReLU
Decoder - Block 2	UpSampling + Conv2D(f_2) ×2 + BN	$4H \times 4W \times f_2$	ReLU
Decoder - Block 3	UpSampling + Conv2D(f_3) ×2 + BN	$8H \times 8W \times f_3$	ReLU
Decoder - Block 4	UpSampling + Conv2D(f_4) ×2 + BN	$16H \times 16W \times f_4$	ReLU
Output	Conv2D(1, 1×1)	$256 \times 256 \times 1$	Sigmoid

Table 4.1: Vision Transformer (ViT) Encoder-Decoder Architecture

4.4.2 U-Net

The second proposed model is a customized implementation of the U-Net architecture, a convolutional neural network widely used for image segmentation tasks. It follows an encoder-decoder structure with skip connections that allow precise localization while preserving contextual information. The model consists of the following main components:

Encoder

The encoder is composed of four convolutional blocks. Each block applies:

- Two convolutional layers (3×3 kernel, padding = "same")
- Batch Normalization
- ReLU activation

A MaxPooling layer follows each block to reduce spatial dimensions. The number of filters increases progressively: $32 \rightarrow 64 \rightarrow 128 \rightarrow 256$. Dropout is applied in deeper layers to reduce overfitting.

Main function: extract hierarchical features and capture contextual information from the input image.

Bottleneck

This is the deepest part of the network. It consists of a convolutional block with 512 filters and dropout. No pooling is applied at this stage.

Main function: learn high level, abstract representations of the image features.

Decoder

The decoder is composed of four upsampling blocks. Each block includes:

- UpSampling2D to increase spatial resolution
- Concatenation with corresponding encoder features (skip connections)
- A convolutional block (two Conv2D + BatchNorm + ReLU)

The number of filters decreases progressively: $256 \rightarrow 128 \rightarrow 64 \rightarrow 32$. Skip connections help recover spatial details lost during downsampling.

Main function: reconstruct the segmentation map while preserving fine-grained details.

Skip Connections

Feature maps from the encoder are concatenated with the decoder at corresponding levels. This helps retain spatial information and improves segmentation accuracy.

Main function: combine low level spatial features with high level semantic information.

Output Layer

A final Conv2D layer (1×1 kernel) with sigmoid activation is used. It produces a binary segmentation mask, where each pixel is assigned a probability indicating the presence of the target region.

Stage	Layer Type	Input Size	Activation
Input	Input Layer	$256 \times 256 \times 3$	–
Encoder - Block 1	Conv2D(32) + BN + ReLU ×2	$256 \times 256 \times 32$	ReLU
Encoder - Block 2	Conv2D(64) + BN + ReLU ×2	$128 \times 128 \times 64$	ReLU
Encoder - Block 3	Conv2D(128) + BN + ReLU ×2	$64 \times 64 \times 128$	ReLU
Encoder - Block 4	Conv2D(256) + BN + ReLU ×2 + Dropout(0.2)	$32 \times 32 \times 256$	ReLU
Bottleneck	Conv2D(512) + BN + ReLU ×2 + Dropout(0.3)	$16 \times 16 \times 512$	ReLU
Decoder - Block 1	UpSampling2D + Concat + Conv2D(256) + BN + ReLU ×2	$32 \times 32 \times 256$	ReLU
Decoder - Block 2	UpSampling2D + Concat + Conv2D(128) + BN + ReLU ×2	$64 \times 64 \times 128$	ReLU
Decoder - Block 3	UpSampling2D + Concat + Conv2D(64) + BN + ReLU ×2	$128 \times 128 \times 64$	ReLU
Decoder - Block 4	UpSampling2D + Concat + Conv2D(32) + BN + ReLU ×2	$256 \times 256 \times 32$	ReLU
Output	Conv2D(1, 1×1)	$256 \times 256 \times 1$	Sigmoid

Table 4.2: U-Net Model Architecture with Activation Function

4.4.3 Attention U-Net

The third proposed model is an enhanced version of the U-Net architecture known as Attention U-Net. This model integrates attention mechanisms into the skip connections, allowing the network to focus on the most relevant regions of the image while suppressing irrelevant background information. It improves segmentation performance, especially in complex medical imaging tasks.

Encoder

The encoder consists of multiple convolutional blocks. Each block includes:

- Two convolutional layers (3×3 kernel, padding = "same")
- Batch Normalization
- ReLU activation

A MaxPooling layer is applied after each block to reduce spatial dimensions. The number of filters increases progressively: $32 \rightarrow 64 \rightarrow 128 \rightarrow 256$. Dropout is applied in deeper layers to prevent overfitting.

Main function: extract hierarchical and contextual features from the input image.

Bottleneck

This is the deepest part of the network. It consists of a convolutional block with 512 filters and dropout. No pooling is applied at this stage.

Main function: capture high-level, abstract feature representations.

Attention Gates

Attention gates are applied to the skip connections before concatenation. They take two inputs:

- Encoder feature map (skip connection)
- Decoder feature map (gating signal)

The mechanism involves:

- Linear transformations using 1×1 convolutions
- Element-wise addition and ReLU activation
- A sigmoid activation to generate attention coefficients

The encoder features are multiplied by these coefficients to emphasize relevant regions.

Main function: highlight important spatial features and suppress irrelevant information.

Decoder

The decoder consists of multiple upsampling blocks. Each block includes:

- UpSampling2D to increase spatial resolution
- Attention-gated skip connection
- Concatenation of features
- A convolutional block (two Conv2D + BatchNorm + ReLU)

The number of filters decreases progressively: $256 \rightarrow 128 \rightarrow 64 \rightarrow 32$.

Main function: reconstruct the segmentation map with improved focus on relevant regions.

Skip Connections with Attention

Unlike standard U-Net, skip connections are filtered using attention gates. This ensures that only the most relevant encoder features are passed to the decoder.

Main function: improve feature fusion and reduce noise in segmentation.

Output Layer

A final Conv2D layer (1×1 kernel) with sigmoid activation is used. It produces a binary segmentation mask, where each pixel is assigned a probability representing the target class.

Stage	Layer Type	Input Size	Activation
Input	Input Layer	$256 \times 256 \times 3$	–
Encoder – Block 1	Conv2D(32) + BN + ReLU $\times 2$	$256 \times 256 \times 32$	ReLU
Encoder – Block 2	Conv2D(64) + BN + ReLU $\times 2$	$128 \times 128 \times 64$	ReLU
Encoder – Block 3	Conv2D(128) + BN + ReLU $\times 2$	$64 \times 64 \times 128$	ReLU
Encoder – Block 4	Conv2D(256) + BN + ReLU $\times 2$ + Dropout(0.2)	$32 \times 32 \times 256$	ReLU
Bottleneck	Conv2D(512) + BN + ReLU $\times 2$ + Dropout(0.3)	$16 \times 16 \times 512$	ReLU
Decoder – Block 1	UpSampling2D + Attention Gate + Concat + Conv2D(256) + BN + ReLU $\times 2$	$32 \times 32 \times 256$	ReLU
Decoder – Block 2	UpSampling2D + Attention Gate + Concat + Conv2D(128) + BN + ReLU $\times 2$	$64 \times 64 \times 128$	ReLU
Decoder – Block 3	UpSampling2D + Attention Gate + Concat + Conv2D(64) + BN + ReLU $\times 2$	$128 \times 128 \times 64$	ReLU
Decoder – Block 4	UpSampling2D + Attention Gate + Concat + Conv2D(32) + BN + ReLU $\times 2$	$256 \times 256 \times 32$	ReLU
Output	Conv2D(1, 1×1)	$256 \times 256 \times 1$	Sigmoid

Table 4.3: Attention U-Net Model Architecture with Activation Function

4.4.4 DeepLabV3+

The fourth proposed model is based on DeepLabV3+, a state-of-the-art architecture for semantic image segmentation. This model combines a powerful convolutional backbone with ASPP and an efficient decoder to capture multi-scale contextual information while preserving fine spatial details.

Backbone Network

A pre-trained ResNet50 model (trained on ImageNet) is used as the encoder. Intermediate feature maps are extracted from:

- A deep layer (`conv4_block6_out`) for high-level semantic features
- A shallow layer (`conv2_block3_out`) for low-level spatial details

The low-level features are processed using a 1×1 convolution (48 filters) followed by Batch Normalization and ReLU activation.

Main function: extract rich hierarchical features at different levels of abstraction.

Atrous Spatial Pyramid Pooling (ASPP)

The ASPP module captures multi-scale contextual information using parallel branches:

- A 1×1 convolution
- Three 3×3 atrous (dilated) convolutions with dilation rates: 6, 12, and 18
- A global average pooling branch followed by upsampling

The outputs of all branches are concatenated and passed through a 1×1 convolution, Batch Normalization, and ReLU activation.

Main function: capture features at multiple scales and enlarge the receptive field without reducing spatial resolution.

Decoder

The ASPP output is upsampled using bilinear interpolation. It is then concatenated with the low level features extracted from the backbone. The combined features are refined using:

- Two convolutional layers (3×3 kernel)
- Batch Normalization
- ReLU activation

Main function: recover fine spatial details and improve segmentation accuracy.

Upsampling Layers

Bilinear upsampling is applied twice:

- First, to align ASPP output with low level features
- Second, to restore the final segmentation map to the original image size

Main function: progressively reconstruct high-resolution feature maps.

Output Layer

A final Conv2D layer (1×1 kernel) with sigmoid activation is used. It produces a binary segmentation mask, where each pixel is assigned a probability indicating the presence of the target class.

