الجمهورية الديمقراطية الشعبية الجزائرية

People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

DR. YAHIA FARES UNIVERSITY OF MEDEA



Faculty of Sciences

Department of Mathematics and Computer Science

Graduation Project Report Submitted for the degree of Master of Computer science

Specialty: Engineering Systems and Web Technology

Brain Edema and Tumor Detection Using novel autoencoder u-net architecture

Proposed and directed by:

- ***Dr.HENTABLI Hamza***

Prepared and Presented by:

- ***Ms.BOUDIAF Katr-ennada***
- ***Mr.HADDAD Islem***

University Year: 2020/2021

Acknowledgment

We are infinitely deeply thankful to our God for giving us power and faith to edit this work.

Firstly, we would like to express our sincere gratitude to our advisor and mentor Dr. HENTABLI HAMZA for the continuous support that he gives us, for his patience, motivation, and immense knowledge. his guidance helped us during this project and writing of this report.

We could not have imagined having better advisor and mentor for our end of study project.

We would like to express our sincere thanks to all the professors who we have taught and who gives their skills to supported us in the continuation of our studies

We deeply thank the members of jury for accepting to judge our work.

Finally, we must express our profound gratitude to our parents for providing us with unfailing support and continuous encouragement throughout our years of study and through the process of researching and writing this project.

This accomplishment would not have been possible without them.

Thank you.

Dedications

Thank you, ALLAH, (my god) for giving me the ability to write and to think, the strength to believe, the patience to go to end of the dream and the happiness of raise my hands to the sky and say it

"YaKayoum".

I dedicate this modest work to the one who gave me life, the symbol of tenderness, which sacrificed itself for my happiness and my success, to my mother

OUAHIBA SILLALI

My dear father SLIMEN for his encouragement, Patience, his continued help along my path of study and financial support.

To My grandmother may God grant her a long life.

Without forgetting My sisters NARDJESS and AMANI (9et9outa) and my brothers HOUSSEM, ANIS, ASSIL AMIR

To my friends: AHLEM and ABIR, ASMAA, AMIRA and all the students of the promotion of 2nd year master ISTW 2020-2021

To my partner and brother ISLEM, for his friendship, loyalty and frankness, and all his family.

KATR-ENNADA

Dedications

Throughout my life one person has always been there, during those difficult and rough times. I would like to dedicate this thesis, and everything I do to my mother,

ZAHRA GUEMGHAR.

In addition to her, I have always been surrounded by strong supportive men. I would not be who I am today without the love and support of my precious father MUHAMMED AMINE HADDAD

To all my family and people around us, who inspired and gave me the perseverance, and determination in order to accomplish this work.

To My friends, To my partner during this project KATR-ENNADA

I Hope this work gets the summit of your wishes about me.

ISLEM

Abstract

Brain tumor and edema localization and segmentation from magnetic resonance imaging (MRI) are hard and important tasks for several applications in the field of medical analysis. As each brain imaging modality gives unique and key details related to each part of the tumor and edema. A tumor could be found in any area of the brain and could be of any size, shape, and contrast. There may exist multiple tumors of different types in a human brain at the same time. Accurate tumor area segmentation is considered primary step for treatment of brain tumors. Deep Learning is a set of promising techniques that could provide better results as compared to nondeep learning techniques for segmenting timorous part inside a brain.

In this work, we propose a fully automatic brain tumor and edema segmentation technique in brain magnetic resonance (MR) images. we focus on the recent trend of deep learning methods in this field. First, an introduction to brain tumors and methods for brain tumor and edema segmentation is given. Then, the proposed network uses BRATS segmentation challenge dataset which is composed of images obtained through four different modalities. Accordingly, we present an extended version of existing network to solve segmentation problem. Finally, an assessment of the current state is presented and future developments to standardize MRI-based brain tumor segmentation methods into daily clinical routine are addressed.

Key words: Brain, edema, Image segmentation, Benchmark, MRI, Multi-modality, Convolutional Neural Network, U-Net, Deep learning.

Résumé

La localisation et la segmentation des tumeurs cérébrales et des œdèmes à partir de l'imagerie par résonance magnétique (IRM) sont des tâches difficiles et importantes pour plusieurs applications dans le domaine de l'analyse médicale. Comme chaque modalité d'imagerie cérébrale donne des détails uniques et clés liés à chaque partie de la tumeur et de l'œdème. Une tumeur peut être trouvée dans n'importe quelle zone du cerveau et peut être de n'importe quelle taille, forme et contraste. Il peut exister plusieurs tumeurs de différents types dans un cerveau humain en même temps. La segmentation précise de la zone tumorale est considérée comme l'étape principale du traitement des tumeurs cérébrales. L'apprentissage profond est un ensemble de techniques prometteuses qui pourraient fournir de meilleurs résultats par rapport aux techniques d'apprentissage non approfondi pour segmenter une partie timorée à l'intérieur d'un cerveau.

Dans ce travail, nous proposons une technique de segmentation entièrement automatique des tumeurs cérébrales et des œdèmes dans des images de résonance magnétique (IRM) cérébrales. nous nous concentrons sur la tendance récente des méthodes d'apprentissage en profondeur dans ce domaine. Tout d'abord, une introduction aux tumeurs cérébrales et aux méthodes de segmentation des tumeurs cérébrales et de l'œdème est donnée. Ensuite, le réseau proposé utilise l'ensemble de données de défi de segmentation BRATS qui est composé d'images obtenues à travers quatre modalités différentes. En conséquence, nous présentons une version étendue du réseau existant pour résoudre le problème de segmentation. Enfin, une évaluation de l'état actuel est présentée et les développements futurs pour standardiser les méthodes de segmentation des tumeurs cérébrales basées sur l'IRM dans la routine clinique quotidienne sont abordés.

Mots clés : Cerveau, œdème, Segmentation d'images, Benchmark, IRM, Multimodalités, Réseau de neurones convolutifs, U-Net, Deep Learning.

الملخص

يعتبر توطين ورم الدماغ والوذمة والتجزئة من التصوير بالرنين المغناطيسي (MRI) من المهام الصعبة والمهمة للعديد من التطبيقات في مجال التحليل الطبي. حيث أن كل طريقة من طرق تصوير الدماغ تعطي تفاصيل فريدة ومهمة تتعلق بكل جزء من الورم والوذمة. يمكن العثور على الورم في أي منطقة من الدماغ ويمكن أن يكون بأي حجم وشكل وتباين. قد توجد أورام متعددة من أنواع مختلفة في دماغ الإنسان في نفس الوقت. يعتبر التقسيم الدقيق لمنطقة الورم خطوة أساسية لعلاج أورام الدماغ. التعلم العميق عبارة عن مجموعة من التقنيات الواعدة التي يمكن أن توفر نتائج أفضل مقارنة بتقنيات التعلم غير العميق لتقسيم الجزء المؤقت داخل الدماغ.

في هذا العمل، نقترح تقنية تجزئة أوتوماتيكية بالكامل لورم الدماغ والوذمة في صور الرنين المغناطيسي للدماغ (MRI).نحن نركز على الاتجاه الحديث لأساليب التعلم العميق في هذا المجال. أولاً، يتم تقديم مقدمة عن أورام المخ وطرق ورم الدماغ وتقسيم الوذمة. بعد ذلك، تستخدم الشبكة المقترحة مجموعة بيانات تحدي تجزئة BRATS التي تتكون من صور تم الحصول عليها من خلال أربع طرق مختلفة. وفقًا لذلك، نقدم نسخة موسعة من الشبكة الحالية لحل مشكلة التجزئة. أخيرًا، يتم تقديم تقييم للحالة الحالية ويتم تناول التطورات المستقبلية لتوحيد أساليب تجزئة ورم الدماغ القائمة على التصوير بالرنين المغناطيسي في الروتين السريري اليومي.

الكلمات الأساسية: الدماغ، الوذمة، تجزئة الصورة، المعيار المعياري، التصوير بالرنين المغناطيسي، الوسائط المتعددة، شبكة عصبية متشابكة ، U-Net، التعلم العميق.

Table of content:

List of Figures I

List of Table.....III

List of Acronyms IV

General introduction:.....1

1.1 Problem: 2

1.2 Research goal 3

1.3 Structure of the thesis : 3

Chapter 01: state of the art

1 Introduction: 5

2 Brain tumor : 5

3 Peritumoral brain edema (PTBE): 6

4 Magnetic Resonance Imaging (MRI): 7

4.1 Modalities of MR imaging: 10

4.1.1 T1-weighted MRI (T1): 10

4.1.2 T2-weighted MRI (T2): 10

4.1.3 T1-weighted MRI with gadolinium contrast enhancement (T1-C): 10

4.1.4 Fluid Attenuated Inversion Recovery (FLAIR): 11

5 Classification of brain tumor: 11

5.1 High-Grade Glioma (HGG): 11

5.2 Low-Grade Glioma (LGG): 11

6 Methods for Brain Tumor Image Segmentation: 12

6.1 Manual Segmentation Methods: 12

6.2 Semi-Automatic Segmentation Methods: 13

6.3 Fully Automatic Segmentation Methods: 13

7 CNN for medical imaging 14

8 Related work: 14

9 Conclusion : 23

Chapter 02: Deep learning for image segmentation

1 introduction.....25

2 Definition: 25

3 Deep learning history: 26

4 How does deep learning attain such impressive results? 27

5 Principe: 28
6 Classification of deep Learning: 28
6.1 Supervised: 28
6.2 Semi-supervised 29
6.3 Unsupervised 29
6.4 Deep Reinforcement Learning (DRL): 29
7 Why deep learning: 31
7.1 Universal learning approach: 31
7.2 Robust: 31
7.3 Generalization: 31
7.4 Scalability 31
8 When and where to apply DL: 32
9 Popular deep learning algorithms: 34
9.1 Convolutional neural networks CNN: 34
9.2 How Convolution Neural Networks Work: 36
9.2.1 Convolutional layer 36
9.2.2 Pooling layer: 37
9.2.3 Fully Connected Layer: 38
9.3 Activation Function: 39
9.3.1 ReLU function: 39
9.3.2 Max Pooling function: 40
9.3.3 SoftMax function: 40
9.4 Other layers : 41
9.4.1 Dropout layer 41
9.4.2 Up sample layer: 41
9.5 U-Net architecture: 42
9.6 Autoencoder: 42
9.6.1 The encoding phase: 43
9.6.2 The decoding phase: 43
10 Conclusion: 43

Chapter 03 : The proposed methodology

1. Introduction: 45
2. Image Processing Benchmarks: 45
3. Materials and Methods: 46
3.1 Brats data set: 46
3.2 Description of the Data set: 47

3.3 Loading and Reading the Data: 55
3.4 Data Pre-processing: 56
4. The proposed approach: 57
4.1 Network Architecture: 58
4.1.1 The proposed CNN network model: 58
4.1.2 The Merge models: 63
4.2 Training: 68
4.3 Prediction: 68
5. Understanding Boxplots 69
5.1 Median 70
5.2 Inter-quartile range 70
5.3 Upper quartile 70
5.4 Lower quartile 70
5.5 Whisker 70
6. Measures of Evaluating the performance of the models: 70
6.1 Dice: 70
6.2 Jaccard coefficient 71
7. Conclusion: 72

Chapter 04: Implimentation and evaluation

1 Introduction: 74
2 Presentation of tools: 74
2.1 Hardware: 74
2.2 Development environment: 75
2.2.1 Python: 75
2.2.2 Python's features: 75
2.2.2.1 Easy-to-learn: 76
2.2.2.2 Free and Open Sources: 76
2.2.2.3 High-Level Programming Language: 76
2.2.2.4 Expressive: 76
2.2.2.5 Portable: 77
2.2.2.6 Object-Oriented: 77
2.2.2.7 Interactive: 77
2.2.2.8 Interpreted: 77
2.2.2.9 Extendable: 77
2.2.2.10 Large Standard Library: 77
2.2.2.11 Embeddable: 78

2.2.2.12 Supports: 78
2.2.2.13 GUI Programming: 78
2.2.2.14 Dynamically Typed: 78
2.2.3 Python's used modules (Environment): 78
2.2.3.1 TensorFlow: 78
2.2.3.2 Keras: 79
2.2.3.3 Backend: 80
2.2.3.4 Glob: 80
2.2.3.5 OS: 80
2.2.3.6 Sys: 80
2.2.3.7 NumPy: 81
2.2.3.8 Pickle: 81
2.2.3.9 Matplotlib: 81
2.2.3.10 CV2: 81
2.2.3.11 Json: 82
2.2.4 Version of modules used: 82
3 Results and Discussions: 83
3.1 Result of no merge models: 83
3.1.1 Comparison of results 83
3.2 Result of no merge models: 88
3.2.1 Comparison of results 88
4 Global evaluation: 96
5 discussion 97
6 conclusion: 98
General conclusion and perspective. 100
Reference: 101

List of Figures:

Figure 1. 1:MR imaging for brain tumor 6
Figure 1. 2:MR imaging for brain Tumor associated with edema (E). 7
Figure 1. 3: 3D image MRI slices 8
Figure 1. 4:images of MRI machine equipment 9
Figure 1. 5: Brain multi-modality MRI images showing a) T1, b) T2, c) T1C and d) FLAIR. 11
Figure 1. 6 :MR images, of the two categories (HGG, LGG) 12
Figure 2. 1 : Neural networks architecture in DL 28
Figure 2. 2: The different classification of deep learning 30
Figure 2. 3: Example images where DL is applied successfully and achieved state of-the-art performance. 33
Figure 2. 4: Generic representation of the CNN architecture. 35
Figure 2. 5: Formation and arrangement of convolution layer. 37
Figure 2. 6: Pooling layer 38
Figure 2. 7: The structure of the fully connected layer. 39
Figure 2. 8 : Relu activation function curve 39
Figure 2. 9: Max pooling 40
Figure 2. 10: Convolutional-neural-network-CNN-architecture-based-on-U-NET 42
Figure 2. 11: Auto encoder architecture 43
Figure 3. 1: Tumor appearance on two imaging modalities (B = T1c, C = T2) with manual annotations, edema region appears in D (D = flair), healthy tissues in A (A= T1) and finally (E=Truth) 47
Figure 3. 2: 3D Ground training views of 176 MRI slices of the flair modality of patient HG0004 obtained from the BRATS-1 dataset. 48
Figure 3. 3:3D Ground training views of 176 MRI slices of the T1 modality of patient HG0004 obtained from the BRATS-1 dataset. 49
Figure 3. 4:3D Ground training views of 176 MRI slices of the T1C modality of patient HG0004 obtained from the BRATS-1 dataset. 50
Figure 3. 5: 3D Ground training views of 176 MRI slices of the T2 modality of patient HG0004 obtained from the BRATS-1 dataset. 51
Figure 3. 6:3D ground truths views of 176 MRI slice of the patient's brats HG0004 obtained from BRATS-1 dataset 52
Figure 3. 7 diagram that clarify how to read and load the data. 55
Figure 3. 8:the different steps of data pre-processing 57
Figure 3. 9: encoder-decoder design of the U-NET approach. 58
Figure 3. 10: The architecture of the proposed CNN network for models without merge. 59
Figure 3. 11 :plot model of no merge U-NET architecture 62
Figure 3. 12 : The architecture of the proposed CNN network for merge models. 64
Figure 3. 13: plot model of merge U-NET architecture 67
Figure 3. 14 : The sample training patches of brain tumor and edema from MRI images (a) and the corresponding ground truths (b). 68
Figure 3. 15: Different parts of a boxplot 69
Figure 3. 16:T1 is the ground truth lesion, and T0 is the area outside T1 within the brain. P1 is the algorithm's predicted lesion, and P0 is the algorithm's predicted area outside P1 within the brain. Overlapped area between T1 and P1 gives us the true positive. [49] 72