Branch	Operation	Kernel	Rate	Channels	Size	Purpose
y_1	Conv2D + BN + ReLU	1×1	1	256	16×16	Capture local features
y_2	Conv2D + BN + ReLU	3×3	6	256	16×16	Medium receptive field
y_3	Conv2D + BN + ReLU	3×3	12	256	16×16	Larger context
y_4	Conv2D + BN + ReLU	3×3	18	256	16×16	Very large context
y_5	Global Avg Pool $\rightarrow$ Conv $\rightarrow$ UpSample	–	–	256	16×16	Global context
Concat	Merge all branches	–	–	1280	16×16	Multi-scale fusion
Output	Conv2D + BN + ReLU	1×1	–	256	16×16	Feature refinement

Table 4.4: Atrous Spatial Pyramid Pooling (ASPP) Module Architecture

Stage	Layer / Block	Output Shape	Filters	Operation	Description
Input	Input Image	$256 \times 256 \times 3$	–	–	RGB image
Encoder	ResNet50 Backbone	–	–	Conv blocks	Feature extraction
Low-Level Features	conv2_block3_out	$64 \times 64 \times 256$	48	1×1 Conv + BN + ReLU	Spatial details
High-Level Features	conv4_block6_out	$16 \times 16 \times 1024$	–	–	Semantic features
ASPP	Multi-scale module	$16 \times 16 \times 256$	256	Dilated Conv	Context aggregation
Upsampling	Bilinear Upsample	$64 \times 64 \times 256$	–	$\times 4$	Match low-level size
Fusion	Concatenation	$64 \times 64 \times 304$	–	–	Combine features
Decoder Conv1	Conv + BN + ReLU	$64 \times 64 \times 256$	256	3×3	Feature refinement
Decoder Conv2	Conv + BN + ReLU	$64 \times 64 \times 256$	256	3×3	More refinement
Final Upsample	Bilinear Upsample	$256 \times 256 \times 256$	–	$\times 4$	Restore resolution
Output	Conv2D (Sigmoid)	$256 \times 256 \times 1$	1	1×1	Segmentation mask

Table 4.5: DeepLabV3+ Model Architecture

4.4.5 TransUNet

The fifth proposed model is TransUNet, a hybrid architecture that combines the strengths of CNNs and transformer-based models. It integrates a U-Net like structure with transformer blocks at the bottleneck to capture both local spatial features and global contextual dependencies.

Encoder

The encoder follows a U-Net style structure composed of convolutional blocks. Each block includes:

- Two convolutional layers (3×3 kernel, padding = "same")
- Batch Normalization
- ReLU activation

Each block is followed by a MaxPooling layer to reduce spatial dimensions. The number of filters increases progressively: $32 \rightarrow 64 \rightarrow 128 \rightarrow 256$.

Main function: extract hierarchical local features and reduce spatial resolution.

Bottleneck (CNN + Transformer Integration)

A convolutional block with 512 filters is first applied. The feature map is reshaped into a sequence of tokens. Several transformer blocks are applied to this sequence. Each transformer block includes:

- Layer Normalization
- Multi-Head Self-Attention (MHSA)
- Residual connections
- Feed-forward network (Dense layers with dropout)

After transformer processing, the sequence is reshaped back into a 2D feature map.

Main function: capture long-range dependencies and global context.

Transformer Blocks

Each block consists of:

- Multi-Head Attention mechanism for global feature interaction
- Feed-forward neural network for feature transformation
- Residual connections for stable training

Main function: enhance feature representation by modeling relationships between distant regions in the image.

Decoder

The decoder mirrors the encoder structure. Each block includes:

- UpSampling2D to increase spatial resolution
- Concatenation with corresponding encoder features (skip connections)
- A convolutional block (two Conv2D + BatchNorm + ReLU)

The number of filters decreases progressively: $256 \rightarrow 128 \rightarrow 64 \rightarrow 32$.

Main function: reconstruct the segmentation map while recovering spatial details.

Skip Connections

Feature maps from the encoder are concatenated with decoder features at corresponding levels. These connections preserve fine grained spatial information lost during downsampling.

Main function: improve localization accuracy and segmentation quality.

Output Layer

A final Conv2D layer (1×1 kernel) with sigmoid activation is used. It produces a binary segmentation mask, where each pixel is assigned a probability indicating the presence of the target class.

Stage	Layer Type	Input Size	Activation
Input	Input Layer	$256 \times 256 \times 3$	–
Encoder – Block 1	Conv2D(32) $\times 2$ + BN	$256 \times 256 \times 32$	ReLU
Encoder – Block 2	Conv2D(64) $\times 2$ + BN	$128 \times 128 \times 64$	ReLU
Encoder – Block 3	Conv2D(128) $\times 2$ + BN	$64 \times 64 \times 128$	ReLU
Encoder – Block 4	Conv2D(256) $\times 2$ + BN	$32 \times 32 \times 256$	ReLU
Bottleneck	Conv2D(512) $\times 2$ + BN	$16 \times 16 \times 512$	ReLU
Transformation	Reshape (Flatten spatial)	256×512	–
Transformer $\times 4$	MHA + FFN + Add & Norm	256×512	ReLU
Reconstruction	Reshape	$16 \times 16 \times 512$	–
Decoder – Block 1	UpSampling + Concat + Conv2D(256) $\times 2$	$32 \times 32 \times 256$	ReLU
Decoder – Block 2	UpSampling + Concat + Conv2D(128) $\times 2$	$64 \times 64 \times 128$	ReLU
Decoder – Block 3	UpSampling + Concat + Conv2D(64) $\times 2$	$128 \times 128 \times 64$	ReLU
Decoder – Block 4	UpSampling + Concat + Conv2D(32) $\times 2$	$256 \times 256 \times 32$	ReLU
Output	Conv2D(1, 1×1)	$256 \times 256 \times 1$	Sigmoid

Table 4.6: Proposed Hybrid U-NetTransformer Architecture

4.5 Training and Hyperparameters

This section provides information on the hyperparameters and training configuration of all proposed models. All architectures had the same training settings as part of a consistent approach to ensure fairness in comparison.

Data Split

The dataset contains a total of **2146 images**, divided into three subsets: **1502 images (70%)** for training, **429 images (20%)** for validation, and **215 images (10%)** for testing. Each subset includes images along with their corresponding COCO annotation files used to generate segmentation masks.

Data Preprocessing

All input images were resized to a fixed resolution of 256×256 pixels. The pixel values were normalized to the range $[0, 1]$ by dividing by 255 to facilitate stable training.

The corresponding segmentation masks were also resized to 256×256 using nearest neighbor interpolation to preserve label integrity. The masks were then converted into binary format, where pixel values greater than 0.5 were assigned to 1 and the rest to 0.

```
def preprocess(image, mask):

    image = tf.image.resize(image, (256, 256))
    if len(mask.shape) == 2:
        mask = tf.expand_dims(mask, axis=-1)
```

```
6
7       mask = tf.image.resize(mask, (256, 256), method='nearest')
8       mask = tf.cast(mask, tf.float32)
9
10      mask = tf.cast(mask > 0.5, tf.float32)
11      image = tf.cast(image, tf.float32) / 255.0
12
13      return image, mask
```

Listing 4.1: Data Preprocessing Function

Data Augmentation

To improve model generalization and reduce overfitting, several data augmentation techniques were applied to the training dataset. These transformations are performed randomly to increase data variability while preserving the correspondence between images and their segmentation masks.

The applied augmentations include:

- Random horizontal flipping with a probability of 0.5
- Random vertical flipping with a probability of 0.5
- Random rotation by multiples of 90° (0°, 90°, 180°, 270°)
- Random brightness adjustment with a small variation factor

All geometric transformations are applied to both the image and its corresponding mask to maintain spatial consistency, while brightness adjustment is applied only to the image.

```
1  def augment(image, mask):
2
3      if tf.random.uniform(()) > 0.5:
4          image = tf.image.flip_left_right(image)
5          mask = tf.image.flip_left_right(mask)
6
7      if tf.random.uniform(()) > 0.5:
8          image = tf.image.flip_up_down(image)
9          mask = tf.image.flip_up_down(mask)
10
11     k = tf.random.uniform(shape=[], minval=0, maxval=4, dtype=tf.int32)
12     image = tf.image.rot90(image, k)
13     mask = tf.image.rot90(mask, k)
14
15     image = tf.image.random_brightness(image, max_delta=0.1)
16
```

```
17      return image, mask
```

Listing 4.2: Data Augmentation Function

Training Configuration

The models were trained using the TensorFlow data pipeline with efficient data loading and parallel processing. The training dataset was repeated indefinitely to ensure continuous batch generation.

The following hyperparameters were used:

- Batch size: 8
- Number of epochs: 200
- Steps per epoch: 188
- Learning rate: 5×10^{-5}

Optimizer

The Adam optimizer was used for training due to its adaptive learning rate capabilities and fast convergence. The initial learning rate was set to 5×10^{-5}.

Loss Function

A combined loss function was employed to improve segmentation performance by balancing region overlap and pixel-wise accuracy. The loss is defined as:

```
1  def dice_coef(y_true, y_pred, smooth=1e-6):
2      y_true_f = K.flatten(y_true)
3      y_pred_f = K.flatten(y_pred)
4      intersection = K.sum(y_true_f * y_pred_f)
5      return (2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(
           y_pred_f) + smooth)
6
7
8  def dice_loss(y_true, y_pred):
9      return 1 - dice_coef(y_true, y_pred)
10
11
12 def combined_loss(y_true, y_pred):
13     dice = dice_loss(y_true, y_pred)
14     bce = tf.keras.losses.binary_crossentropy(y_true, y_pred)
15     return 0.7 * dice + 0.3 * bce
```

Listing 4.3: Loss Functions

This combination helps address class imbalance while ensuring precise boundary prediction.