Figure 4. 1: Google Collab logo. 74
Figure 4. 2:python logo. 75
Figure 4. 3: TensorFlow logo. 79
Figure 4. 4: keras logo. 79
Figure 4. 5:Figure represent the Boxplot tumor and edema prediction results for no merging models of dice coefficient. 84
Figure 4. 6: Figure represent the Boxplot tumor and edema prediction results for no merging models of Jaccard coefficient 84
Figure 4. 7: Examples of the prediction result of model T1 87
Figure 4. 8: Examples of the prediction result of model T2 87
Figure 4. 9: Examples of the prediction result of model T1C 87
Figure 4. 10: Examples of the prediction result of model FLAIR 87
Figure 4. 11: Figure represent the Boxplot tumor and edema prediction results for merging models of Jaccard coefficient. 89
Figure 4. 12: Figure represent the Boxplot tumor and edema prediction results for merging models of dice coefficient. 89
Figure 4. 13: Examples of the prediction result of model T1/FLAIR. 94
Figure 4. 14: Examples of the prediction result of model T1/T1C 94
Figure 4. 15: Examples of the prediction result of model T1/T2. 94
Figure 4. 16: Examples of the prediction result of model T1C/FLAIR. 95
Figure 4. 17: Examples of the prediction result of model T2/FLAIR. 95
Figure 4. 18: Examples of the prediction result of model T1C/T2 95
Figure 4. 19: Figure represent the Boxplot tumor and edema prediction results for merging and no merging models of Jaccard and dice coefficient. 96
Figure 4. 20: Examples of the prediction result of low-grade for merge models. 97

List of tables:

Table 1. 1: Overview of recent methods on the basis of different features. 22
Table 3. 1: Properties of BRATS-1 dataset for high-grade. 53
Table 3. 2: Properties of BRATS-1 dataset for Low-grade. 54
Table 3. 3: No merge Model: "U-NET" summary. 60
Table 3. 4: Merge model "U-NET" summary. 65
Table 4. 1: versions of the python modules used 82
Table 4. 2: Table represent the tumor prediction results for no merging models(T1,T2,T1C,FLAIR) 85
Table 4. 3: Table represent the edema prediction results for no merging models(T1,T2,T1C,FLAIR)86
Table 4. 4: Table represent the tumor prediction results for merging models (T1/FLAIR, T1/T1C, T1/T2)..... 90
Table 4. 5: Table represent the edema prediction results for merging models (T1/FLAIR, T1/T1C, T1/T2)..... 91
Table 4. 6: Table represent the tumor prediction results for merging models (T1C/FLAIR, T2/FLAIR, T1C/T2)..... 92
Table 4. 7: Table represent the edema prediction results for merging models (T1/FLAIR, T2/FLAIR, T1C/T2)..... 93

Acronyms:

MRI: magnetic resonance imaging

T1: T1-weighted

T2: T2-weighted

T1C: T1-weighted contrast

FLAIR: Fluid Attenuated Inversion Recovery

U-NET: Unified Narcotics Enforcement Team

HG: High-Grade

LG: low-Grade

PTBE: Peritumoral brain edema

BRATS: brain tumor segmentation

DL: Deep learning

DRL: Deep Reinforcement Learning

CNN: convolutional neural network

RELU: Rectified Linear Units

FCL: fully connected layer

MPL: Multi-layer perceptron

GPU: graphics processing unit

GLOB: global

OS: operating system

SYS: System-specific parameters and functions

NUMPY: Numeric Python" or "Numerical Python"

JSON: JavaScript Object Notation

Cv2: OpenCV-Python

General introduction

General introduction

Cancer is defined as uncontrolled, abnormal cell growth and division in the body. Occurrence, as a mass, of these unnatural cell growth and division in the brain tissue is called a brain tumor. While brain tumors are uncommon, they are among the deadliest cancers. [1]

Based on brain tumor estimates, "it is estimated that 58418 New cases (Males:27 328 Female:31 090) in Algeria will be diagnosed with primary cancerous tumors of the brain and spinal cord". "Moreover, 4 116 adults are died and 38 886 expected to die in 2020 due to primary cancerous brain and CNS tumors''. [2]

Brain tumor detection and segmentation deals with separating the tumor portions from the normal tissues like Grey matter, White Matter and Cerebrospinal Fluid [3]. It can be done manually or automatically with the aid of a software. However human decisions are not always accurate and are time consuming which leads to computer analysis of tumor samples. [1][3]

Brain tumor can pose a serious risk to a large number of patients, with a one-year survival rate for high-grade gliomas. These tumors are often accompanied by peritumoral edema, which corresponds to extensive perifocal swelling. The major method for assessing diseased areas is magnetic resonance imaging (MRI). As a result, precise MRI segmentation of tumor areas and edema is a critical preparatory effort for a comprehensive and repeatable diagnosis of brain tumors. Manual segmentation might take an inordinate amount of time and would never be repeatable in clinical practice. [4]

In order to overcome these drawbacks, automatic or semi_automatic segmentation techniques are highly suggested. Several previous research investigations looked into this issue by segmenting both the tumor and the edema areas in one or more modalities (T1, T1C, T2, and Flair), but it's still a difficult process. [4]

Most of the existing algorithms are still not fast and not flexible enough, especially for realistic clinical scenarios and needs. One could notice some difficulties inherent to the brain tumor segmentation because, in general, brain tumors have shape and intensity characteristics that may vary dramatically from one subject to another, which impedes building reliable models from training data. For instance, the brain tumor region shape may be arbitrary, and does not necessarily fall within a category (or class) of shapes that could be learned from one finite set of training subjects as is the case of most medical image segmentation problems. [4]

Furthermore, tumors may have intensity patterns that are very similar to those of the other normal areas in the picture under consideration.

Deep learning methods enable efficient processing and objective evaluation of large amounts of MRI-based image data. There are a number of existing review articles focusing on traditional methods of segmentation of brain tumor images by MRI. Different from others, in this article we focus on the recent trend of deep learning methods in this field.

Deep learning is widely used in the field of medical imaging in a variety of ways, such as denoising, finding biomarkers, predicting patterns, and detecting lesions and tumors. Applying deep learning techniques to multimodal MRI images for tumor segmentation is naturally a difficult task due to the high dimensionality of the input data, poor quality and image issues during capture. such as the bias field, and after all, designing an architecture appropriate for the specific purpose.[5]

Most recent tumor segmentations have focused on deep learning methods. U-Net proposed to use jump layers to avoid loss of spatial information and was suggested for fine segmentation of medical images. Several methods have proposed to use U-Net for the segmentation of brain tumors. U-Net has shown promising results in medical image segmentation tasks. So, several researches have focused on modifying U-Net to achieve even better results.

1.1 Problem:

The segmentation of brain edema and tumors is an important task in the processing of medical images. Early diagnosis of brain tumors plays an important role in improving treatment options and increasing patient survival rates. Manual segmentation of brain edema and tumors for cancer diagnosis, from a large quantity of MRI images generated in routine clinical practice, is a difficult and time-consuming task. There is a need for automatic segmentation of brain edema and tumor images. [6]

Tumor and edema have been poorly detected in low-grade classification in recent years, in addition to the two major obstacles that one can face: first, a healthy patient is misdiagnosed as sick by the system or doctors; this error does not endanger the patient's life because after the tests, he will realize that he is in good health. On the other hand, the second issue arises when we are unable to discover a cancerous growth in a patient.

In this work the interesting and most changing problem is how can we train the algorithm to detect tumor and edema in HGG and specially in LGG classification for better outcomes? And How to use each modality of MRI to give better detection of brain edema and tumor image?

1.2 Research goal

The main contributions of this study are to propose a new approach to detect the brain tumor and edema in T1, T2, T1C, FLAIR with both of High-Grade (HG) and low-Grade (LG) by using technique of deep learning. And finally, to propose a new segmentation technique using the deep learning also in the new fused modalities of MR images to localize better the brain tumor and edema.

In order to achieve our main objective, other specific sub-objectives have been specified:

- To present a state of the art of existing detection methods;
- Propose new approach for the detection of brain edema and tumor image based on a justified choice of efficient detection methods and image segmentation;
- Implement, present and discuss the results of the proposed approach.

1.3 Structure of the thesis :

This thesis is organized into four chapters: In the first chapter, we will present an overview of brain tumors, brain edema, MRI modalities and brain tumor segmentation methods. Then, in the second chapter, we will see the different methods and techniques used to implement our research; Then, in the third chapter, we will present the architecture models to detect and segment brain edema and tumor image that we have proposed, as well as the methods and techniques used and the description of the dataset we used. In the fourth chapter, the implementation of the methods, details of the hardware, software, configuration parameters. Also, we will present in detail the results obtained with discussions. Finally, we will conclude our thesis, with a summary of the results and perspective of the work carried out during this project.

Chapter 01

State of the art

1 Introduction:

In this chapter, firstly we define what means brain tumors and its dangerous effects on the humans' lives, as well as we discuss the Peritumoral brain edema. Then we present the most popular imaging techniques specifically the Magnetic Resonance Imaging, with the different modalities of MRI, moreover, the brain tumor's methods of segmentation with the two categories (LG, HG). finally, some research work.

2 Brain tumor :

The brain is the most important and crucial structural element of the human body, consisting approximately 50–100 trillion neurons. It is often referred to as the center region of the human body. Furthermore, it is known as the nervous system's processor or kernel since it plays the most important and critical role in the nervous system.[7]

In general, the brain is the most sensitive and the most complex organ in the human body which plays a remarkable role in our daily lives [8]. Unlike other disorders, the aberrant or bulk development of cells in the brain causes a wide range of alterations in brain activity and features. It also creates resistance to performing brain functions. This anomaly or dysfunction is a symptom of a brain tumor, which is a very severe disease.

Brain tumor and other cancers killed more people than any other neurological illness. It is the tenth largest cause of death in both men and women. In other words, a brain tumor is caused by uncontrolled cell development within the brain. ‘‘It causes cancer, which may be the cause of death, and accounts for around 13% of all fatalities globally." The danger level of a brain tumor is determined by criteria such as tumor style, behavior, size, location, and growth status. [9].

Brain tumors are classified as either primary or metastatic, depending on their origin. The origin of the cells in primary ones is brain tissue cells, but cells in metastatic ones become malignant in any other region of the body and move into the brain. [1]

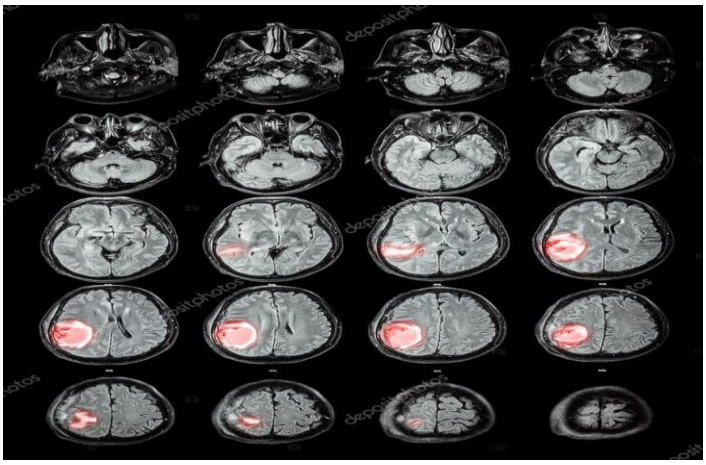


Figure 1. 1:MR imaging for brain tumor

3 Peritumoral brain edema (PTBE):

Cerebral edema is swelling caused by abnormal accumulation of water in the brain parenchyma, is seen commonly in the clinic. Peritumoral brain edema (PTBE) also impacts patient cognition and therefore should be taken into consideration when planning the target area for radiotherapy. In general, PTBE is associated with an incomplete blood tumor barrier and excess tissue space arising from aberrant metabolism of water in tumor tissue. This results in accumulation of a large volume of liquid and soluble components of the blood plasma. Recent advances in molecular biology have enabled the identification of basic molecular mechanisms underlying PTBE, but many issues still remain to be resolved. Only with deeper investigation of the molecular mechanism can we work out more effective strategies for controlling PTBE. [10]

A developing tumor and any associated cerebral edema may cause a rise in intracranial pressure (ICP), disturbing the brain's homeostatic environment. Many of the negative symptoms linked with brain cancer, such as headaches, nausea, and seizures, are caused by a rise in ICP. If left unchecked, ICP can lead to patient death. [11]

For these reasons, it has been suggested that measuring the degree of edema or variations in ICP may give enhanced prognostic potential in terms of early illness diagnosis and better tumor border delimitation.[10]

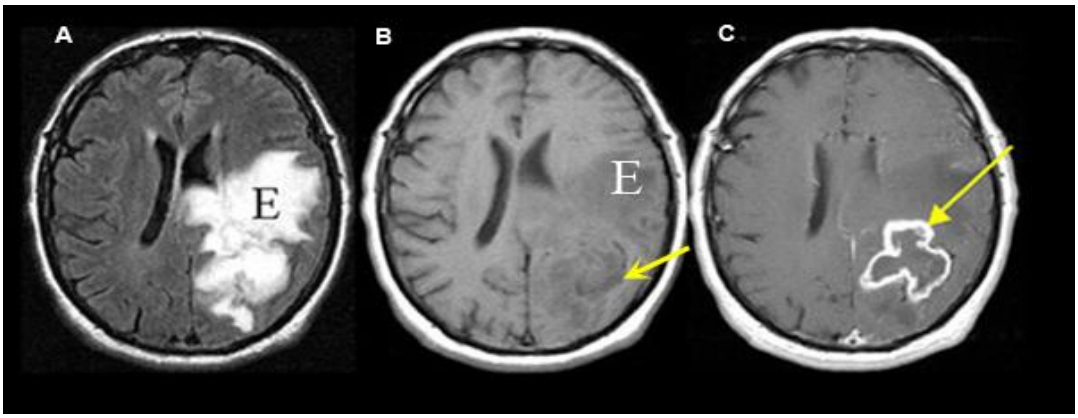


Figure 1. 2 :MR imaging for brain Tumor associated with edema (E).

4 Magnetic Resonance Imaging (MRI):

There is a radical change in the medical field due to various types of new imaging techniques. The most popular imaging techniques are magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET), single-photon emission computed tomography (SPECT), Ultrasound, and X-ray. These medical imaging techniques are used to detect complex diseases like brain tumors, COVID-19, cancerous cells or CNS cancer in humans.[12]

MRI is a more popular non-invasive in vivo imaging technology that use radio frequency waves to stimulate target tissues, causing them to create internal image under the influence of a very intense magnetic field. It used to offer useful information regarding the shape, size, location, and metabolism of brain tumors, which aided in diagnosis. While these modalities are used in conjunction to offer the most thorough information about brain tumors, MRI is regarded the standard approach due to its strong soft tissue contrast and widespread availability. [1]

The most common imaging test for the brain and spinal cord is magnetic resonance imaging (MRI). It's frequently used to aid in the diagnosis of:

- Aneurysms of cerebral vessels
- Disorders of the eye and inner ear
- Multiple sclerosis
- Spinal cord disorders
- Stroke
- Tumors
- Brain injury from trauma

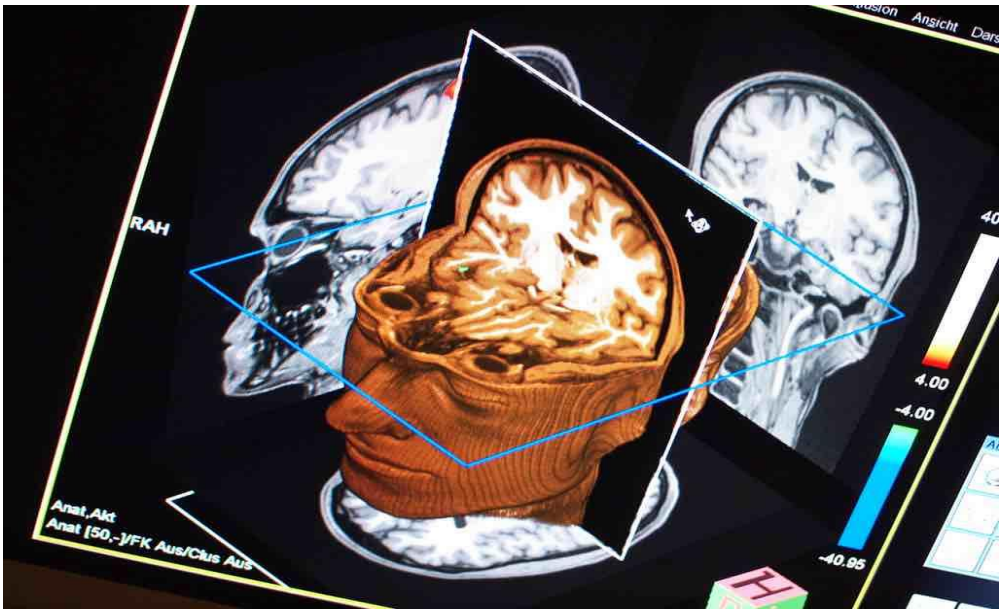


Figure 1.3: 3D image MRI slices

The functional MRI of the brain is a unique form of MRI (fMRI). It generates pictures of blood flow to specific brain regions. It may be used to look at the architecture of the brain and figure out which regions of the brain are in charge of important functions. This aids in the identification of critical language and movement control regions in the brains of individuals undergoing brain surgery. Damage from a head injury or diseases like Alzheimer's disease can also be assessed using functional MRI. [13]

The MRI machine looks like a long narrow tube that has both ends open. The patient lies down on a movable table that slides into the opening of the tube. A technologist monitors him from another room. he can talk with the person by microphone. [13]

The MRI equipment surrounds the patient with a powerful magnetic field and directs radio waves at his body. It is a painless process. There are no moving things around him, and he don't feel the magnetic field or radio waves. In some cases, a contrast material, usually gadolinium, will be injected into a vein in his hand or arm via an intravenous line. Certain details are enhanced by the contrast material. Gadolinium produces allergic reactions in a small minority of patients. [13]



Figure 1.4: images of MRI machine equipment

An MRI can last anywhere from 15 minutes to more than an hour. The patient must hold still because movement can blur the resulting images. During a functional MRI, he might be asked to perform a number of small tasks, such as tapping the thumb against the fingers, rubbing a block of sandpaper or answering simple questions. This helps pinpoint the portions of the brain that control these actions. [13]

4.1 Modalities of MR imaging:

Different modalities of MR imaging enhance separate components of a brain tumor. It consists of Four sequences are T1-weighted MRI (T1), T2-weighted MRI (T2), T1-weighted MRI with gadolinium contrast enhancement (T1-C) and Fluid Attenuated Inversion Recovery (FLAIR).