Evaluation Metric

The Dice coefficient was used as the primary evaluation metric during training, as it is well-suited for medical image segmentation tasks and measures the overlap between predicted and ground truth masks.

Training Strategy

To enhance training stability and prevent overfitting, the following strategies were applied:

- **Early Stopping**: Training was stopped if the validation loss did not improve for 15 consecutive epochs, and the best model weights were restored.
- **Learning Rate Scheduling**: The learning rate was reduced by a factor of 0.5 when the validation loss plateaued for 5 epochs.
- **Model Checkpointing**: A checkpointing mechanism was employed to save the best-performing model during training. The model weights were stored whenever an improvement in validation loss was observed, ensuring that the optimal model parameters were preserved.

These techniques ensured efficient convergence and improved generalization performance across all models.

4.6 Results and Discussion

4.6.1 Evaluation Metrics

The quality of segmentation is evaluated by Dice Similarity Coefficient (DSC), also known as Dice Score. DSC is utilized to measure how well the model performs and measures the overlap between predicted segmentation and ground truth. DSC is commonly used in medical image segmentation because it is very effective in measuring the overlap.

The Dice Score is defined as:

$$\text{Dice} = \frac{2 \times |P \cap G|}{|P| + |G|}$$

where P represents the set of predicted pixels and G represents the set of ground truth pixels.

Dice Score measures how similar two segmentations are; it ranges between 0 and 1, where 1 means the predicted segmentation fully matches the true label and 0 means there is no overlap between the prediction and the actual region. This metric is used because it evaluates the similarity of regions and performs well when one class is a minority, which is common in medical images where lesions occupy a small proportion, and provides a clear evaluation of segmentation performance in this field. Other metrics can also be used, such as IoU and precision-recall; however, the Dice score is more intuitive for segmentation assessment in this area.

4.6.2 Quantitative Results

The performance of the models was evaluated on a test dataset using two metrics: the Dice coefficient and the loss value. These metrics respectively reflect the accuracy of segmentation and the optimality of model training. The obtained results are summarized in Table 4.7.

Table 4.7: Quantitative results of different models on the test dataset

Model	**Dice Coefficient**	**Test Loss**
Vision Transformer (ViT)	0.6636	0.2687
U-Net	0.7639	0.1889
Attention U-Net	0.7495	0.2002
DeepLabV3+	**0.7840**	**0.1742**
TransUNet	0.7587	0.1933

Among all the models, ViT achieved the lowest performance with a Dice coefficient of 0.6636 and a loss value of 0.2687. This can be attributed to the fact that Vision Transformers generally require large-scale datasets to effectively learn feature representations.

In contrast, U-Net achieved a Dice coefficient of 0.7639 and a loss of 0.1889, significantly improving segmentation performance. This is mainly due to its strong ability to capture spatial features through skip connections.

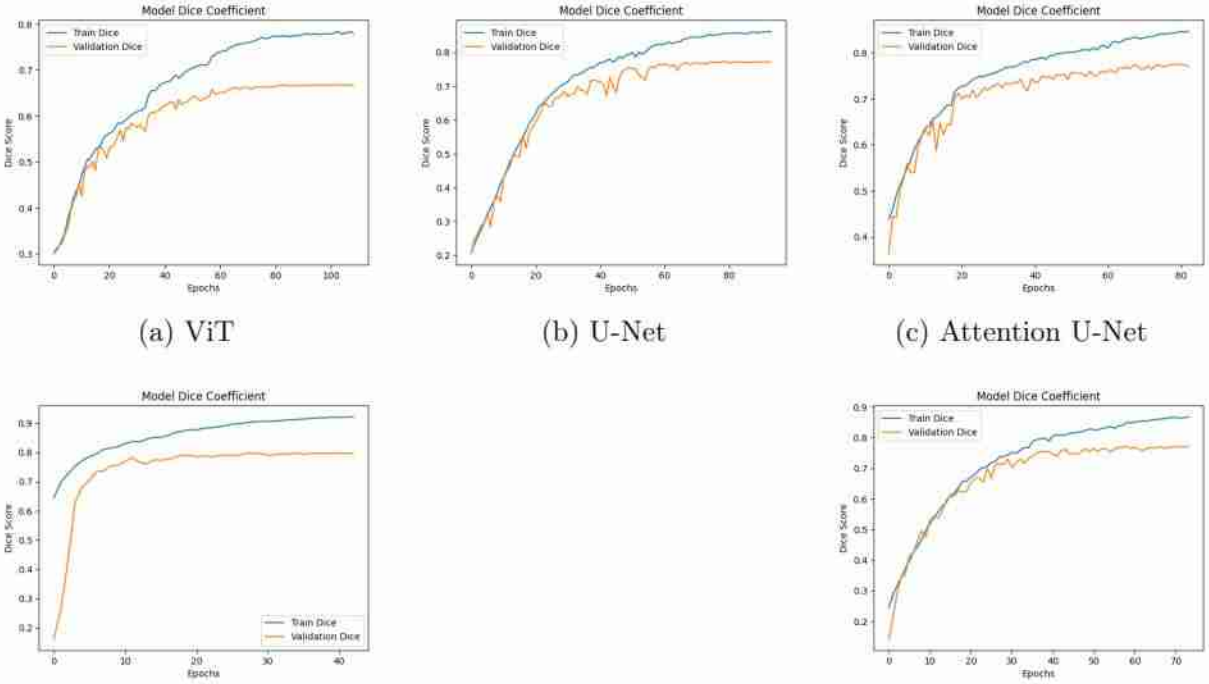
Attention U-Net obtained competitive results with a Dice coefficient of 0.7495 and a loss of 0.2002, slightly lower than the standard U-Net.

DeepLabV3+ achieved the best performance overall, with a Dice coefficient of 0.7840 and the lowest loss value of 0.1742, demonstrating its effectiveness in capturing multi-scale contextual information.

Finally, TransUNet also showed strong performance, reaching a Dice coefficient of 0.7587 and a loss of 0.1933, benefiting from the combination of Transformer-based global feature learning and convolutional-based local feature extraction.

In addition to the final evaluation metrics, the training dynamics of the models are illustrated through learning curves, including the evolution of loss, Dice Coefficient. These curves provide insight into convergence behavior, stability, and potential overfitting.

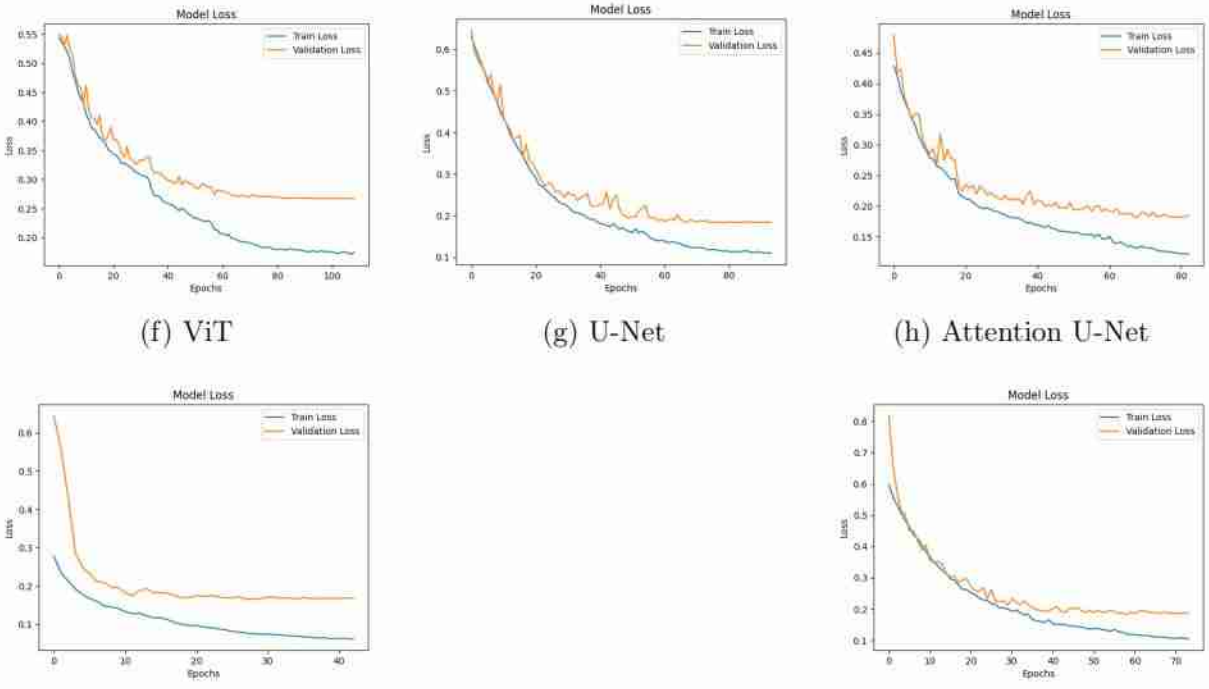
(a) Dice Coefficient (Training vs Validation)



(a) ViT (b) U-Net (c) Attention U-Net

(d) DeepLabV3+ (e) TransUNet

(b) Loss (Training vs Validation)



(f) ViT (g) U-Net (h) Attention U-Net

(i) DeepLabV3+ (j) TransUNet

Figure 4.9: Training and validation curves of Dice Coefficient and loss for all evaluated models

Figure 4.9 presents the learning curves of the different models, including the evolution of the Dice coefficient (a-e) as well as the loss function (f-j) over time.