4.1.1 T1-weighted MRI (T1):

The T1 image are used for distinguishing healthy tissues.

4.1.2 T2-weighted MRI (T2):

The T2 image are used to delineate the edema region which produces bright signal on the image.

4.1.3 T1-weighted MRI with gadolinium contrast enhancement (T1-C):

In T1-C images, the tumor border can easily be distinguished by the bright signal of the accumulated contrast agent in the active cell region of the tumor tissue. Since necrotic cells do not interact with the contrast agent, they can be observed by hypo intense part of the tumor core making it possible to easily segment them from the active cell region on the same sequence.[1]

4.1.4 *Fluid Attenuated Inversion Recovery (FLAIR):*

The signal of water molecules is decreased in FLAIR pictures, which aids in identifying the edema area from the Cerebrospinal Fluid (CSF).

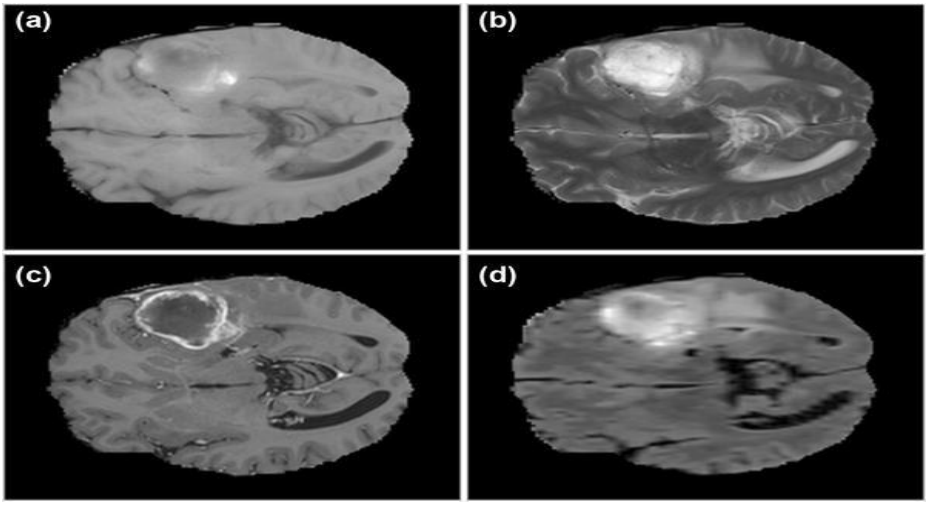


Figure 1.5: Brain multi-modality MRI images showing a) T1, b) T2, c) T1C and d) FLAIR.

5 *Classification of brain tumor:*

MRI is the most commonly used imaging solution to detect the tumor location, size, and morphology[14].

Based on this grading system, which is determined by the pathologic evolution of the tumor, a malignant tumor can be classified into the following categories:

5.1 *High-Grade Glioma[1] (HGG):*

are well-differentiated; these tend to exhibit benign tendencies and foreshadow a better prognosis for the patient. However, they have a uniform rate of recurrence and increase in grade over time, so can end up being classified as malignant [15]

5.2 *Low-Grade Glioma (LGG):*

are undifferentiated; these are malignant and carry a worse prognosis.

[1] Glioma is the medical name of brain tumor. A glioma is a type of tumor that starts in the brain or spine. It is called a glioma because it arises from glial cells. Gliomas make up about 30% of all brain and central nervous system tumors and 80% of all malignant brain tumors.

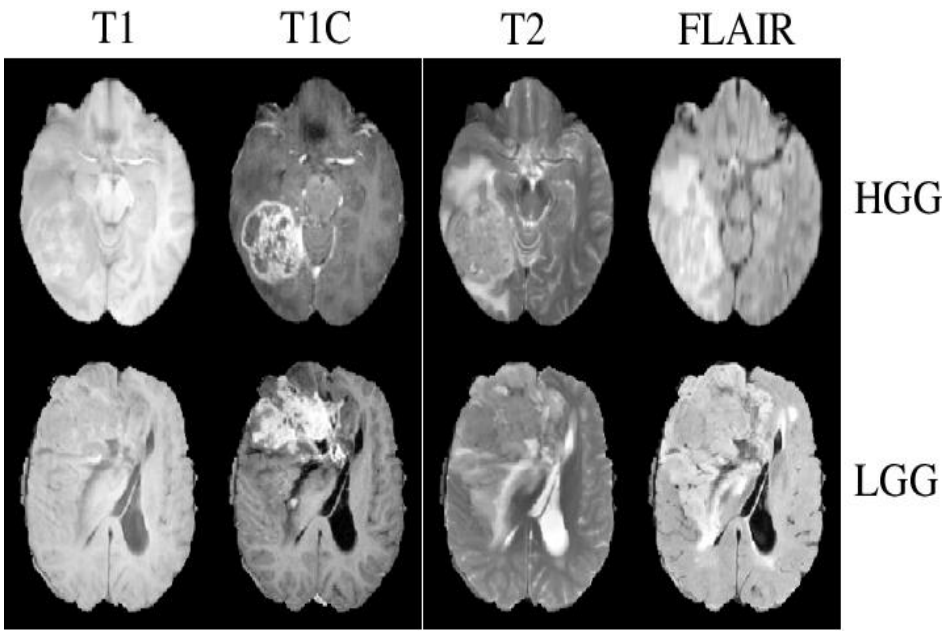


Figure 1.6:MR images, of the two categories (HGG, LGG)

6 Methods for Brain Tumor Image Segmentation:

Brain tumor detection and segmentation deals with separating the tumor portions from the normal tissues, to achieve this separation there are some methods grouped into three categories based on the degree of required human interaction, manual methods, semi-automatic methods and fully automatic methods. [1]

6.1 Manual Segmentation Methods:

Manual segmentation necessitates the utilization of multi-modality information provided by MRI scans, as well as anatomical and physiological result acquired via training and experience. Procedure involves the radiologist going through multiple slices of images slice by slice, diagnosing the tumor and manually drawing the tumor regions carefully. Apart from being a time-consuming task, manual segmentation is also radiologist dependent and segmentation results are subject to large intra and inter-rater variability. However, manual segmentations are widely used to evaluate the results of semi-automatic and fully automatic methods[1], [16].

6.2 Semi-Automatic Segmentation Methods:

Semi-automatic methods require user interaction for three main purposes: initialization, intervention or feedback response and evaluation.

- **Initialization:**

Is typically performed by defining a region of interest containing the approximate tumor region to be analyzed by the automated algorithm. Pre-processing technique parameters can also be modified based on the input images.

- **Intervention or feedback response:**

In addition to initialization, automated algorithms can be directed toward a desired output during the process by receiving feedback and making modifications in response.

- **Evaluation:**

In addition, the user can evaluate the results and modify or repeat the process if they are not satisfied.

A unique classification approach was used in a recent semi-automatic method. The segmentation problem was changed into a classification issue in this method, and a brain tumor was segmented by training and classifying inside the same brain only. Machine learning classification approaches for brain tumor segmentation, in general, need a large number of brain MRI images (with known ground truth) from various cases to train on.

Despite the fact that semi-automatic brain tumor segmentation approaches take less time than manual approaches and produce more accurate results, they are nevertheless subject to intra and inter rater/user variability. As a result, the majority of current brain tumor segmentation research is centered on fully automated approaches.[1]

6.3 Fully Automatic Segmentation Methods:

There is no need for operator interaction in fully automatic brain tumor segmentation methods. To fix the segmentation problem, artificial intelligence and previous knowledge are primarily merged.[1]

7 CNN for medical imaging

The initial step toward diagnosis is the delineation of the tumor and its components, also known as tumor area segmentation, using several modalities. This technique is carried performed by radiologists in a clinical setting, which takes time, and manual segmentation gets more difficult as the number of patients increases. As a result, to conduct segmentation operations and save radiologist work, automated algorithms are necessary. The diffused tumor boundaries and partial volume effect in MRI make segmentation of the distinct sections of the tumor on various MR imaging modalities even more difficult. Deep Learning approaches, particularly Convolutional Neural Networks (CNN), have recently obtained state-of-the-art results in the segmentation of various tumor components from a variety of MR images. Organs are typically imaged as 2-D entities because to the volumetric nature of medical pictures, and as a result, we use the nature of 2D CNN based architectures for segmentation.[17]

8 Related work:

The method of segmentation of a brain tumor is challenging due to the heterogeneous intensities of the tumor and its ambiguous boundaries. To compensate for the heterogeneity of the magnetic field, intensity normalization or bias field correction is frequently used [1]. Various investigations have used intensity, locality, and texture as common features. Table 1 summarizes the different based approaches used in segmentation and detection of brain edema and tumor detection from MRI Images, along with a brief summary of previous some research work. Only a few approaches are fully automated, while the others require user interaction.

Pereira (2016) [17] suggested a three convolution block (11 levels of depth) consisting of six convolutional layers with 3×3 filters, two max-pooling layers, and three fully-connected layers. The network authors conducted prepossessing to equalize intensity across all pictures by normalizing, as well as noise filtering by determining the standard deviation and mean intensity value across all training photos before training. In the BraTS dataset, the suggested model has an accuracy of 88% percent for the whole tumor, 83% percent for the core tumor, and 77% percent for an active tumor.

Salma X Sun (2019) [16] proposed a SegNet for automated brain tumor segmentation by training all modalities individually and integrating the SegNet output after post-processing. To begin, the inputs are prepossessed to remove undesired artifacts using normalization and bias field correction, which improves segmentation performance. SegNet is used to train each of the four MRI modalities individually. The architecture consists of a pair of the encoder (down sampling) and decoder (up sampling). The encoder employs 13 convolutional layers with 3×3 filters, batch normalization layers, ReLU layers, and max-pooling layers with 2×2 filters. In order to match the encoder, the decoder likewise comprises 13 convolutional layers. The decoder's high-dimensionality features are sent to SoftMax layers, which categorize each class pixel independently. The accuracy of the segmentation methodology was 85% percent for the whole tumor, 81% percent for the core tumor, and 79% percent for the augmenting tumor.

Wang T(2009) [18] proposed a Fluid vector flow (FVF) is introduced to address the problem of unsatisfactory capture range and poor convergence for concavities.

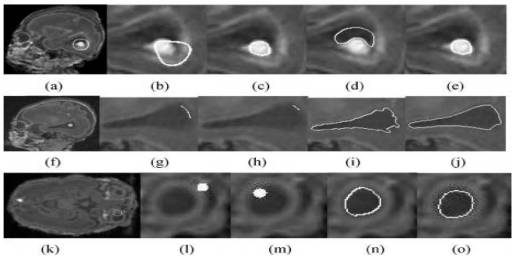


Fig. 11. (a) Image with an initial contour on the outside of the high-intensity region (intraventricular hemorrhage), and the results (zoomed-in) of (b) GVF, (c) BVF, (d) MAC, and (e) FVF. (f) Image with a small initial contour at the inside of the brain ventricle, and the results (zoomed-in) of (g) GVF, (h) BVF, (i) MAC, and (j) FVF. (k) Image with an initial contour overlapping the eye, and the results (zoomed-in) of (l) GVF, (m) BVF, (n) MAC, and (o) FVF.

Harati et al. [19] demonstrated an improved fuzzy connectedness (FC) algorithm, where seed points are selected automatically to segment the tumor region.

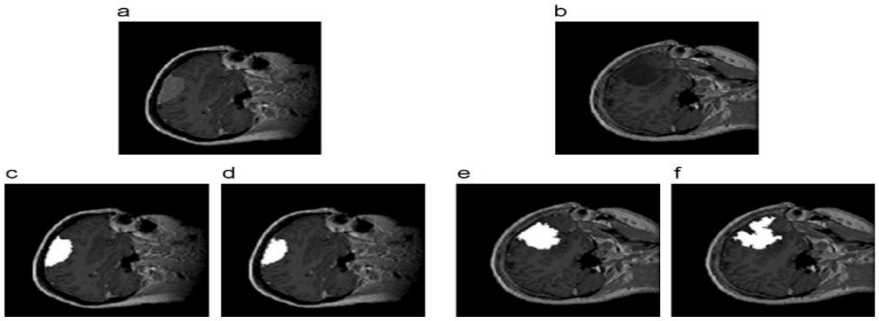


Fig. 6. Comparison between the proposed segmentation algorithm and general fuzzy connectedness on two types of images including tumor with distinct and fading boundaries: (a) input image consists of tumor with distinct boundaries, (b) input image consists of tumor with fading boundaries, (c) segmentation result of tumor with distinct boundaries by proposed approach, (d) segmentation result of tumor with distinct boundaries by general fuzzy connectedness algorithm, (e) segmentation result of tumor with fading boundaries by proposed approach, and (f) segmentation result of tumor with indistinct boundaries by general fuzzy connectedness.

Saha et al. [20]proposed a fast novel method to locate the bounding box around tumor or edema using Bhattacharya coefficient. In their proposed clustering technique axial view of brain image is divided into left and right halves, and then a rectangle is used to compare the corresponding regions of left half with right half to find the most dissimilar region within the rectangle.

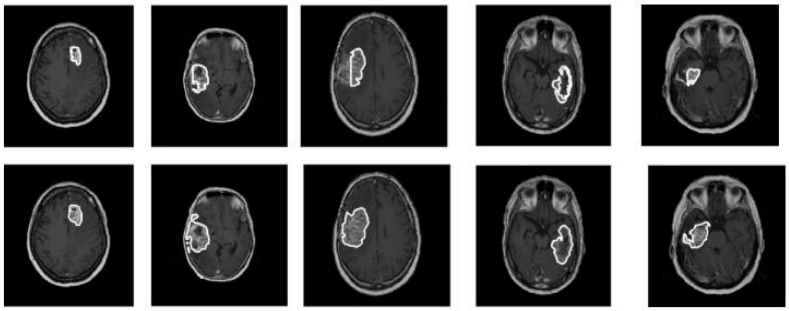


Fig. 17. Tumor detection from T1C axial brain MRI slices: top row, results of knowledge based thresholding algorithm; bottom row, results of Chan–Vese algorithm within bounding box found by FBB.

Zhu et al. [21] proposed a semiautomatic brain tumor segmentation method, where initial segmentation is performed through ITK-Snap tool. Voxel-based segmentation and deformable shape-based segmentation are combined into the software pipeline.