According to these curves, the U-Net and DeepLabV3+ models are characterized by fast and stable convergence, reflected by a steady increase in the Dice coefficient and a gradual decrease in the loss function, with few fluctuations between the training and validation phases.

The TransUNet model also shows satisfactory behavior, although slightly less stable, with some variations observed in the validation curves.

The Attention U-Net model demonstrates generally correct convergence, but with more pronounced fluctuations, particularly in the validation Dice score, indicating slightly lower stability compared to the classical U-Net.

On the other hand, the ViT model presents slower convergence as well as greater variability, reflecting difficulties in effectively learning from the considered dataset.

Furthermore, the relatively small gap between training and validation curves for most models suggests limited overfitting, which confirms the effectiveness of the adopted strategies, notably the use of early stopping.

4.6.3 Qualitative Results

In this section, qualitative results are displayed for visual inspection. In particular, for representative samples from the dataset, results are presented by overlaying the predicted segmentation masks with the corresponding ground truth annotations. In each case, four different parts are shown: the original input MRI image, the corresponding ground truth mask, the predicted segmentation obtained with the model, and the overlay where the predicted mask is superimposed on the input image to show where the tumor has been located and aligned with ground truth. This representation facilitates a clear visual assessment of the model's ability to localize the tumor region and align it with ground truth.

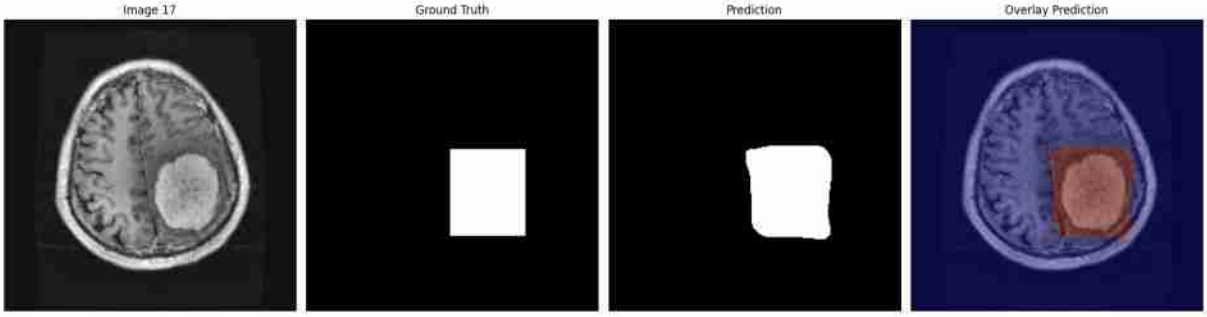


Figure 4.10: Qualitative Segmentation Results for ViT

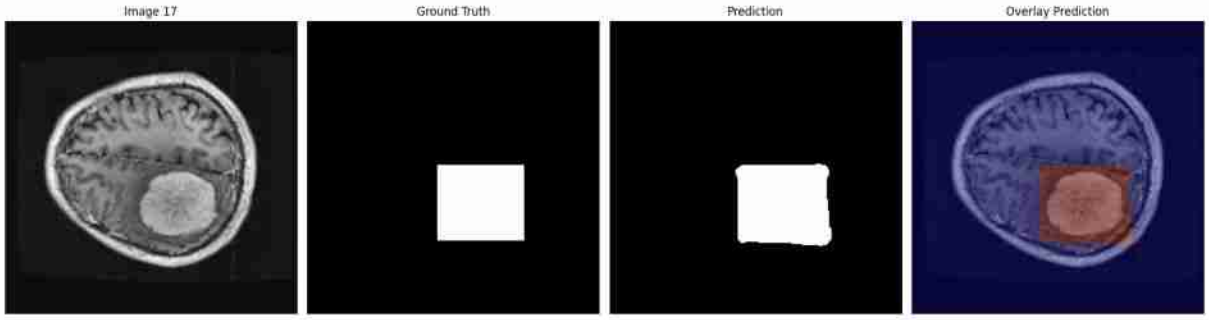


Figure 4.11: Qualitative Segmentation Results for U-Net

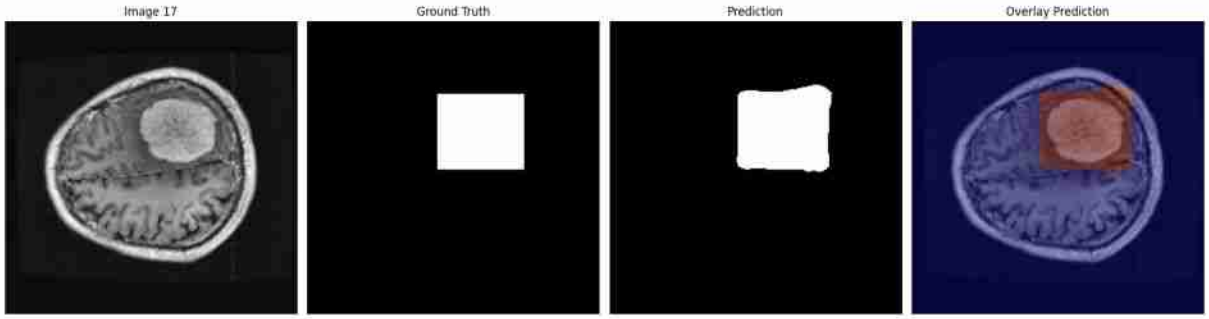


Figure 4.12: Qualitative Segmentation Results for Attention U-Net

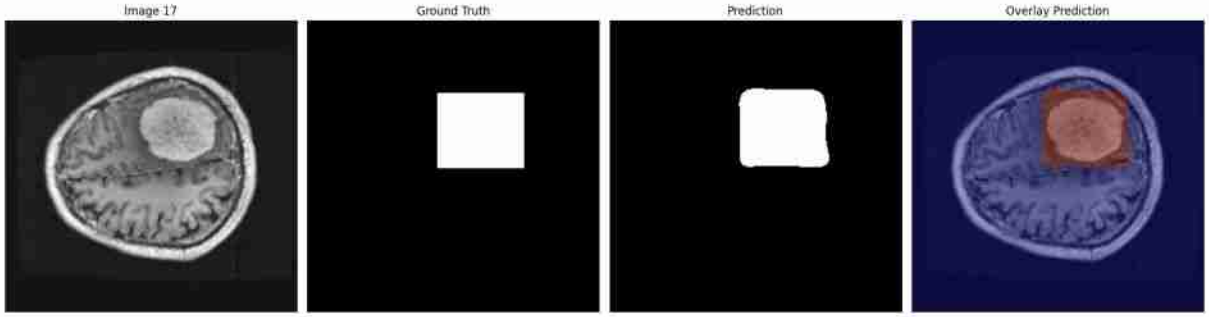


Figure 4.13: Qualitative Segmentation Results for DeepLabV3+

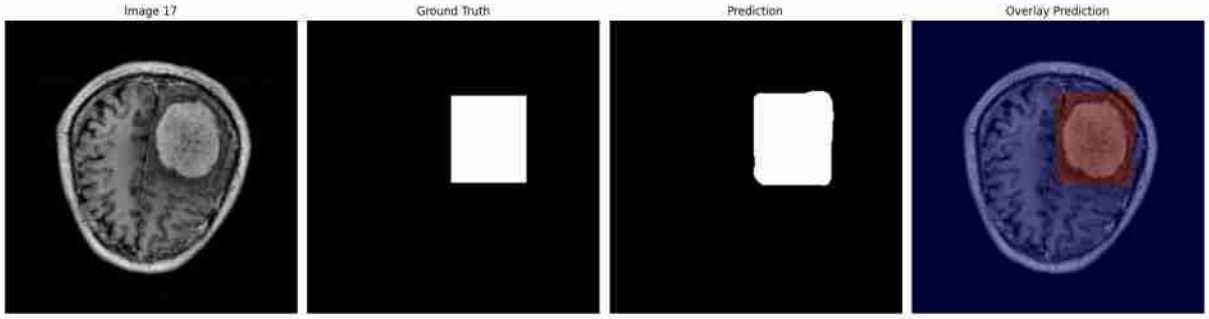


Figure 4.14: Qualitative Segmentation Results for TransUNet

4.6.4 Discussion and Comparison

In the current study, five deep learning models are evaluated for the segmentation of brain metastases in MRI images, with emphasis on their segmentation performance, convergence behavior, and robustness. The findings provide both quantitative and qualitative insights, offering a comprehensive understanding of the strengths and weaknesses of the different architectures.

According to the results presented in Table 4.7, DeepLabV3+ achieved the highest Dice coefficient and the lowest loss value, with scores of 0.7840 and 0.1742, respectively. Its superior performance is attributed to its ability to capture multi-scale contextual information through atrous convolutions and spatial pyramid pooling. These mechanisms enable more accurate segmentation while preserving fine structural details in medical images.

U-Net follows closely with a Dice coefficient of 0.7639 and a loss value of 0.1889. Its encoder-decoder architecture combined with skip connections contributes to effective spatial information preservation and improved localization accuracy. Additionally, the training and validation curves indicate stable convergence and good generalization, with no significant signs of overfitting.

Attention U-Net achieved a slightly lower Dice coefficient of 0.7495. Although attention gates are designed to focus on relevant regions, the results suggest that this added complexity does not lead to a significant performance improvement in this case. This behavior may be influenced by the size or characteristics of the dataset, which may not fully exploit the benefits of attention mechanisms.

TransUNet, which combines transformer-based components with convolutional layers, obtained a Dice coefficient of 0.7587 and a loss value of 0.1333. This hybrid architecture allows the model to capture both local features and global contextual information. However, its performance remains slightly below that of U-Net and DeepLabV3+, suggesting that the transformer component may not be fully utilized under the current training conditions.