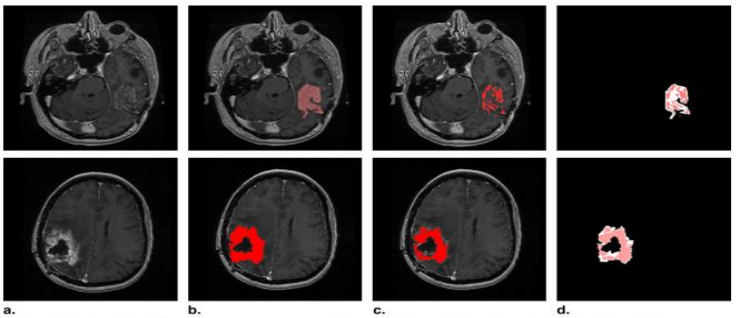
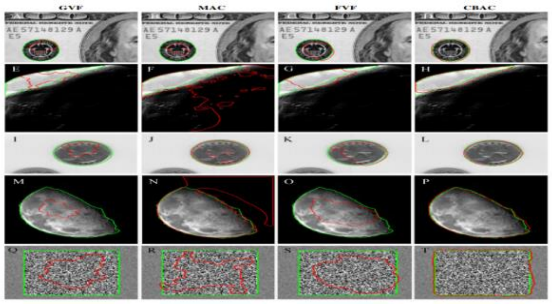


Figure 7. Difference between manual and semi-automatic segmentations: **(a)** original image; **(b)** manual segmentation; **(c)** semi-automatic segmentation; **(d)** difference between manual segmentation (*background white shape*) and semi-automatic segmentation (*highlighted red shape*).

Sachdeva et al. [22] used texture information with intensity in active contour model (ACM) to overcome the issue observed in previous techniques like FVF, boundary vector flow (BVF) and gradient vector flow (GVF). In previous techniques selection of false edges or false seeds corresponds to pre-convergence problem and selection of weak edges leads to over-segmentation due to the edema around the tumor.



Rexilius et al. [23]proposed a new region growing method for segmentation of brain tumor. Probabilistic model is used to achieve the initial segmentation, which is further refined by region growing to give better segmentation results. Global affine and non-rigid registration method is used to register multi-spectral histograms gathered from patients' data with a reference histogram.

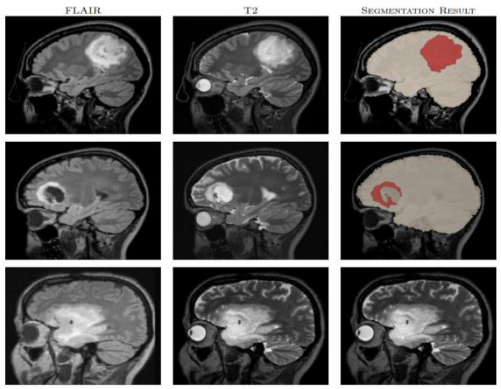


Figure 6. Segmentation results of different patient datasets. From left to right: Slice of FLAIR dataset, slice of T2 dataset, combined overlay of skull stripping and tumor segmentation result.

Corso et al. [24] used a top-down approach to distribute the product over generative model. Later, sparse graph is given as input to graph cut method, where each edge uses features to find similarity between neighboring nodes having the affinity. Segmentation by weighted aggregation (SWA) is used to provide the multi-level segmentation of data.

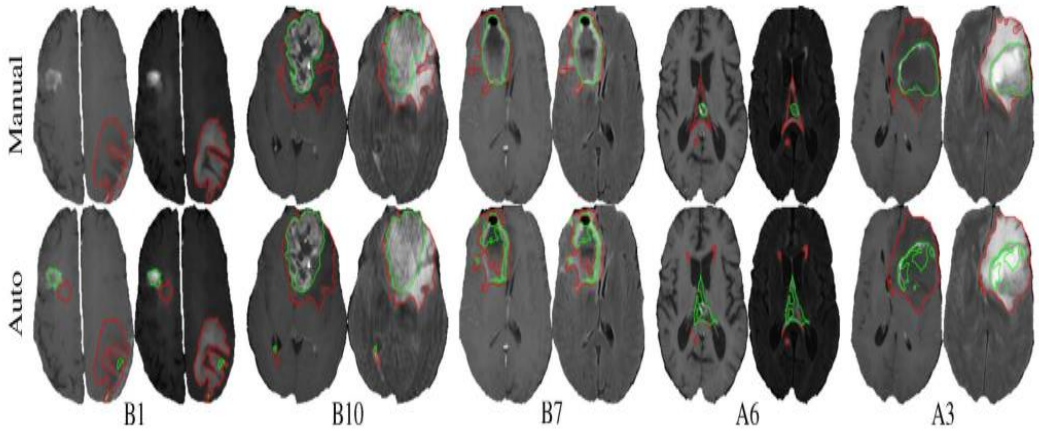


Fig. 7. These five cases represent the different failure modes in our current system. In each group, the left column shows the T1CE channel and the right column shows the FLAIR. Complete discussion in text.

Ruan et al. [25] proposed a supervised machine learning technique to track the tumor volume. The complete process is categorized into two main steps. In the first step to make it efficient and reduce computational time, only T1 modality is used to identify the abnormal area. In the second step, the abnormal area is extracted from all modalities and fused to segment the tumor.



Figure.3. Results of the classification corresponding to the Figure.2: first step (left), second step (right).

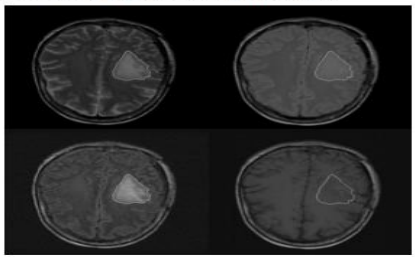


Figure.4. Results of the tumor segmentation for the first examination. The obtained contour is superposed on the : T2 (a), PD (b), FLAIR (c), T1 (d).

Irfan et al. [26] introduced a technique in which brain images are separated from non-brain part, and then ROI is used with the saliency information to bind the search of normalization cut (N-Cut) [14] method. Saliency information is the combination of multi-scale contrast and image curvature points. Multiscale contrast image is acknowledged when image is decomposed at multiple scales by using Gaussian pyramid (GP), and Euclidean distance is calculated with neighboring pixels at those scales.

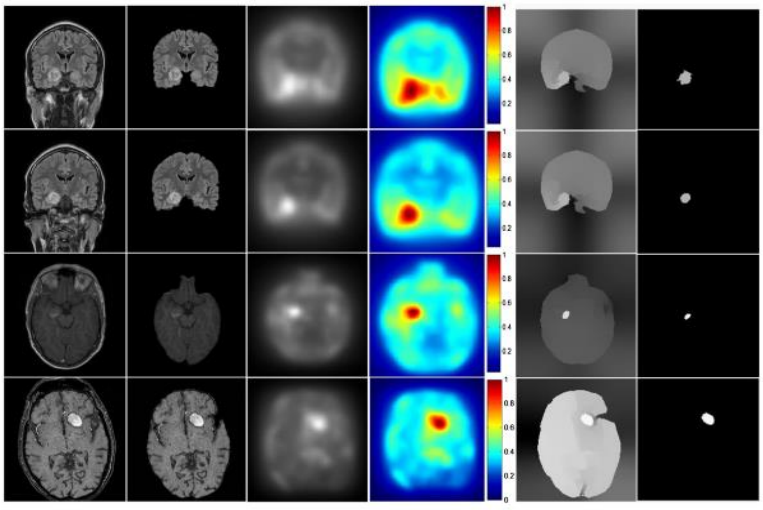


Fig. 13. Segmentation results: from left to right: input image, brain tissue, saliency gray map, saliency heat map, saliency weighted N-Cut segmentation, and segmented tumor region.

Festa J [27] used the random forest (RF) for the Automatic segmentation, where features include MR sequence intensities, neighborhood information, context information and texture. Post-processing is performed for the sake of good results.

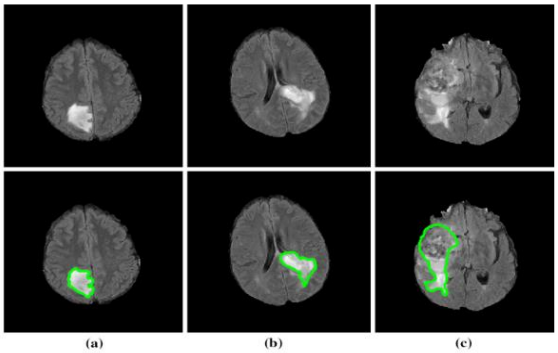


Fig. 7 FLAIR images with different tumour grades in *upper row* and their ground-truth manual segmentation of the FLAIR hyperintensity in the *lower row*. Tumour grades are: **a** grade II, **b** grade III and **c** grade IV

Zhao [28] used Markov random field (MRF) model on super voxels to automatically segment tumor.

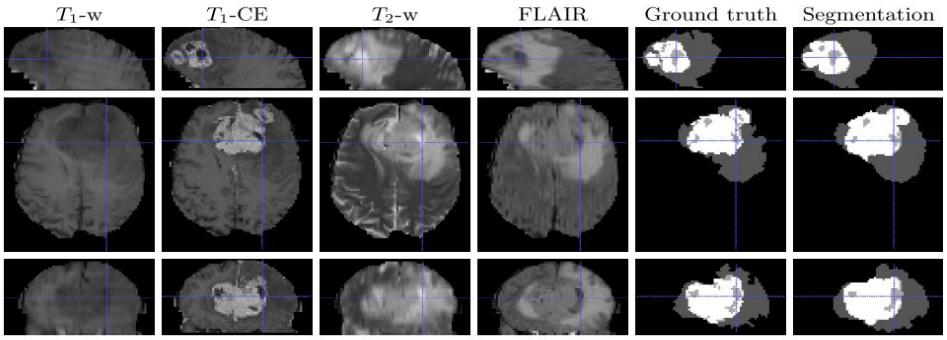
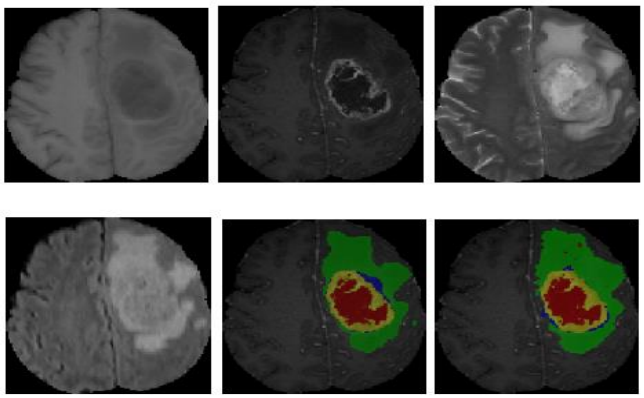


Fig. 4. Real high-grade case 15. From left to right: MR images, ground truth, and segmentation. From top to bottom: saggital, axial, and coronal views. Images are cropped.

Guo X [28] proposed a ACM combines the edge-based and region-based techniques, where user draws region of interest (ROI) in different images on the basis of tumor type and grade.



In Bahadure (2017) [29], the authors proposed discrete wavelet transformation (DWT) based techniques where the feature extraction techniques are used there. However, if these feature extraction techniques are used then the significant details of brain images lost.

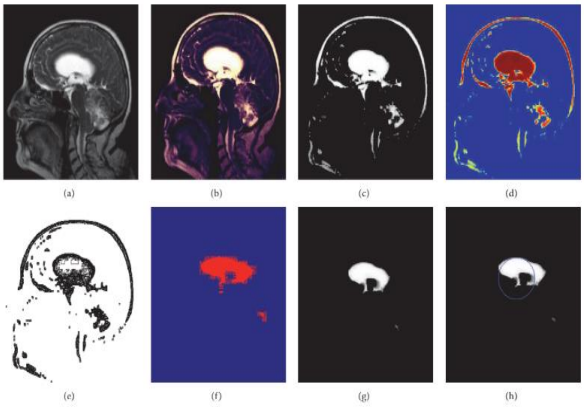
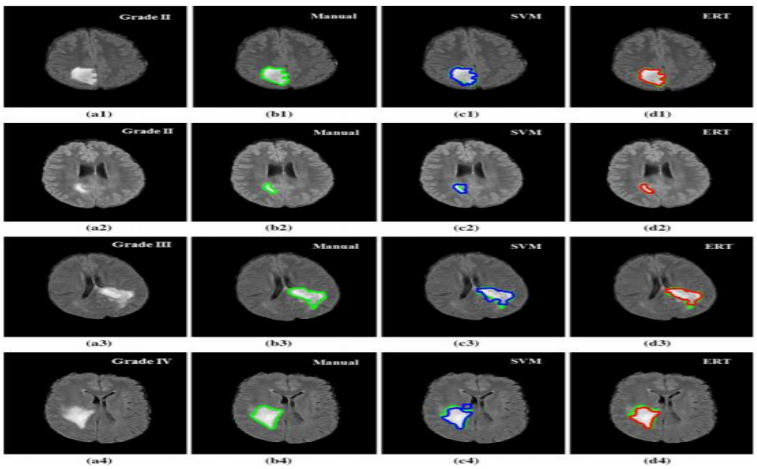


FIGURE 6: Experimental results of image 3. (a) Original image. (b) Enhanced image. (c) Skull-stripped image. (d) Wavelet decompose image. (e) Intense segmented image. (f) Dice overlap image. (g) Tumor region. (h) Area extracted tumor region.

In Soltaninejad (2017a) [27] , the researchers proposed superpixel based method for automated brain tumor identification and segmentation from Fluid-Attenuated Inversion Recovery Magnetic Resonance Imaging (FLAIR-MRI). The experimental results show that the pro- posed scheme improves the detection and segmentation performance by using Extremely randomized trees (ERT) classifier. For their own clinical dataset, the mean identification sensitivity, balanced error rate and the Dice overlap value for the tumor are 89.48%, 6% and 0.91, respectively; whereas, for the BRATS-2012 dataset, the corre- sponding assessment outcomes are 88.09%, 6% and 0.88, respectively.

Another learning based method was proposed in Soltaninejad et al. (2017b) [27] which used both machine-learned and hand-crafted features for automated segmentation of brain tumor from the multi-modal MRI images.



In Soltaninejad et al. (2018) [30] , the authors introduced a 3D super voxel-based segmentation method for tumor segmentation in multi- modal MRI brain images. They are extracted several feature sets for each super voxel, including first order intensity statistical features, histograms of text on descriptor to classify each super voxel into the following three classes tumor core, edema or healthy brain tissue by using random forest (RF) classifier. The method was evaluated based on two datasets such as (i) local clinical dataset: 11 images of patients and (ii) BRATS 2013 clinical dataset: 30 multimodal images. The experimental results shown that the average detection sensitivity of tumor using local clinical dataset is 86% with balanced error rate (BER) 7% and the Dice score for automatic tumor segmentation is 0.84. The resembling results of the BRATS 2013 dataset are 96%, 2% and 0.89, respectively.

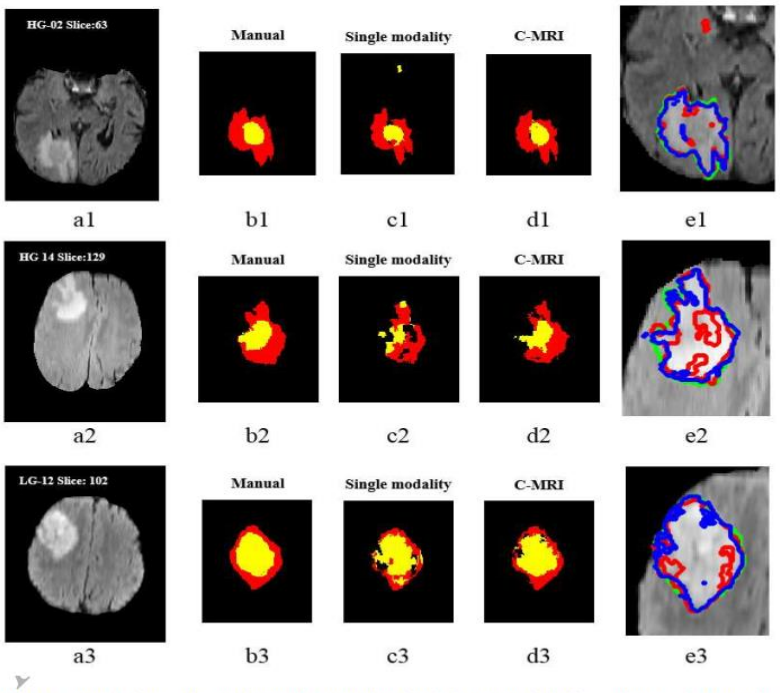


Fig. 11. Segmentation results overlay on the ground truth (whole tumour including oedema and core), using single (FLAIR) and multimodal (conventional MRI including FLAIR, T1, T1-contrast and T2), for three different cases with grade IV tumours; a) FLAIR image, b) manual segmentation of core (yellow region) and

Table 1. 1: Overview of recent methods on the basis of different features.