In contrast, the ViT recorded the lowest Dice coefficient 0.6636 and the highest loss value. The training curves reveal slower convergence and a larger gap between training and validation metrics, indicating weaker generalization. This behavior aligns with the known limitation of transformer-based models, which typically require large-scale datasets to perform optimally.

The analysis of training dynamics further supports these observations. DeepLabV3+ exhibits the fastest and most stable convergence, followed by U-Net and TransUNet, which demonstrate balanced learning behavior. Attention U-Net shows relatively stable convergence, while ViT presents more erratic and less efficient learning patterns.

Qualitative results reinforce the quantitative findings. DeepLabV3+ produces smoother

and more consistent tumor boundaries, closely matching the ground truth masks. U-Net and TransUNet generate accurate segmentations, although minor boundary irregularities may appear. Attention U-Net provides acceptable localization but does not consistently outperform the baseline U-Net. In contrast, ViT struggles with precise boundary delineation and shows limited adaptability to variations in tumor shape.

Overall, the results indicate that convolution-based and hybrid architectures are more reliable for medical image segmentation tasks, particularly when working with limited datasets. Among the evaluated models, DeepLabV3+ offers the best balance between segmentation accuracy, convergence stability, and robustness.

Comparison with Existing Work

They further compared these results with published literature to test the efficacy of those proposed models.[Gopavarapu et al.(2024),[102]] conducted a recent study on brain tumor segmentation using the TumorSeg dataset, which revealed that Dice scores for U-Net and DeepLabV3+ were roughly 0.75 and 0.77, respectively.

By contrast, results from this work are competitive and may be marginally better in some cases. The Dice coefficient of DeepLabV3+ was 0.7840, which was higher than the literature's findings, while U-Net achieved a similar score of 0.7639.

The research also indicates that semantic segmentation techniques are more effective than instance-based methods for segmenting continuous structures like tumors. Our findings align with the fact that models created for pixel-wise segmentation (U-Net, DeepLabV3+, and TransUNet) are more efficient than the ViT model.

Ultimately, the findings indicate that advanced architectures that incorporate multi-scale feature extraction and contextual modeling significantly enhance segmentation performance.

Table 4.8: Quantitative comparison of segmentation models on the TumorSeg dataset

Authors	**Dataset**	**Model**	**Dice**
Gopavarapu et al., 2024	TumorSeg	U-Net	0.75
Gopavarapu et al., 2024	TumorSeg	DeepLabV3+	0.77
Our Work	TumorSeg	ViT	0.6636
Our Work	TumorSeg	U-Net	0.7639
Our Work	TumorSeg	Attention U-Net	0.7495
Our Work	TumorSeg	DeepLabV3+	**0.7840**
Our Work	TumorSeg	TransUNet	0.7587

The comparison suggests that the models evaluated exhibit competitive performance in contrast to current work. While U-Net and TransUNet are more effective, DeepLabV3+ is the most efficient because it captures multi-scale contextual information. On the other hand, transformer-only models like ViT are not as efficient in this situation because of

data limitations. These results show the importance of architectural design to medical image segmentation.

4.6.5 Challenges Encountered

The implementation and experimentation phases were characterized by several challenges that hindered the models' training and evaluation.

One of the primary issues was the limitation of computational resources. Google Colab and Kaggle are examples of cloud-based platforms that were tested, offering limited GPU availability and execution time. This constraint required careful management of the training sessions and optimization of model configurations.

Another issue was the variability in results, resulting from training in deep learning that is stochastic. Occasionally, the performance metrics for different runs of the same model were slightly different, particularly in terms of Dice Score. This was addressed by consistent evaluation methods, and the best models were kept.

ViT and TransUNet were among the complexity architectures that posed challenges. These models need large amounts of data to function properly, and their performance was limited by the relatively small dataset.

Additionally, models were found to be overfitted in some cases, resulting in good performance on training data but poor performance when validating or testing set specificities. This problem was addressed by means of early stopping and data augmentation.

The medical imaging data had to be handled with great care during preprocessing and preparation to maintain consistency between input images and corresponding segmentation masks, which is essential for training models.

4.7 Presentation of the Proposed Application

The initial interface of NeuroSeg serves as the entry point of the application, offering a modern and intuitive design with a dark theme and blue accents suitable for medical imaging. The application name and the subtitle "Brain Tumor Segmentation Explorer" are centrally displayed, clearly indicating its purpose, along with an AI related icon.

Key features Multi-Model, Real-Time, 4-Panel View, and Metrics are presented in a clear and organized manner to highlight the system's capabilities. A central "Get Started" button facilitates easy navigation to the main functionalities.

At the bottom, the interface references the deep learning models used (e.g., U-Net, DeepLabV3+, ViT), emphasizing the advanced techniques integrated into the system. Overall, the interface is designed to be clear, engaging, and user friendly.

Figure 4.15: Application Landing Page

The second interface of the **NeuroSeg** application represents the main working environment where users perform brain tumor segmentation. This screen is designed to provide a structured, interactive, and efficient workflow, allowing users to easily upload data, select models, and visualize results.

The interface is divided into two main sections:

Left Panel (Control Panel)

The left side of the interface contains all user controls, organized into three functional blocks:

- **Upload MRI Image:** This section allows users to import medical images in multiple formats (PNG, JPG, BMP, TIFF). A drag-and-drop area or file selection button (*Choose File*) is provided for ease of use. Once an image is selected, it becomes the input for the segmentation process.
- **Model Selection:** Users can choose the deep learning model used for segmentation from a dropdown menu. In the displayed example, the **U-Net** model is selected. A brief description of the model is provided, indicating its encoder-decoder architecture with skip connections and binary output configuration (sigmoid activation with thresholding). This helps users understand the behavior and characteristics of the selected model.
- **Run Segmentation:** This section includes a *Segment* button used to launch the segmentation process. The button remains inactive until an image is uploaded, ensuring proper workflow order and preventing invalid operations.

Right Panel (Visualization Area)

The right side of the interface is dedicated to displaying the segmentation results. Initially, this area is empty and shows a placeholder message indicating that results will appear after processing. Once the segmentation is executed, this panel dynamically updates to present the output, which may include predicted masks and visual comparisons.

Additional Interface Elements

- **Top Navigation Bar:** Displays the application name and confirms API connectivity status (e.g., *API online*), ensuring that the backend service is functioning correctly.
- **Footer:** Provides system status information and version details of the application.

Overall, this interface ensures a smooth and logical user experience by clearly separating input controls from output visualization. It simplifies the segmentation workflow into three main steps:

Upload Image → Select Model → Run Segmentation

This design makes the system accessible even for non-expert users.

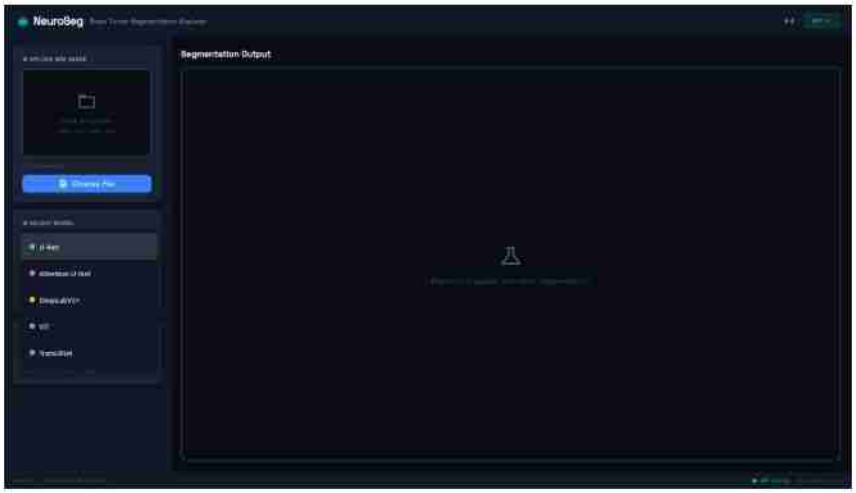


Figure 4.16: Segmentation Processing Interface

The third interface of the NeuroSeg application presents the segmentation results after processing, transforming the workspace into an interactive analysis environment. Unlike the previous screen, which focuses on input and execution, this interface emphasizes visual interpretation and quantitative evaluation of the model's output.

Once the segmentation is completed, the right panel is dynamically updated to display the MRI image with the predicted tumor region overlaid in red, allowing immediate

visual identification of the affected area. This overlay enhances interpretability by clearly distinguishing the tumor from surrounding brain tissues.

To facilitate deeper analysis, the interface provides multiple visualization modes accessible through dedicated buttons:

- **Overlay**: Displays the segmentation mask superimposed on the original MRI image.
- **Mask**: Shows the binary segmentation output highlighting only the detected tumor region.
- **Heatmap**: Represents the model's confidence distribution across the image.
- **Original**: Displays the input image without any processing.

In addition to visual outputs, the interface includes a set of quantitative metrics that provide insight into the segmentation results:

- **Tumor Area (%)**: Indicates the proportion of the image classified as tumor.
- **Confidence Score (%)**: Reflects the model's certainty in its prediction.
- **Tumor Pixels**: Represents the number of pixels identified as tumor.
- **Inference Time (ms)**: Measures the time required to perform the segmentation.

The left panel remains visible and unchanged, allowing users to easily modify inputs, select another model, or process a new image without leaving the results view. A confirmation message (Segmentation complete) provides feedback on the successful execution of the process.

Overall, this interface enhances the usability of the application by combining clear visual feedback with detailed numerical evaluation, enabling users to better assess the performance of the segmentation model and interpret the results efficiently.

Figure 4.17: Segmentation Results Interface

4.8 Conclusion

This chapter outlined the experimental testing of multiple deep learning models for brain tumor classification using MRI images. The study included ViT, U-Net, Attention U-Net, DeepLabV3+, and TransUNet, which all were subject to the same conditions.

Quantitatively, the best-performing program was DeepLabV3+ with a Dice Score of 0.7840, followed by U-Net and TransUNet. The Vit was the least successful of the models tested. The findings indicate that medical image segmentation tasks can be effectively executed using convolution-based architectures, particularly those designed to capture multi-scale contextual information.

It was found to have achieved competitive results while maintaining a relatively simple and efficient architecture, as shown by comparison with existing literature. This serves to demonstrate the significance of what methodology is used and the effectiveness of the models employed.

However, these promising results were met with various problems: computational difficulties; variation in the outcome of trainings; and sensitivity to sample dataset sizes, which could affect model performance. The challenges outlined above demand a delicate equilibrium between model complexity, data accessibility, and computational resources. Generally speaking, the findings from this chapter serve as a solid basis for further refinement. The future work may involve improving model architectures, expanding the dataset size, and exploring hybrid methods to further improve segmentation performance.

General Conclusion

In a medical context marked by the increasing complexity of neurological diseases and the growing need for reliable diagnostic tools, this project was part of the dynamic of artificial intelligence applied to medical imaging. The main objective was to design, implement, and evaluate DL architectures for the automatic segmentation of brain tumors from MRI images, in order to assist practitioners in precisely, reproducibly, and quickly delineating lesions.

To achieve this goal, we structured our work into four complementary chapters. The first chapter laid out the fundamentals of medical imaging, focusing on the specificities of MRI and the challenges related to noise and spatial resolution. The second chapter covered image segmentation methods, from classical approaches (thresholding, contours, regions) to modern supervised techniques. The third chapter focused on an in-depth study of DL, detailing the functioning of CNNs, attention mechanisms, and hybrid architectures integrating transformers. Finally, the fourth chapter detailed the experimental phase, including data preprocessing, implementation of five state-of-the-art architectures (ViT, U-Net, Attention U-Net, DeepLabV3+, TransUNet), and their rigorous evaluation on a clinically annotated dataset.

The experimental results obtained on a set of 2146 MRI images highlighted significant differences between the tested models. Based on the Dice coefficient and loss function, the results showed that DeepLabV3+ offered the best overall performance (Dice = 0.7840, Test Loss = 0.1742), thanks to its ASPP module capable of capturing multi-scale contextual information without losing spatial resolution. The classic U-Net model emerged as a solid and stable reference (Dice = 0.7639), confirming its historical relevance in medical segmentation. On the other hand, the ViT architecture presented the most modest results (Dice = 0.6636), mainly due to the modest size of the dataset and the lack of inherent spatial bias induction in pure Transformers. The analysis of learning curves confirmed rapid and stable convergence for convolutional and hybrid models, validating the effectiveness of the regularization strategies (Early Stopping, Data Augmentation) adopted.

Despite the encouraging results, this work presents some limitations. First, the relatively small size of the dataset and the imbalance between classes (tumor vs healthy tissue) may have influenced the generalization capacity of the models, particularly for small or diffuse tumor structures. Second, computational constraints related to the Google Colab environment limited the exploration of heavier configurations or extended training. Finally, the intrinsic nature of MRIs (heterogeneity of contrasts, motion artifacts, biologically blurred boundaries) remains a major challenge for any fully automated approach.

Following this work, several avenues for improvement and future perspectives emerge. It would be relevant to integrate multi-modal MRI data (T1, T1c, T2, FLAIR) to enrich the semantic context and improve the differentiation of tumor subregions (edema, necrosis, active core). Moving to 3D segmentation would also allow exploiting the volumetric continuity between slices. Architecturally, adopting hybrid CNN-Transformer models such as 'Swin-UNet' or SegFormer, or using self-supervised pretraining could significantly boost robustness against limited data. Finally, clinical validation on real data and the integration of complementary metrics (Hausdorff distance, Mean Surface Distance) will be essential steps for deployment in a hospital setting.

In summary, this thesis demonstrates that the integration of DL into MRI analysis is not only feasible but necessary to automate complex, time-consuming, and observer variable tasks. By combining methodological rigor, technical implementation, and critical analysis of results, this work constitutes a concrete contribution to the advancement of medical AI. It opens the way for smarter, interpretable, and accessible diagnostic assistance systems, thereby strengthening the link between computer innovation and clinical excellence.

Bibliography

[1] M. T. Parisi. "Importance of medical imaging—A perspective". In: *Imaging in Medicine* 15.6 (2023), pp. 125–126. URL: https://www.openaccessjournals.com/articles/importance-of-medical-imaginga-perspective-17079.html.

[2] Digital Image Basics. *Digital Image Basics. Bitmaps, Pixmaps, and theParisi2023 RGB Color Space*. PDF document: pixmaps-rgb.pdf. Unpublished course material. Chap. 1.

[3] Ouarda Assas. "Classification floue des images". Consulté en 2026. Thèse de doctorat en sciences. Batna, Algérie: Université de Batna, 2013. URL: https://dspace.univ-batna2.dz/server/api/core/bitstreams/8eae5c52-60b5-4079-9a84-91dcc94b802f/content.

[9] Abdulrahman Moffaq Alawad et al. "Fuzzy logic based edge detection method for image processing". In: *International Journal of Electrical and Computer Engineering (IJECE)* 8.5 (2018), pp. 3796–3804. DOI: 10.11591/ijece.v8i5.pp3796-3804.

[11] Nalin Kumar and M. Nachamai. "Noise removal and filtering techniques used in medical images". In: *Oriental Journal of Computer Science & Technology* 10.1 (2017), pp. 103–113. DOI: 10.13005/ojcst/10.01.14. URL: http://dx.doi.org/10.13005/ojcst/10.01.14.

[13] Jianhong Gan et al. "Multi-scale ultrasound image denoising algorithm based on deep learning model for super-resolution reconstruction". In: *Proceedings of the 2023 4th International Conference on Computer Recognition and Intelligent Systems (CCRIS '23)*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 6–11. DOI: 10.1145/3622896.3622898. URL: https://doi.org/10.1145/3622896.3622898.

[15] Pierrick Coupé et al. "Robust Rician Noise Estimation for MR Images". In: *Medical Image Analysis* 14.4 (2010). Preprint available at HAL-Inserm: https://inserm.hal.science/inserm-00486495v1/file/FinalVersionResub_last.pdf, pp. 483–493. DOI: 10.1016/j.media.2010.03.001.

[16] Ajay Kumar Boyat and Brijendra Kumar Joshi. "A Review Paper: Noise Models in Digital Image Processing". In: *arXiv preprint arXiv:1505.03489* (2015).

[18] Sunyoung Jung. "Motion Artifact-Informed Brain Tissue Segmentation Network via Disentanglement Learning". Master's Thesis. Seoul, South Korea: Yonsei University, 2025.

[20] Ziyang Wang and Irina Voiculescu. "Dealing with unreliable annotations: A noise-robust network for semantic segmentation through a transformer-improved encoder and convolution decoder". In: *Applied Sciences* 13.13 (2023), p. 7966. DOI: 10.3390/app13137966. URL: https://www.mdpi.com/2076-3417/13/13/7966.

[24] P. R. Patel and O. De Jesus. "CT scan". In: *StatPearls*. Treasure Island, FL: StatPearls Publishing, 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK567796/.

[25] G. Katti, S. A. Ara, and A. Shireen. "Magnetic resonance imaging (MRI)–A review". In: *International Journal of Dental Clinics* 3.1 (2011), pp. 65–70.

[26] Suraj Serai. "Basics of magnetic resonance imaging and quantitative parameters T1, T2, T2*, T1rho and diffusion-weighted imaging". In: *Pediatric Radiology* 52 (Apr. 2021), pp. 1–11. DOI: 10.1007/s00247-021-05042-7.

[30] Alain Faivre-Chauvet, Cécile Bourdeau, and Mickaël Bourgeois. "Radiopharmaceutical good practices: Regulation between hospital and industry". In: *Frontiers in Nuclear Medicine* 2 (2022). Accessed: 2026-01-26, p. 990330. DOI: 10.3389/fnume.2022.990330. URL: https://www.frontiersin.org/journals/nuclear-medicine/articles/10.3389/fnume.2022.990330/full.

[31] Stefan Bauer et al. "A Survey of MRI-based Medical Image Analysis for Brain Tumor Studies". In: *Medical Image Analysis* 17.8 (2013), pp. 1–26.

[32] Ahmeed Suliman Farhan et al. "XAI-MRI: an ensemble dual-modality approach for 3D brain tumor segmentation using magnetic resonance imaging". In: *Frontiers in Artificial Intelligence* 8 (2025), p. 1525240. DOI: 10.3389/frai.2025.1525240. URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1525240/full.

[39] PhD Whelan Sarah. "Benign vs Malignant Tumors". In: *Technology Networks* (2024). Accessed: 2026-02-02.

[44] Huan Wang et al. "Role of the nervous system in cancers: a review". In: *Cell Death Discovery* 7 (2021). Accessed: 2026-02-02, p. 76. DOI: 10.1038/s41420-021-00450-y. URL: https://www.nature.com/articles/s41420-021-00450-y.