No	Work	Modalities	Method	Time	Automatic
1	Wang et al.	*T1*	FVF and brain tumor segmentation	5 s	SA
2	Harati et al.	T1C	Fully automatic Fuzzy Connectedness algorithm	2.5 m	FA
3	Saha et al.	T1C	Quick detection of tumor using symmetry	0.5 m	FA
4	Zhu et al.	T1C, T2	Software pipeline with post-processing	4 m	SA
5	Sachdeva et al.	T1, T1C, T2	Texture features +ACM	–	SA
6	Rexilius et al.	T1C, T2, FLAIR	Region growing + multi - spectral histogram model adaption	10 m	SA
7	Corso et al.	T1, T1C, T2, FLAIR	Generative affinity model and graph cut method are used with SWA	7 m	FA
8	Ruan et al.	T1, T1C, T2, FLAIR, PD	Multi-modality MRI with SVM classification	5 m	FA
9	Irfan et al.	T1, T1-weighted, T2, T2- weighted	Prioritization of brain MRI volumes using image perception model	_	FA
10	Festa et al.	T1, T1C, T2, FLAIR	Multi-sequence MRI using RF	20–25 m	FA
11	Zhao et al.	T1, T1C, T2, FLAIR	MRF + super voxels	4 m	FA
12	Guo et al.	T1, T1C, T2, FLAIR	Semiautomatic segmentation using ACM	1 m	SA
13	Soltaninejad et al. (2018)	T1c, T2	Super voxel based random forests	_	_
14	Soltaninejad et al. (2017b)	FLAIR, T1, T1c, T2 FLAIR,	FCNN+Texton+RF U-Net	_	FA
15	Soltaninejad et al. (2017a)	FLAIR	SBGFRLS	_	SA
16	Bahadure et al. (2017)	T1, T1c, T2, FLAIR, PD	Super pixel based extremely randomized trees	_	FA

Different dataset is used. FA denotes fully automatic, and SA denotes semiautomatic.

9 Conclusion :

This chapter allowed us to define and make clear the most key words of our search working represented on brain edema and tumor detection from MR imaging using hybrid intelligent algorithms such as deep learning. In this way we gave a glimpse to the next chapter, where we discuss our proposed approach.

Chapter 02

Deep Learning for image segmentation

1 Introduction :

In this chapter, we are going to talk about Deep learning approach in the Artificial intelligence domain, there principals, where and when to use DL and how good it is for image segmentation, we will also talk about how to use these methods for both supervised and non-supervised classification, finally the most popular algorithm used in DL.

2 Definition:

Deep learning is a collection of machine learning algorithms that model high-level abstractions in data utilizing architectures comprised of many nonlinear transformations, he is a branch of Machine Learning, which is a subfield of Artificial Intelligence.[31]

Artificial neural networks underpin deep learning technologies (ANNs). These ANNs are constantly fed learning algorithms and ever-increasing volumes of data in order to improve the efficiency of training procedures. The larger the data volumes are, the more efficient this process is. The training process is referred to as "deep" since a neural network covers an increasing number of layers over time.[32]

The aim of deep learning is to learn feature levels of increasing abstraction with the minimum human contribution. The feature hierarchy is learned by assembling lower level features to produce higher level features in the hierarchy.

Automatically learning features at several levels of abstraction enables a system to learn complicated functions mapping input to output directly from data, rather than relying only on human-crafted features.[33]

In other words, deep learning entails automating the feature engineering process, in which multiple levels of representation are obtained by composing simple but non-linear modules, each of which transforms the representation at one level (beginning with the raw input) into a representation at a higher, slightly more abstract level. When a sufficient number of such changes are combined, even the most difficult functions may be learnt. Higher layers of representation accentuate aspects of the input that are significant for discriminating while suppressing irrelevant variations in classification tasks. [34]

3 Deep learning history:

Below is a brief history of neural networks highlighting key events:

▪ 1943: McCulloch & Pitts show that neurons can be combined to construct a Turing machine (using ANDs, ORs, & NOTs) [35] .

▪ 1958: Rosenblatt shows that perceptron's will converge if what they are trying to learn can be represented.

▪ 1969: Minsky & Papert show the limitations of perceptron's, killing research in neural networks for a decade.

▪ 1985: The backpropagation algorithm by Geoffrey Hinton et al revitalizes the field.

▪ 1988: Neocognitron: a hierarchical neural network capable of visual pattern recognition.

▪ 1998: CNNs with Backpropagation for document analysis by Yan LeCun.

Since 2006, deep structured learning, or more commonly called deep learning or hierarchical learning, has emerged as a new area of machine learning research. [36]During the past several years, the techniques developed from deep learning research have already been impacting a wide range of signal and information processing work within the traditional and the new, widened scopes including key aspects of machine learning and artificial intelligence. A series of workshops, tutorials, and special issues or conference special sessions in recent years have been devoted exclusively to deep learning and its applications to various signal and information processing areas. These include according to [37]:

• 2008 NIPS Deep Learning Workshop;

• 2009 NIPS Workshop on Deep Learning for Speech Recognition and Related Applications;

• 2009 ICML Workshop on Learning Feature Hierarchies;

• 2011 ICML Workshop on Learning Architectures, Representations, and Optimization for Speech and Visual Information Processing;

• 2012 ICASSP Tutorial on Deep Learning for Signal and Information Processing;

• 2012 ICML Workshop on Representation Learning;

• 2012 Special Section on Deep Learning for Speech and Language Processing in IEEE Transactions on Audio, Speech, and Language Processing (T-ASLP, January);

• 2010, 2011, and 2012 NIPS Workshops on Deep Learning and Unsupervised Feature Learning;

• 2013 NIPS Workshops on Deep Learning and on Output Representation Learning;

• 2013 Special Issue on Learning Deep Architectures in IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI, September).

• 2013 International Conference on Learning Representations;

• 2013 ICML Workshop on Representation Learning Challenges;

• 2013 ICML Workshop on Deep Learning for Audio, Speech, and Language Processing;

• 2013 ICASSP Special Session on New Types of Deep Neural Network Learning for Speech Recognition and Related Applications.

4 How does deep learning attain such impressive results?

Deep learning achieves recognition accuracy at very higher levels even sometimes exceeds human-level performance in some tasks for example speech pathology detection in audio.

Deep learning was theorized very long time ago, back in the 1950s, and it has become useful only recently because of:

1. The large amounts of data that Deep learning need. classifying objects in images for example, requires thousands or millions of images.

2. Hardware availability that Dl requires (The huge computing potentials). [38]

5 Principe:

Neural network architectures are widely used in deep learning that can have more than 150 hidden layers, which is why they are called deep. Typical neural networks have 2 to 3 hidden layers. **Figure 2.1** is a diagram illustrating that neural networks consist of layers consisting of a set of interconnected nodes. [38]

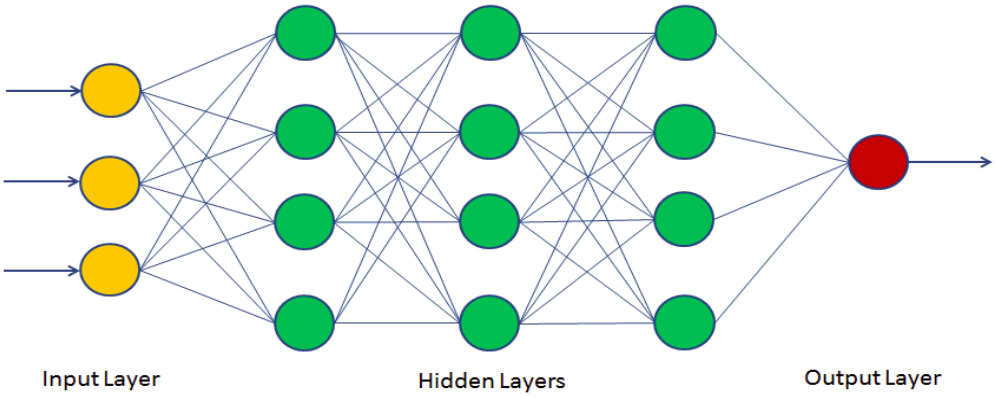


Figure 2. 1 : Neural networks architecture in DL

6 Classification of deep Learning:

Deep learning is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised, unsupervised and Reinforcement Learning

6.1 Supervised:

Supervised learning is a learning technique that uses labeled data. In the case of supervised DL approaches, the environment has a set of inputs and corresponding outputs $(xt, yt) \sim \rho$. For example, if for input xt, the intelligent agent predicts $\hat{yt} = f(xt)$, the agent will receive a loss value $l(yt, \hat{yt})$. The agent will then iteratively modify the network parameters for better approximation of the desired outputs. After successful training, the agent will be able to get the correct answers to questions from the environment. There are different supervised learning approaches for deep leaning including Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) including Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). [39]

6.2 Semi-supervised

Semi-supervised learning is learning that occurs based on partially labeled datasets (often also called reinforcement learning (DRL)). . In some cases, DRL and Generative Adversarial Networks (GAN) are used as semi-supervised learning techniques. Additionally, RNN including LSTM and GRU are used for semi-supervised learning as well.

6.3 Unsupervised

Unsupervised learning systems are ones that can without the presence of data labels. In this case, the agent learns the internal representation or important features to discover unknown relationships or structure within the input data. Often clustering, dimensionality reduction, and generative techniques are considered as unsupervised learning approaches. There are several members of the deep learning family that are good at clustering and non-linear dimensionality reduction, including Auto Encoders (AE), Restricted Boltzmann Machines (RBM), and the recently developed GAN.

In addition, RNNs, such as LSTM and RL, are also used for unsupervised learning in many application domains.

6.4 Deep Reinforcement Learning (DRL):

Deep Reinforcement Learning is a learning technique for use in unknown environments. DRL began in 2013 with Google Deep Mind. From then on, several advanced methods have been proposed based on RL. Sometimes this approach is called semi-supervised learning as well. There are many semi-supervised and un-supervised techniques that have been implemented based on this concept. In RL, we do not have a straight forward loss function, thus making learning harder compared to traditional supervised approaches. The fundamental differences between RL and supervised learning are: first, you do not have full access to the function you are trying to optimize; you must query them through interaction, and second, you are interacting with a state-based environment: input xt depends on previous actions. Depending upon the problem scope or space, you can decide which type of RL needs

to be applied for solving a task. If the problem has a lot of parameters to be optimized, DRL is the best way to go. If the problem has fewer parameters for optimization, a derivation free RL approach is good. An example of this is annealing, cross entropy methods, and SPSA. We conclude this section with a quote from Yann LeCun: **"If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing, and reinforcement learning would be the carry."** – Yann LeCun

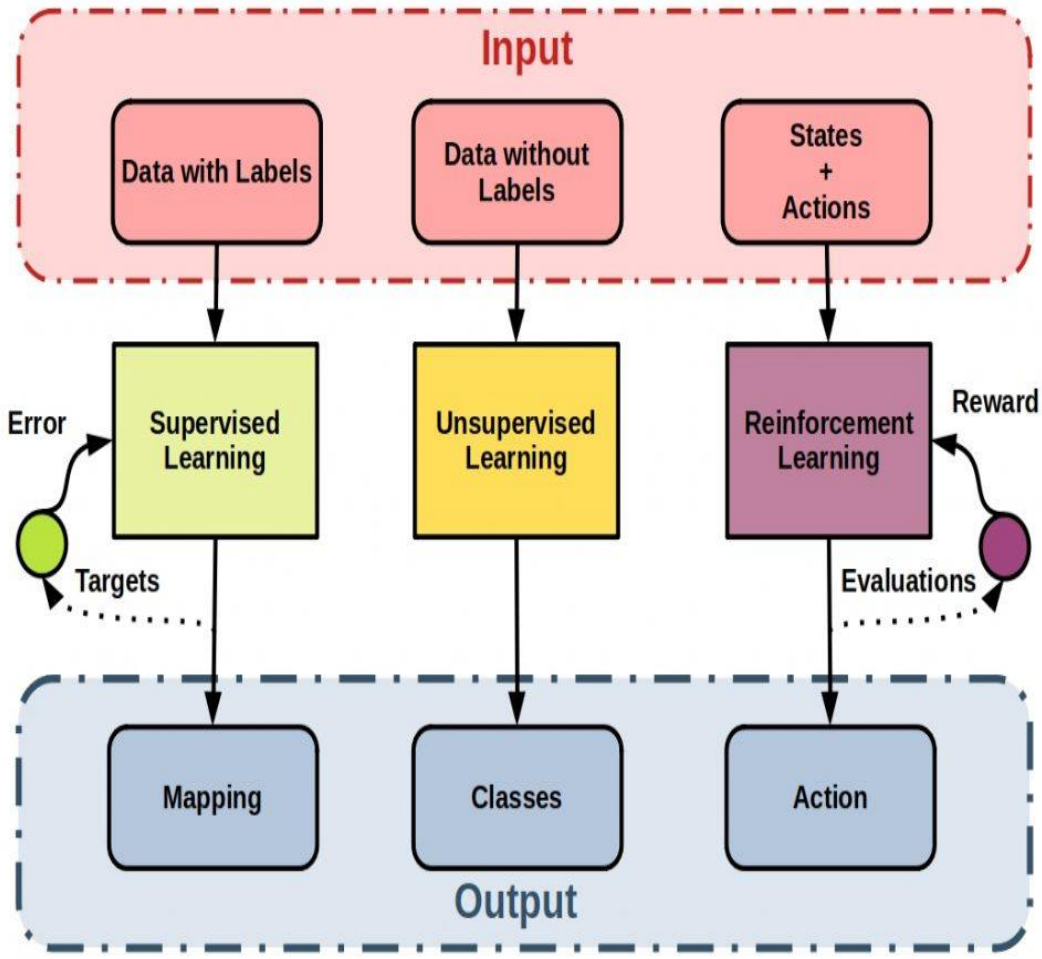


Figure 2.2: The different classification of deep learning

7 Why deep learning:

7.1 Universal learning approach:

Deep learning is now being used in almost every field. As a consequence, this method is sometimes referred to as a universal learning method. [39]

7.2 Robust:

Deep learning methods do not necessitate the development of functionality in advance. Features that are best for the job at hand are immediately mastered. As a consequence, the data's robustness to natural fluctuations is taught automatically.

7.3 Generalization:

The same deep learning approach can be applied to a variety of programs and data types. This method is useful where there isn't enough evidence to solve the problem and it is known as a transfer learning method. There are several papers that have been published based on this concept.

7.4 Scalability

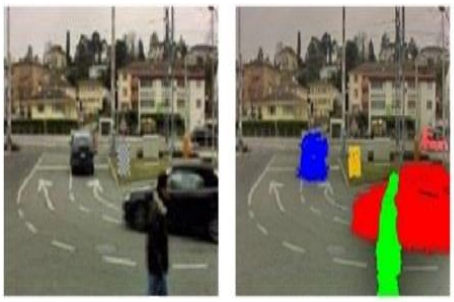
Deep learning is a highly scalable process. Microsoft described the ResNet network in a 2015 paper. This network has 1202 layers and is often implemented on supercomputers. At Lawrence Livermore National Laboratory (LLNL), a large effort is underway to create architectures for networks of thousands of nodes.

8 When and where to apply DL:

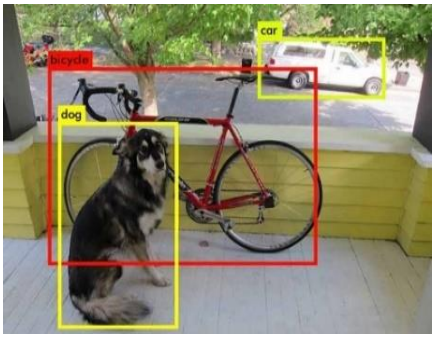
DL is used in a variety of cases where artificial intelligence is beneficial:

1. The challenge is much too large for our meager reasoning skills to handle (calculation webpage ranks, matching ads to Facebook, sentiment analysis).
2. When time passes, the problem's solution evolves (tracking, weather prediction, preference, stock, price prediction)
3. Humans are incapable of explaining their knowledge (speech recognition, vision and language understanding)
4. Solutions must be tailored to the specific circumstances (biometrics, personalization).
5. Due to the absence of a human specialist (navigation on Mars)

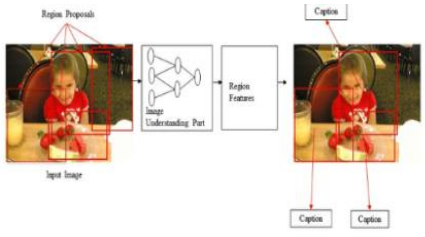
Deep learning is now being used in almost every field. As a consequence, this method is sometimes referred to as a universal learning method. **Figure 2.3** depicts few examples of implementations. [39]



Object localization



Object detection



Media and entertainment



Image or Video Captioning



Image or video Segmentation



Autonomous Car



Machine translation



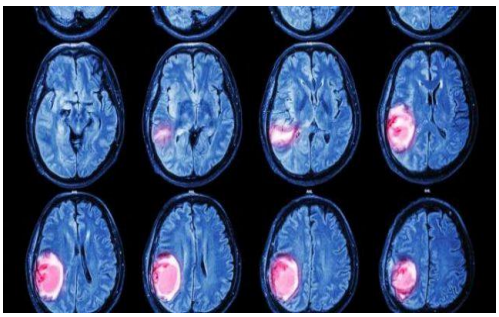
Speech recognition



Security and Defense



Medicine and bio



Brian Cancer Detection



Skin cancer recognition

Figure 2.3: Example images where DL is applied successfully and achieved state of-the-art performance.

9 Deep learning algorithms used:

Deep Learning has proven to be a very useful technique over the last few decades due to its capacity to process vast quantities of data. The popularity of using hidden layers has surpassed that of conventional techniques, especially in pattern recognition. Convolutional Neural Networks are one of the most widely used deep neural networks.

9.1 Convolutional neural networks CNN:

Convolutional neural networks or CNNs is a network of directly propagated neurons (feedforward) with several types of special layers. For example, convolutive layers apply a filter to the input image (or sound) by dragging this filter over all the incoming signal, in order to produce an activation map (activation map) with n dimensions.

CNN are a specialized kind of neural network for processing data that has a known grid-like typology. Examples includes time series data, which can be thought as 1-D grid taking samples at regular time intervals, and image data, which can be thought as 2-D grid of pixels. The name “convolutional neural networks” indicates that the network employs a mathematical operation called “convolution” which is a specialized kind of linear operation. To be specific, convolutional networks are simply neural network that uses convolutions in place of general matrix multiplication in at least one of their layers. The motivation behind using convolutional networks in place of traditional neural networks is reducing intra-modular network interactions, sharing of parameters and learning of equivariant representations.[40]

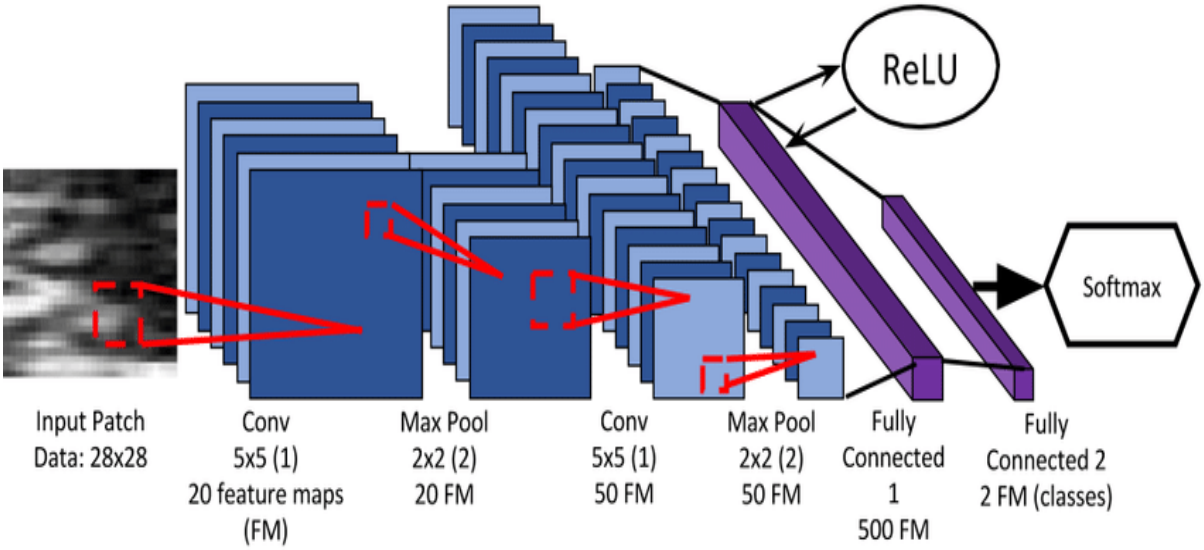


Figure 2.4: Generic representation of the CNN architecture.

Before getting more into detail, we would like to define some key concepts related to neural networks: [41]

- An artificial neuron is a mathematical function designed to mimic the behavior of biological neurons. Artificial neurons are the building blocks of a neural network.
- A node's activation function specifies its output in response to an input or combination of inputs.
- A layer is an operation performed in an input that is part of the network.
- A weight is a parameter obtained from the layers.
- An epoch is a training iteration over the dataset.
- he learning rate is a positive scalar determining the size of the step for each training iteration.
- The batch size defines number of samples that are going to be propagated through the network. For example, suppose you have 1050 training samples and wish to set batch size to 100. The algorithm trains the network using the first 100 samples (from one to one hundred) from the training dataset. The network is then trained again using the second 100 samples (from 101st to 200th).

9.2 How Convolution Neural Networks Work:

Convolutive networks generally have several hidden layers that are organized in blocks, but some convolutive networks have hundreds of hidden layers. Hidden layers closer to the input layer detect very simple features such as edges and color gradients in the images. Then, the upper layers combine these simple features in more complex patterns. Finally, the layers close to the output combine these more complex patterns.

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below:

9.2.1 Convolutional layer

In the case of a CNN, the primary goal of convolution is to extract features from the input picture (feature map). By learning visual attributes from tiny squares of input data, convolution retains the spatial connection between pixels.

Convolutions are defined by two parameters:

- Size of extracted features (usually 3x3 or 5x5 pixels).
- The depth of the output characteristics map, which corresponds to the number of filters applied

To generate a feature map, a filter is slid across the input picture (convolution process). A separate feature map is obtained by convolution of another filter over the same picture. It is important to notice that the Convolution operation preserves the original image's local dependencies.

In fact, a CNN learns the values of these filters on its own during the training process (but factors such as the number of filters, filter size, network architecture, and so on must still be specified before to the training process). The more filters we have, the more picture characteristics are retrieved, and our network grows stronger at spotting patterns in previously unseen photos.

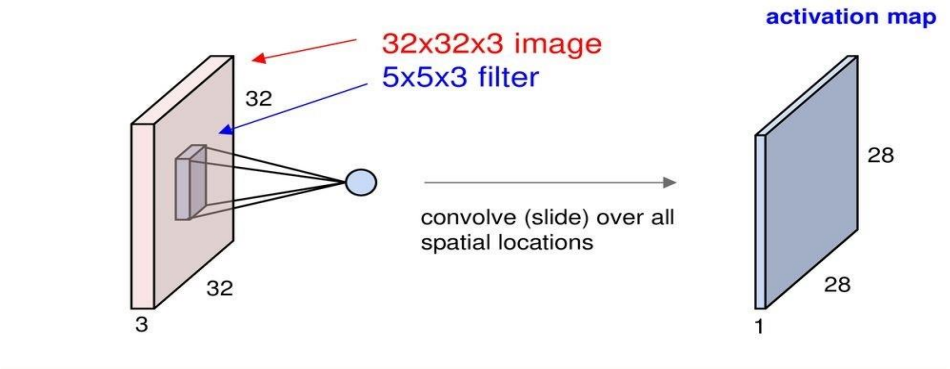


Figure 2.5: Formation and arrangement of convolution layer.

9.2.2 ***Pooling layer:***

A Convolutional Layer is usually followed by a Pooling Layer. This layer's principal objective is to decrease the size of the convolved feature map in order to reduce computational expenses. This is accomplished by reducing the connections between layers and operating independently on each feature map. Pooling operations are classified into several types based on the method used. In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer. [42]

The function of pooling is to progressively reduce the spatial size of the input representation. To be more exact, pooling:

- Makes the input representations (feature dimension) smaller and more manageable.
- Reduces the number of parameters and computations in the network.
- Makes the network invariant to small transformations, distortions and translations in the input image.

- Helps us arrive at an almost scale invariant representation of the image, also referred as is equivariant. This is very powerful since we can detect objects in an image no matter where they are located
- One of the main disadvantages of pooling is losing output spatial resolution.

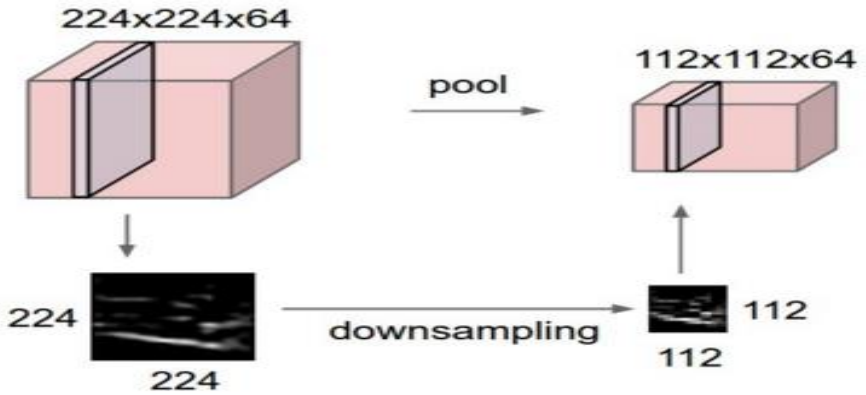


Figure 2.6: Pooling layer

9.2.3 ***Fully Connected Layer:***

The Fully Connected Layer is a Multi-Layer Perceptron (MLP) that uses an activation function in the output layer. An activation function of a node defines the output of that node given an input or set of inputs. The most common activation function used is a SoftMax function/classifier. The term "Fully Connected" implies that every neuron in the previous layer is connected to every neuron on the next layer.

The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset. [41]

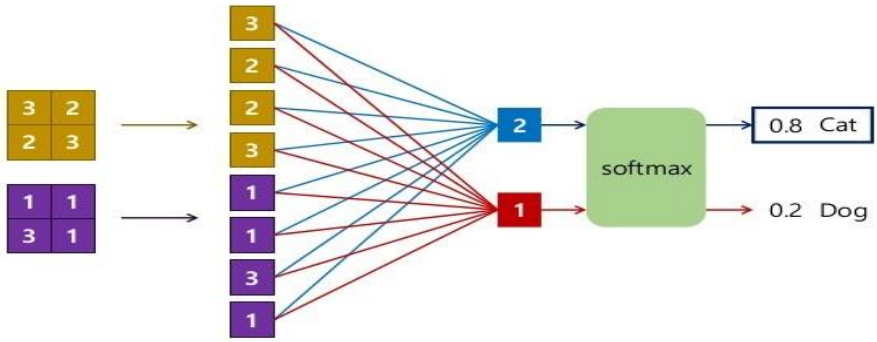


Figure 2.7: The structure of the fully connected layer.

9.3 Activation Function:

9.3.1 *ReLU function:*

Rectified Linear Unit (ReLU) is a non-linear operation. All negative pixel values in the feature map are replaced by zero using ReLU. Because much of the real-world data we want CNNs to learn is non-linear, the aim of ReLU is to bring non-linearity into CNNs. The ReLU operator can be stated as follows:

$$f(x) = \text{maximum } 0, x$$

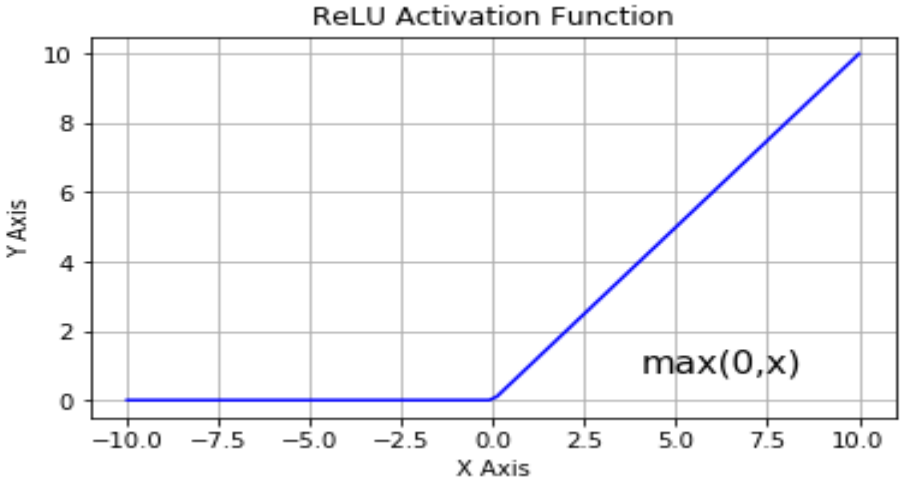


Figure 2. 8: Relu activation function curve

9.3.2 ***Max Pooling function:***

A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. When doing Max Pooling, we define a spatial neighborhood and take the largest element from the rectified feature map within that window. Instead of taking the largest element we could also take the average or sum of all elements in that window. In practice, the most common kind of Pooling used is Max Pooling since it has been shown to work better.[41]

Example:

In Max Pooling, we define a specific window and take the biggest element from the ReLU feature map from that window. **Figure.2.9** explains more:

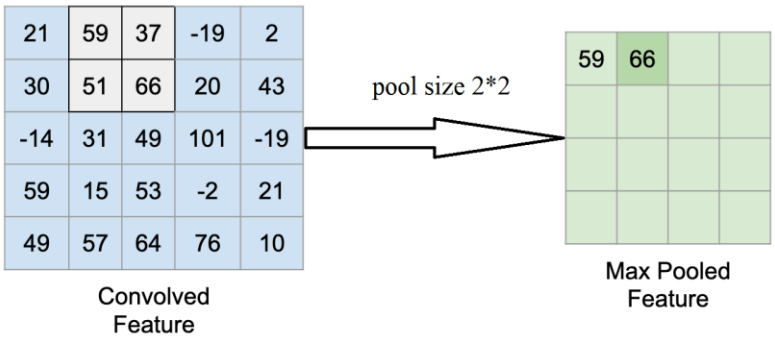


Figure 2.9: Max pooling

9.3.3 ***SoftMax function:***

SoftMax is the activation function in the output layer of the Fully Connected Layer. This mathematical function turns a vector of integers into a vector of probabilities, with the probability of each value proportional to the vector's relative scale.

$$\text{SoftMax(xi)}= \frac{exp^{(Xi)}}{\sum_{j=0} exp^{(xj)}}$$

9.4 Other layers :

9.4.1 Dropout layer

A dropout layer may be seen as a particular type of a noise layer. It assigns the value 0 to a randomly chosen fraction of its input data. The name dropout comes from the action to dropping some units of the input [43]. There are different kinds of dropout layer used in neural networks:

- **Standard dropout:**

randomly selects neurons and blocks their incoming and outgoing connections. The neuron is therefore ignored during forward and backward propagation.

- **Spatial dropout:**

aims to block not only the connections of single neurons but of correlated groups of neurons. For example, if a single value of a 300-dimensional word embedding is dropped, it can be estimated based on the other 299 values. To prevent this, the full embedding vector with its 300 values is dropped at once.

- **Recurrent dropout:**

is a special kind of dropout that is used in recurrent neural network. It affects the updates of recurrent cell states. [44]

9.4.2 Up sample layer:

The Up-sampling layer is a basic, no-weights layer that doubles the dimensions of input and may be utilized in a generative model when followed by a standard convolutional layer.

We employ up-sampling When applied to a sequence of samples of a signal or other continuous function, it generates an estimate of the sequence that would have been produced if the signal had been sampled at a faster rate (or density, as in the case of a photograph).