[47] Yan Xu et al. "Advances in medical image segmentation: A comprehensive review of traditional, deep learning and hybrid approaches". In: *Bioengineering* 11.10 (2024). PMCID: PMC11505408, p. 1034. DOI: 10.3390/bioengineering11101034. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC11505408/.

[48] Namya Musthafa, Qurban A. Memon, and Mohammad M. Masud. "Advancing brain tumor analysis: Current trends, key challenges, and perspectives in deep learning–based brain MRI tumor diagnosis". In: *Eng* 6.5 (2025). Accessed: 31 January 2026, p. 82. DOI: 10.3390/eng6050082. URL: https://www.mdpi.com/2673-4117/6/5/82.

[49] Debasish Chakraborty, Goutam Sen, and Sugata Hazra. *Image segmentation Techniques*. Dec. 2012. ISBN: 978-3-8473-1137-9.

[51] Yan Xu et al. "Advances in Medical Image Segmentation: A Comprehensive Review of Traditional, Deep Learning and Hybrid Approaches". In: *Bioengineering* 11.10 (2024). ISSN: 2306-5354. DOI: 10.3390/bioengineering11101034. URL: https://www.mdpi.com/2306-5354/11/10/1034.

[52] Roshan Daniel, Mithisha Tavares, and Vinayak Sanjeev Deshpande. "UW-Madison GI Tract Image Segmentation to Track Healthy Organs to Improve Cancer Treatment". In: *Proceedings of the Conference (unpublished)*. Figure 1 used: "Example of Image Segmentation in medical imaging" (ResearchGate). Available at: https://www.researchgate.net/figure/Example-of-Image-Segmentation-in-medical-imaging_fig1_373042406. Aug. 2023.

[53] Sujata Bhairnallykar and Vaibhav Narawade. "Exploration of Image Segmentation: A Comprehensive Review of Traditional Segmentation Techniques". In: Dec. 2023, pp. 1–6. DOI: 10.1109/IEMENTech60402.2023.10423439.

[55] Sahar Zafari. "Segmentation of Overlapping Convex Objects". Figure: "An example of edge-based segmentation" retrieved from ResearchGate. Available at: https://www.researchgate.net/figure/An-example-of-edge-based-segmentation-18_fig2_283447261 (accessed: 2026-02-20). MA thesis. Lappeenranta, Finland: Lappeenranta-Lahti University of Technology (LUT), 2014.

[56] Università degli Studi di Verona. *IP-L8-Lecture: Segmentation*. Lecture Notes / Technical Report matdid125113. Accessed: 2026-02-20. Dipartimento di Informatica, Università degli Studi di Verona, unknown.

[57] Himanshu Mittal et al. "A Comprehensive Survey of Image Segmentation: Clustering Methods, Performance Parameters, and Benchmark Datasets". In: *Multimedia Tools and Applications* (2021). Figure 2: "Example of clustering-based image segmentation — original vs. segmented" (ResearchGate). Available at: https://www.researchgate.net/figure/Example-of-clustering-based-image-

segmentation-a-Original-image-b-Segmented-image_fig2_349192395 (accessed: 2026-02-20).

[58] Noor Ibrahim and Narjis Shati. "Review on graph theory-based image segmentation with its methods". In: *Journal of Al-Qadisiyah for Computer Science and Mathematics* 16 (June 2024). DOI: 10.29304/jqcsm.2024.16.21541.

[61] Ramez Yousri Adli, Gehad Hani, and Merna Said. "Artificial Intelligence: An Overview". In: Sept. 2022.

[62] Bernhard Mueller. *Machine Learning-Based Image Segmentation*. Bachelor's thesis. Supervised by Prof. Dr.-Ing. André Borrmann and Alexander Braun, M.Sc. Munich, Germany, 2018.

[66] Xia Zhao et al. "A Review of Convolutional Neural Networks in Computer Vision". In: *Artificial Intelligence Review* 57.4 (2024), p. 99. DOI: 10.1007/s10462-023-10620-9.

[68] Zewen Li et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2021), pp. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.

[69] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Figure: "Fully convolutional neural network architecture (FCN-8)" retrieved from ResearchGate. Available at: https://www.researchgate.net/figure/Fully-convolutional-neural-network-architecture-FCN-8_fig1_327521314 (accessed: 2026-02-21). 2015.

[70] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[72] Nahian Siddique et al. "U-Net and Its Variants for Medical Image Segmentation: Theory and Applications". In: *arXiv preprint arXiv:2011.01118* (2020). arXiv: 2011.01118 [cs.CV]. URL: https://arxiv.org/abs/2011.01118.

[73] Ao Wang. "Vision Transformers (ViTs): A New Era in Computer Vision - A Review". In: *Journal of Computing and Electronic Information Management* 18 (Oct. 2025), pp. 23–30. DOI: 10.54097/41t0vx90.

[75] Nikoleta Taoukidou, Dimitrios Karpouzos, and Pantazis Georgiou. "Flood Hazard Assessment Through AHP, Fuzzy AHP, and Frequency Ratio Methods: A Comparative Analysis". In: *Water* 17.14 (2025), p. 2155. DOI: 10.3390/w17142155.

[77] Mohammad Hossin and Md Nasir Sulaiman. "A Review on Evaluation Metrics for Data Classification Evaluations". In: *International Journal of Data Mining & Knowledge Management Process (IJDKP)* 5.2 (2015), pp. 1–11. DOI: 10.5121/ijdkp.2015.5201.

[78] Moazma Ijaz, Nimra Tariq, and Amina Malik. "Performance Evaluation of the U-Net Model for Medical Image Segmentation Using Dice Coefficient, IOU, and Loss Metrics". In: *History of Medicine* 10.2 (2024), pp. 1314–1324. DOI: 10.48047/HM.10.2.2024.1314-1324.

[79] Abdel Aziz Taha and Allan Hanbury. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool". In: *BMC Medical Imaging* 15.1 (2015), p. 29. DOI: 10.1186/s12880-015-0068-x. URL: https://doi.org/10.1186/s12880-015-0068-x.

[80] Md Alamin Talukder et al. "ACU-Net: Attention-based Convolutional U-Net Model for Segmenting Brain Tumors in fMRI Images". In: *DIGITAL HEALTH* 11 (2025), pp. 1–18. DOI: 10.1177/20552076251320288.

[81] Mohammad Abrar et al. "Enhancing Brain Tumor Segmentation Using Attention-Based Convolutional U-Net on MRI Images". In: *Scientific Reports* 15 (2025), p. 36603. DOI: 10.1038/s41598-025-20329-7.

[82] Amin Pourmahboubi, Nazanin Arsalani Saeed, and Hamed Tabrizchi. "A Brain Tumor Segmentation Enhancement in MRI Images Using U-Net and Transfer Learning". In: *BMC Medical Imaging* 25 (2025), p. 307. DOI: 10.1186/s12880-025-01837-4.

[83] Bonian Chen et al. "MRI Brain Tumour Segmentation Using Multiscale Attention U-Net". In: *Informatica* 35.4 (2024), pp. 751–774. DOI: 10.15388/24-INFOR574.

[84] Shuli Xing et al. "Semantic Segmentation of Brain Tumors Using a Local–Global Attention Model". In: *Applied Sciences* 15.5981 (2025). DOI: 10.3390/app15115981.

[85] H. Alquran et al. "An Improved U-Net Model for Brain Tumor Segmentation in MRI Images". In: *Applied Sciences* 14.6504 (2024). DOI: 10.3390/app14065004.

[102] Abhinav Gopavarapu. "Instance vs. Semantic Segmentation for Brain Tumor Delineation in MRI: A Comparative Study on the TumorSeg Dataset". In: (May 2025).

Webography

[4] Wikimedia Commons. *Digital picture*. https://commons.wikimedia.org/wiki/File:DigitalPicture.jpg. Image, Creative Commons license. 2007.

[5] Mohammed Hasan. *Multimedia: Lecture 3 – Sampling and Quantization*. Lecture notes, university course. 2024.

[6] RadiologyKey. *Image resolution: Pixel and voxel size*. Radiology Key. Jan. 2016. URL: https://radiologykey.com/image-resolution-pixel-and-voxel-size/.

[7] *Illustration of CT scan pixels, slice thickness and voxels*. Accessed: 2026-01-26. URL: https://www.researchgate.net/figure/Illustration-of-CT-scan-pixels-slice-thickness-and-voxels_fig1_381737431.

[8] Lenovo. *What is grayscale? Definition, uses, and applications in digital imaging*. Lenovo Glossary. URL: https://www.lenovo.com/us/en/glossary/grayscale/ (visited on 01/24/2026).

[10] David C. Preston. *Metastatic Disease Late: Metastatic Brain Tumor (Breast Cancer)*. Online neuroimaging teaching resource. Case Western Reserve University School of Medicine, Department of Neurology. Revised 11/30/06. Copyrighted 2006. Nov. 2006. URL: https://case.edu/med/neurology/NR/MetastaticDisLate.htm.

[12] Ephraim Ndiritu. *Noise in Image Processing and How to Add It to Images in Python*. Medium article, accessed 26 Jan 2026. 2024. URL: https://medium.com/@muyandiritujr/noise-in-image-processing-and-how-to-add-it-to-images-in-python-456a434dcc3f.

[14] Mdf. *Photon-noise.jpg*. Wikimedia Commons. A photon noise simulation image. June 2010. URL: https://commons.wikimedia.org/wiki/File:Photon-noise.jpg.

[17] Fernando Chamizo Lorente. *Periodic noise in images*. https://matematicas.uam.es/~fernando.chamizo/dark/d_periodic.html. Accessed: 2026-01-27. Departamento de Matemáticas, Universidad Autónoma de Madrid.