9.5 U-Net architecture:

For a dense prediction task, Ranneberger [45] proposed an encoder-decoder U-Net framework with long and short skip connections that concatenate feature maps from the corresponding levels in the encoder and decoder.

The U-Net architecture has the ability to combine low-level detail features and high-level semantic representation [46] , this generally helps the network to predict dense segmentation results and reducing the risks of the vanishing gradient problem. Variants of the U-Net have been used in diverse medical image segmentation problems and have demonstrated outstanding performance, four among the top teams, including the top team, exploited the U-Net architecture. [47]

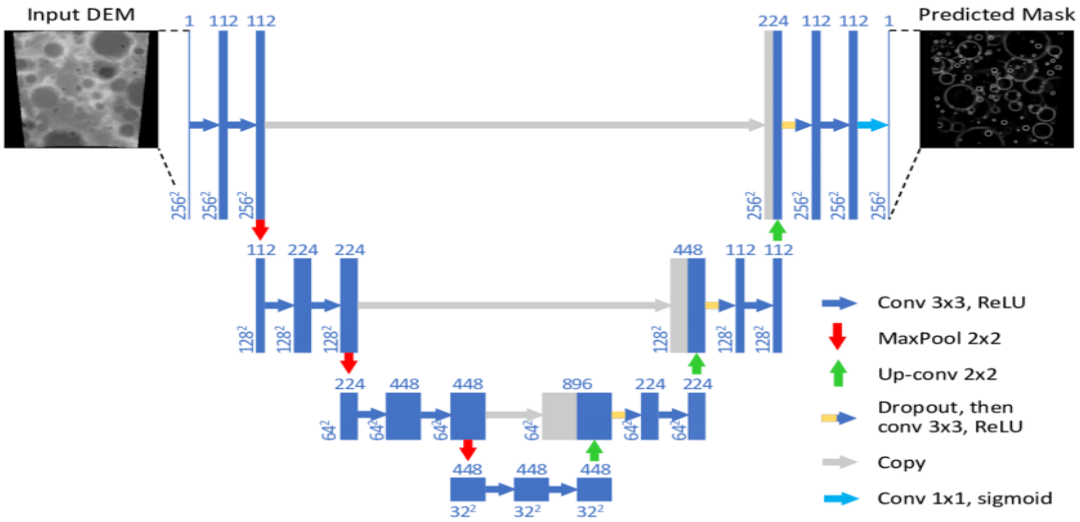


Figure 2.10: Convolutional-neural-network-CNN-architecture-based-on-U-NET

9.6 Autoencoder:

A class of unsupervised deep learning algorithms in which the output shape is the same as the input that allows the network to better learns basic representations.[48].

The main purpose of an auto encoder is to learn and represent (encoding) of data, usually for data dimensionality reduction Compression, fusion, and a variety of other techniques are available.

This auto encoder technique is divided into two components:

9.6.1 ***The encoding phase:***

the input samples are typically mapped in a lower dimensional feature space with a constructive feature representation. This procedure can be repeated until the necessary feature dimensional space is obtained.

9.6.2 ***The decoding phase:***

We regenerate actual features from lower dimensional features with reverse processing.

The conceptual diagram of auto-encoder with encoding and decoding phases is shown in **Figure 2.11.**

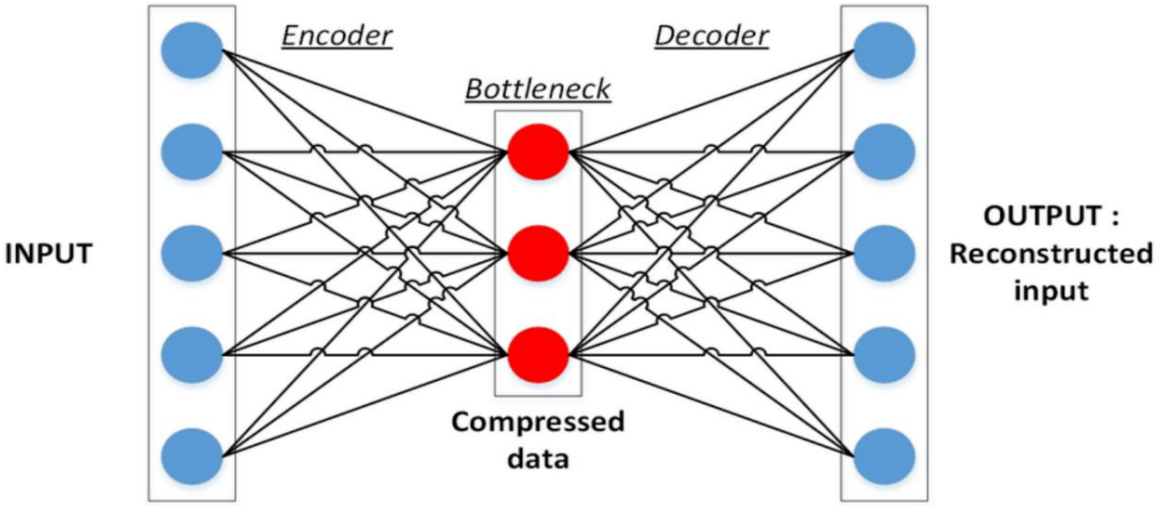


Figure 2.11: Auto encoder architecture

10 Conclusion:

In this second chapter, we are talked about the deep learning approach: definition, history, where and when to use DL and how good it is for image segmentation, Popular deep learning algorithms such as Convolution Neural Networks (CNN) and autoencoder .in order to specify the best method.

Chapter 03

The proposed methodology

1. Introduction:

In this chapter, the proposed methodology is discussed in detail where several methods are used to detect brain tumor and edema. We will also present the data set with description, and the Network Architecture proposed for simple models and merge models; we define the measures of Evaluating the performance of models that we use like Jaccard and dice coefficients.

2. Image Processing Benchmarks:

Benchmarks that compare how well different learning algorithms perform in specific tasks have gained a prominent role in the machine learning community. In recent years, the idea of benchmarking has also gained popularity in the field of medical image analysis .Such benchmarks ,sometimes referred to as “challenges,” all share the common characteristic that different groups optimize their own methods on a training dataset provided by the organizers , and then apply them in a structured way to a common ,independent test dataset .This situation is different from many published comparisons , where one group applies different techniques to a dataset of their choice, which hampers a fair assessment as this group may not be equally knowledgeable about each method and invest more effort in optimizing some algorithms than others. Once benchmarks have been established, their test dataset often becomes a new standard in the field on how to evaluate future progress in the specific image processing task being tested. The annotation and evaluation protocols also may remain the same even when new data are added (to overcome the risk of over-fitting this one particular dataset that may take place after a while), or when related benchmarks are initiated. A key component in benchmarking is an online tool for automatically evaluating segmentations submitted by individual groups, as this allows the labels of the test set never to be made public. This helps ensure that any reported results are not influenced by unintentional overtraining of the method being tested, and that they are therefore truly representative of the method's segmentation performance in practice. [49]

Recent examples of community benchmarks dealing with medical image segmentation and annotation include algorithms for artery centerline extraction vessel segmentation and

stenosis grading, liver segmentation, detection of microaneurysms in digital color fundus photographs, and extraction of airways from CT scans.

Rather few community-wide efforts have focused on segmentation algorithms applied to images of the brain (a current example deals with brain extraction ("masking") although many of the validation frameworks that are used to compare different segments and segmentation algorithms, such as STAPLE, have been developed for applications in brain imaging, or even brain tumor segmentation. [49]

3. Materials and Methods:

The first step in image processing is pre-processing. After which the segmentation of the image is done, image classification using various types of DL algorithms.

3.1 BRATS data set:

Dataset is a collection of instances that has a great significance in research field. Many researchers use different types of 2D & 3D image dataset e.g., open access data and BRATS dataset. [9]

The BRATS dataset is a collection of brain tumor MRI images obtained from many different facilities under standard clinical settings, but with variable equipment and imaging methods, resulting in dramatically varying picture quality reflecting diverse clinical practice across institutions.

The following tumor annotation methodology, on the other hand, was created to allow for identical ground truth delineations across many annotators. [50]

3.2 Description of the Data set:

We have used BRATS-1 dataset that provides multimodal brain MRIs. it's composed of two parts, the first one is the truth brain tumor and edema segmentations such that the area colored in white is the tumor and the area colored in gray is the edema (**Figure 3.6**) annotated by physicians and experts his size is 3192 images. the second part is the training data, consisted of MR scans of 10 LG and 20 HG brain tumor patients, His size is 3192 images. Scans were a mix of pre- and post-therapy images, which were taken over the course of several years using different MR scanners with different scanning protocols (eg 2D or 3D). Each subject in the dataset was scanned in four modalities: T1, T1c, T2, and FLAIR, in every single one there is 176 MRI slices (as demonstrated in **Figure.3.2,3.3,3.4,3.5**). In order to annotate these images, three different types of cells structures were defined: “edema,” “tumor” and “healthy tissue” (**Figure.3.1**). An annotation protocol was used by expert raters to manually annotate each case. Then the segmentation results from all raters were fused to obtain a single consensus segmentation for each subject. This segmentation label is considered as the ground truth for the dataset. The BRATS organizers performed an experiment for Inter-Rater variability of manual segmentations. The present study uses this dataset for performance evaluation of segmentation.

Table 3.1 and table 3.2 enlists the detailed scan image properties of the sequences used.

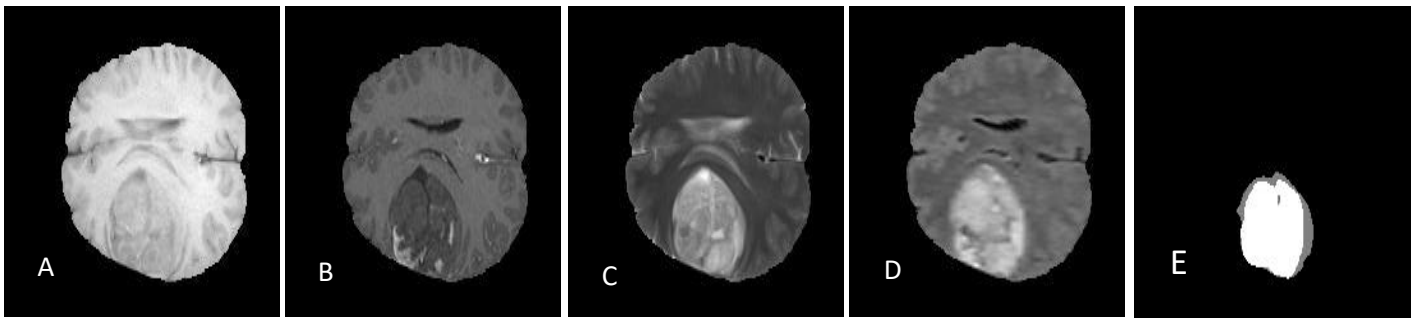


Figure 3. 1: Tumor appearance on two imaging modalities (B = T1c, C = T2) with manual annotations, edema region appears in D (D = flair), healthy tissues in A (A= T1) and finally (E=Truth)

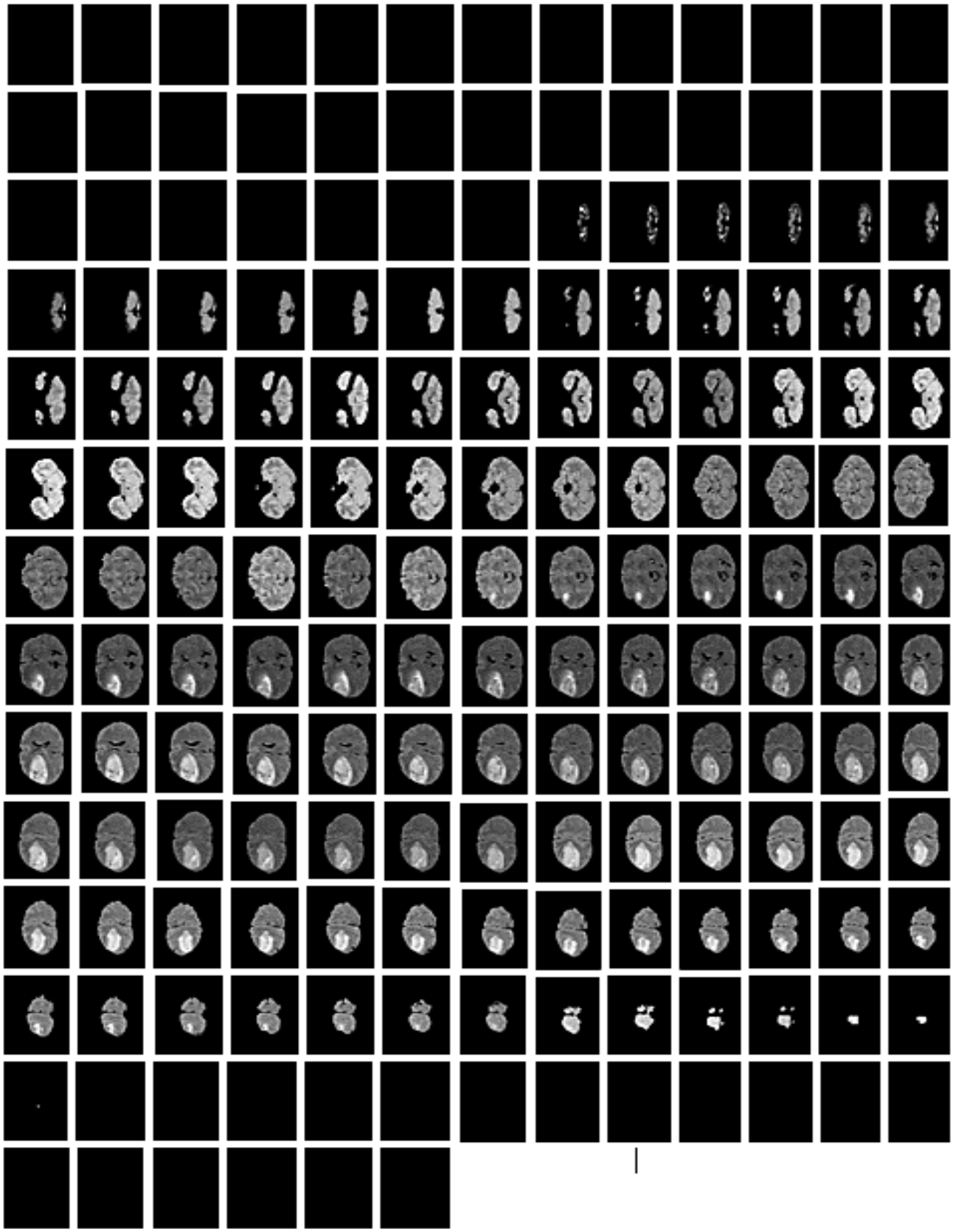


Figure 3.2: 3D Ground training views of 176 MRI slices of the flair modality of patient HG0004 obtained from the BRATS-1 dataset.