[19] Shireen Y. Elhabian. *Motion Artifacts in Diffusion Imaging*. https://www.sci.utah.edu/~shireen/diffusion.html. Accessed: 2026-01-27. Scientific Computing and Imaging Institute, University of Utah, 2022.

[21] ScienceDirect Topics. *Image blurring*. In Computer Science. URL: https://www.sciencedirect.com/topics/computer-science/image-blurring (visited on 01/24/2026).

[22] ECPI University. *What is radiography*. URL: https://www.ecpi.edu/blog/what-is-radiography#:~:text=Radiography%20is%20a%20crucial%20imaging,can%20be%20analyzed%20by%20professionals (visited on 01/24/2026).

[23] annotated by Mikael Häggström Blausen Medical. *Projectional Radiography Components*. Wikimedia Commons. Illustration showing components of projectional radiography including an X-ray generator and detector , Accessed: 2026-01-27. Apr. 2017. URL: https://commons.wikimedia.org/wiki/File:Projectional_radiography_components.jpg.

[27] Cancer Council Australia. *Ultrasound*. https://www.cancer.org.au/ultrasound. Accessed: 2026-01-25. 2026. URL: https://www.cancer.org.au/ultrasound.

[28] *Diagram of an Ultrasound Scanning Device [Vector Illustration]*. Shutterstock. Stock vector illustration showing the components of an ultrasound scanning device; asset id 2331981325 , Accessed: 2026-01-28. 2023. URL: https://www.shutterstock.com/fr/image-vector/diagram-ultrasound-scanning-device-2331981325.

[29] Ballad Health. *PET, SPECT & Nuclear Medicine*. https://www.balladhealth.org/medical-services/pet-spect-nuclear-medicine. Accessed: 2026-01-25. 2026. URL: https://www.balladhealth.org/medical-services/pet-spect-nuclear-medicine.

[33] Mayfield Brain & Spine. *Brain tumors: Overview of types, diagnosis, treatment options*. https://mayfieldclinic.com/pe-braintumor.htm. Accessed: 2026-01-29. Mayfield Clinic.

[34] J. Murray. *Brain Tumor Causes, Symptoms, and Treatments*. https://southmiamineurology.net/brain-tumor-causes-symptoms-and-treatments/. Accessed: 2026-02-02. South Miami Neurology, 2025.

[35] Cancer Research UK. *Primary and secondary brain tumours*. https://www.cancerresearchuk.org/about-cancer/brain-tumours/types/primary-secondary-tumours. Accessed: 2026-01-29. 2023.

[36] Anna Divoli et al. *A "Textbook" View of the Metastatic Progression of a Malignant Tumor [Figure 1]*. https://www.researchgate.net/figure/A-textbook-view-of-the-metastatic-progression-of-a-malignant-tumor-The-tumors_fig1_51716500. Accessed: 2026-02-02. ResearchGate, 2011.

[37] Dalvoy. *Primary vs. metastatic malignancies*. https://www.dalvoy.com/en/upsc/mains/previous-years/2013/medical-science-paper-i/primary-vs-metastatic-malignancies. Accessed: 2026-01-29.

[38] Ketan Patel et al. *Benign vs. malignant tumors*. 2020. URL: https://www.ncbi.nlm.nih.gov/books/NBK537077/ (visited on 01/29/2026).

[40] Mayo Clinic Staff. *Brain Tumor - Symptoms and Causes*. Consulté le 31 janvier 2026. 2025. URL: https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084.

[41] Mayo Clinic Staff. *Glioma — Symptoms and Causes*. https://www.mayoclinic.org/diseases-conditions/glioma/symptoms-causes/syc-20350251. Accessed: 2026-02-02. Mayo Clinic, 2024.

[42] National Cancer Institute. *Choroid Plexus Tumor: Diagnosis and Treatment*. https://www.cancer.gov/rare-brain-spine-tumor/tumors/choroid-plexus-tumors. Accessed: 2026-02-02. National Cancer Institute, 2024.

[43] Dr. Marcos Devanir. *Meningiomas*. https://drmarcosdevanir.com.br/meningiomas/. Accessed: 2026-02-02. Dr. Marcos Devanir, 2025.

[45] MD Aaron Cohen-Gadol. *Types of Pituitary Tumors*. https://www.aaroncohen-gadol.com/en/patients/pituitary-tumor/pituitary-tumors/types. Accessed: 2026-02-02. Aaron Cohen-Gadol, MD, 2024.

[46] Encord. *Medical image segmentation definition*. https://encord.com/glossary/medical-image-segmentation-definition/. Accessed: 2026-01-31.

[50] *Image Segmentation: The Deep Learning Approach*. https://indiaai.gov.in/article/image-segmentation-the-deep-learning-approach. Accessed: 2026-02-20. INDIAai - National AI Portal of India.

[54] Nikolaj Buhl. *Image Thresholding in Image Processing*. https://encord.com/blog/image-thresholding-image-processing/. Accessed: 2026-02-20. Encord, Sept. 2023.

[59] Vighnesh Birodkar. *Graph based Image Segmentation*. https://vcansimplify.wordpress.com/2014/05/16/graph-based-image-segmentation/. Accessed: 2026-02-20. May 2014.

[60] ChatSlide. *Présentation simplifiée de l'IA et ses usages éducatifs*. https://www.chatslide.ai/shared/presentation-de-lia-avec-ses-avantages-et-s-gsakbp. Online presentation created with ChatSlide.ai. 2026.

[63] TheDataScientistKiran. *Regression vs Classification in Machine Learning: What's the Difference?* Published on Python in Plain English. 2025. URL: https://python.plainenglish.io/regression-vs-classification-in-machine-learning-whats-the-difference-19c57837c203.

[64] MathWorks. *What Is Unsupervised Learning?* https://www.mathworks.com/discovery/unsupervised-learning.html. MathWorks overview of unsupervised learning in machine learning. 2026.

[65] AltexSoft Editorial Team. *Reinforcement Learning Explained: Overview, Comparisons and Applications in Business*. https://www.altexsoft.com/blog/reinforcement-learning-explained-overview-comparisons-and-applications-in-business/. Online article providing an overview of reinforcement learning and its business applications. 2019.

[67] MathWorks, Inc. *Convolutional Neural Network (CNN) - What it is and how it works*. https://www.mathworks.com/discovery/convolutional-neural-network.html. Accessed: 2026-02-21. 2026.

[71] Abhishek. *U-Net Architecture Explained: A Simple Guide with PyTorch Code*. https://medium.com/@AIchemizt/u-net-architecture-explained-a-simple-guide-with-pytorch-code-fc33619f2b75. Medium blog post. Accessed: 2026-02-21. Aug. 2025.

[76] Rachel Draelos. *Measuring Performance: The Confusion Matrix*. https://glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/. Accessed: 2026-02-20. 2019.

[86] CMLABS. *What Is Google Colab? Definition, Benefits, and Examples*. Accessed: 2026-03-29. 2024. URL: https://cmlabs.co/en-en/seo-terms/what-is-google-colab.

[87] StickPNG. *Google Colab Logo PNG (Transparent Background)*. Accessed: 2026-04-20. Free PNG image, personal use only. n.d. URL: https://www.stickpng.com/img/icons-logos-emojis/tech-companies/google-colab-logo.

[88] DataCamp. *What is Kaggle?* Accessed: 2026-03-29. 2022. URL: https://www.datacamp.com/blog/what-is-kaggle.

[89] Kaggle. *Kaggle Logo*. Accessed: 2026-03-31. 2024. URL: https://commons.wikimedia.org/wiki/File:Kaggle_Logo.svg.

[90] Amazon Web Services. *What is Python?* Accessed: 2026-03-29. 2026. URL: https://aws.amazon.com/what-is/python/.

[91] Wikimedia Commons contributors. *Python logo and wordmark.* Accessed: 2026-03-31. 2026. URL: https://commons.wikimedia.org/wiki/File:Python_logo_and_wordmark.svg.

[92] Databricks. *What is TensorFlow?* Accessed: 2026-03-30. 2025. URL: https://www.databricks.com/fr/blog/what-is-tensorflow.

[93] Wikipedia contributors. *TensorFlow.* Accessed: 2026-03-31. 2026. URL: https://en.wikipedia.org/wiki/TensorFlow.

[94] Nefe Emadamerho-Atori. *The Good and Bad of Keras Deep Learning API.* Accessed: 2026-03-30. 2025. URL: https://www.altexsoft.com/blog/keras-pros-and-cons/.

[95] Keras Team. *Keras: Deep Learning for Humans.* Accessed: 2026-03-31. 2026. URL: https://keras.io/.

[96] Matplotlib Development Team. *Matplotlib: History.* Accessed: 2026-03-30. 2026. URL: https://matplotlib.org/stable/project/history.html.

[97] Matplotlib Development Team. *Matplotlib: Visualization with Python.* Accessed: 2026-03-31. 2026. URL: https://matplotlib.org/.

[98] NumPy Developers. *What is NumPy?* Accessed: 2026-03-30. 2026. URL: https://numpy.org/doc/stable/user/whatisnumpy.html.

[99] NumPy Developers. *NumPy Source Code Repository.* Accessed: 2026-03-31. 2026. URL: https://github.com/numpy/numpy.

[100] Ultralytics. *COCO Dataset for Object Detection.* Accessed: 2026-03-30. 2026. URL: https://docs.ultralytics.com/datasets/detect/coco/.

[101] Peyman Darabi. *Brain Tumor Image Dataset: Semantic Segmentation.* Accessed: 2026-03-31. 2023. URL: https://www.kaggle.com/datasets/pkdarabi/brain-tumor-image-dataset-semantic-segmentation.