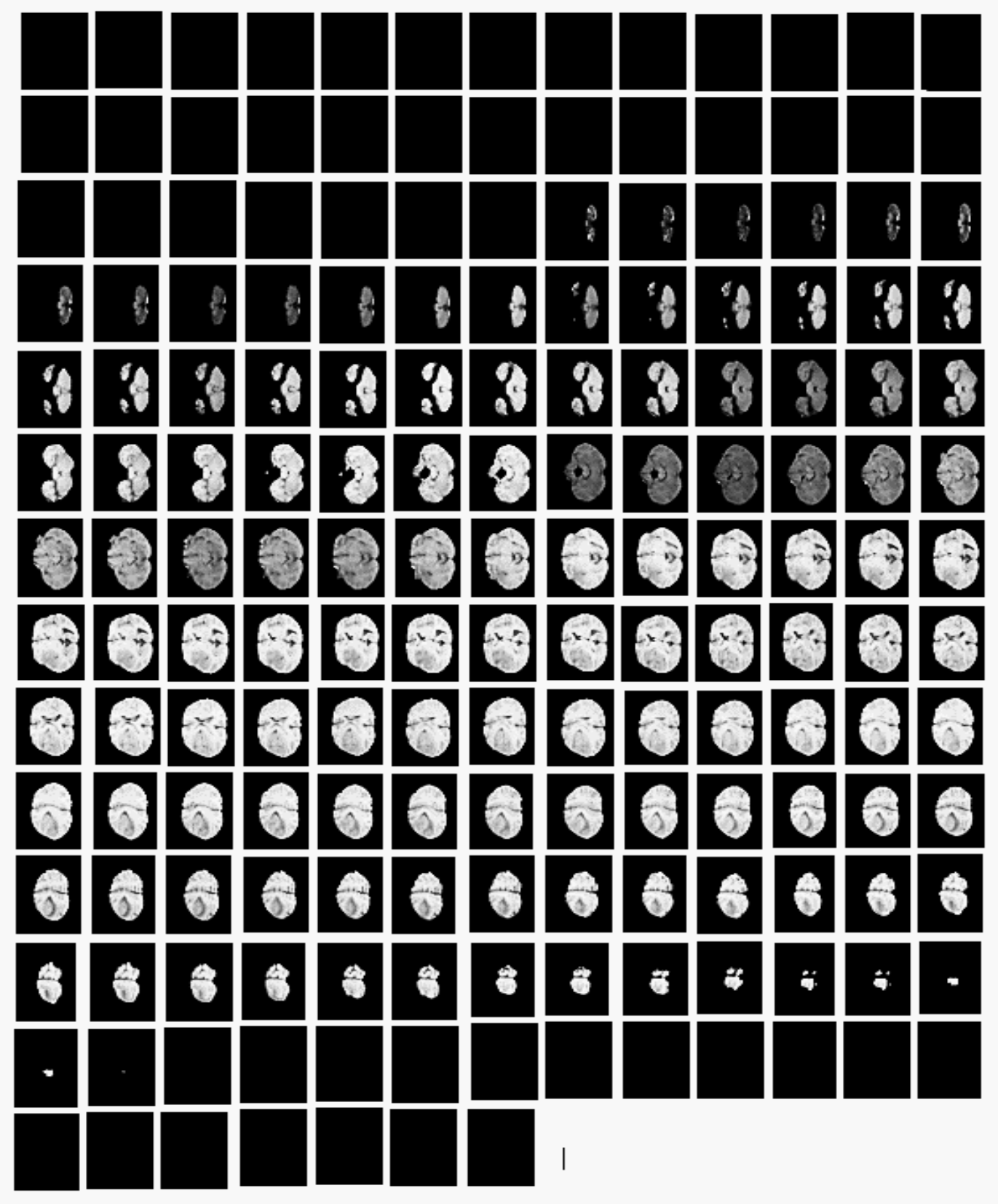


Figure 3.3:3D Ground training views of 176 MRI slices of the T1 modality of patient HG0004 obtained from the BRATS-1 dataset.

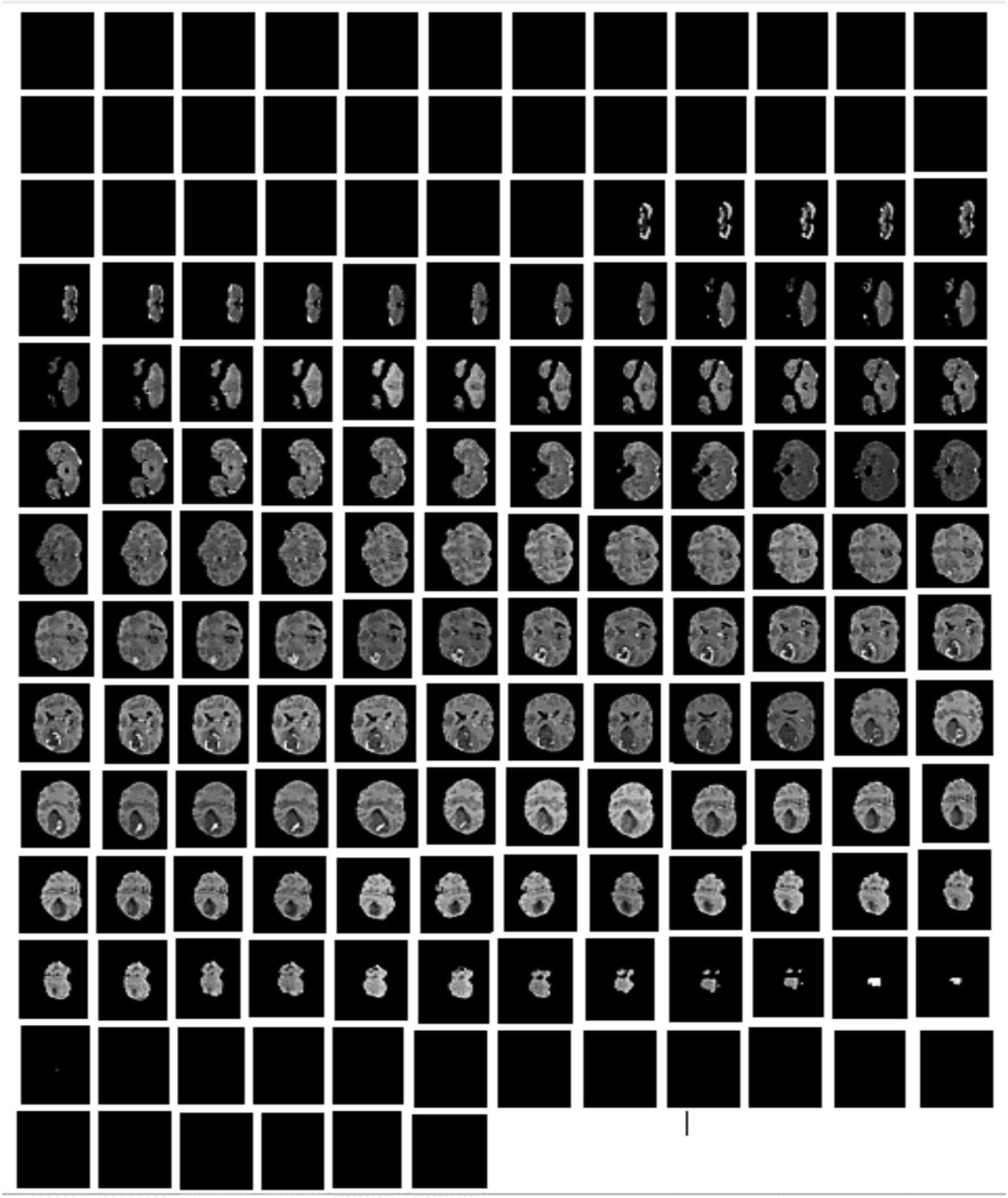


Figure 3.4:3D Ground training views of 176 MRI slices of the T1C modality of patient HG0004 obtained from the BRATS-1 dataset.

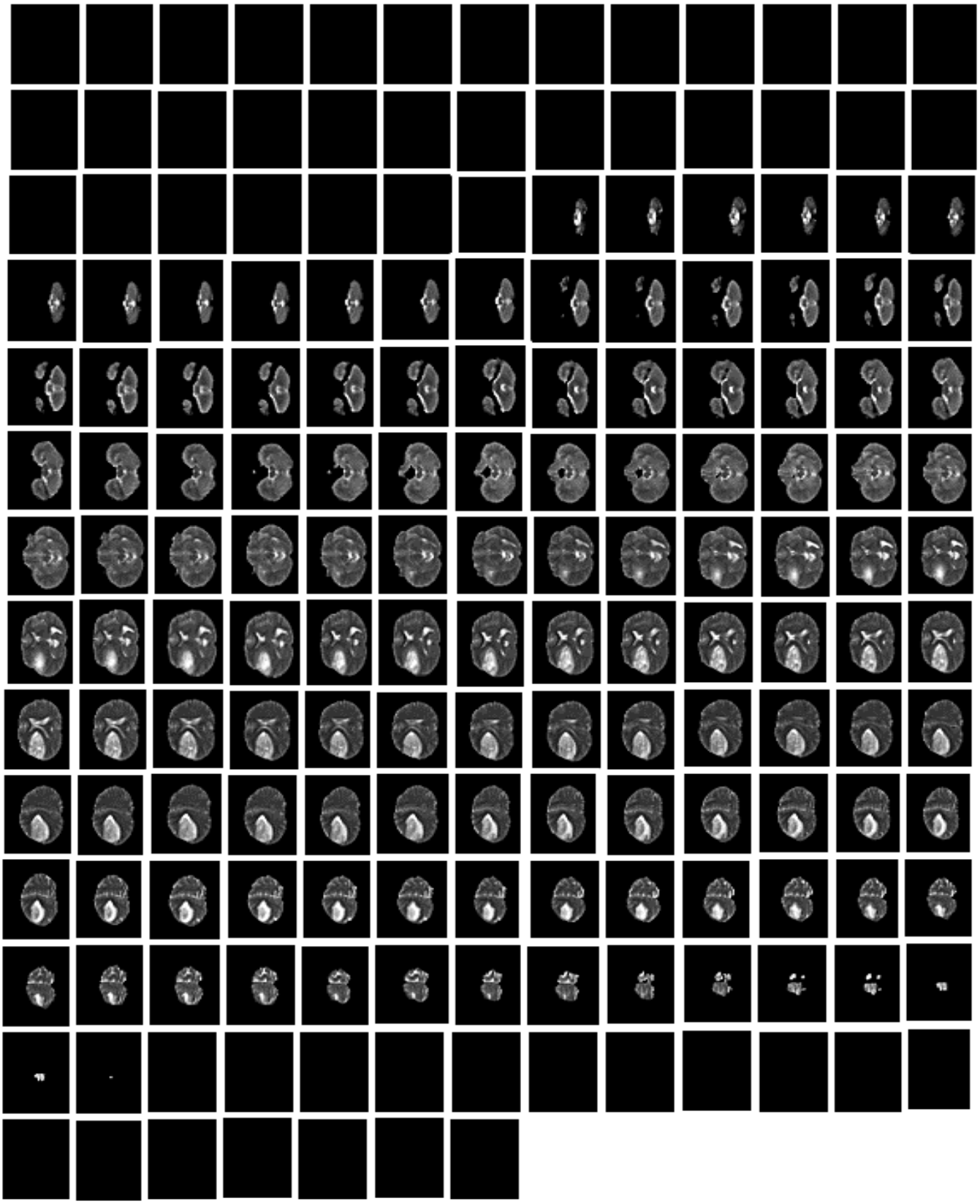


Figure 3.5: 3D Ground training views of 176 MRI slices of the T2 modality of patient HG0004 obtained from the BRATS-1 dataset.

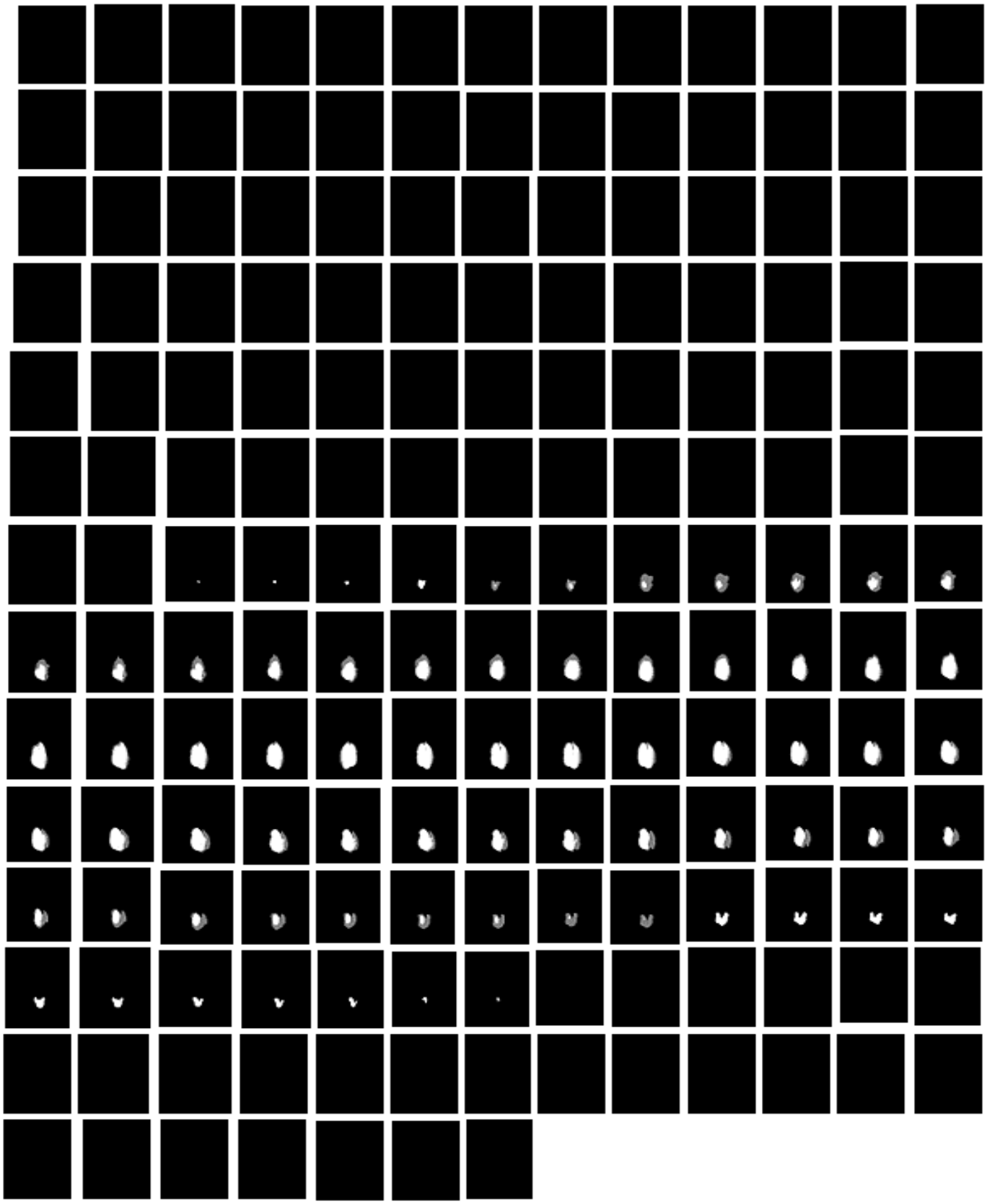


Figure 3. 6: 3D ground truths views of 176 MRI slice of the patient's brats HG0004 obtained from BRATS-1 dataset

Table 3. 1: Properties of BRATS-1 dataset for high-grade.

Patient No.	No. of slices per patient	No. of normal slices	No. of tumor slices	No. of edema Slices	Width × Height (Pixel)
BRATS_HG0001	176	102	74	58	160×216
BRATS_HG0002		111	65	47	
BRATS_HG0003		91	85	33	176×16
BRATS_HG0004		106	74	55	160×216
BRATS_HG0005		112	64	55	
BRATS_HG0006		102	85	55	216×236
BRATS_HG0007		116	60	43	176 × 216
BRATS_HG0008		97	79	45	160 × 216
BRATS_HG0009		103	73	55	
BRATS_HG0010		135	41	35	
BRATS_HG0011		108	78	54	176 × 216
BRATS_HG0012		132	44	33	
BRATS_HG0013		145	31	22	
BRATS_HG0014		102	74	60	
BRATS_HG0015		96	80	47	160 × 216
BRATS_HG0022	165	108	57	41	230 × 230
BRATS_HG0024	240	181	60	41	168 × 240
BRATS_HG0025	230	158	72	58	162 × 230
BRATS_HG0026	220	139	81	58	168 × 220
BRATS_HG0027	230	173	67	58	162 × 230

Table 3. 2: Properties of BRATS-1 dataset for Low-grade.

Patient No.	No. of slices per patient	No. of normal slices	No. of tumor slices	No. of edema Slices	Width × Height (Pixel)
BRATS_LG0001	176	135	41	37	216 ×196
BRATS_LG0002		101	75	58	216 × 236
BRATS_LG0004		121	55	22	
BRATS_LG0006	196	157	39	19	159 × 216
BRATS_LG0008	176	144	31	30	216 × 236
BRATS_LG0011	230	180	49	49	162 × 230
BRATS_LG0012	165	123	42	32	230 × 230
BRATS_LG0013	220	158	52	45	168 × 220
BRATS_LG0014	230	193	27	22	162 × 230
BRATS_LG0015	163	122	41	36	230 × 230

3.3 Loading and Reading the Data:

To load and read data we used google. Collab library, got the AUTH Token for the desktop application for modifying, reading and adding data.

Reading the data need to be accurate, reading the images needs to be in a precise pattern and not randomly, every image needs to have its own mask in the truths grounds, so at first we use the glob library to read and match a certain string that we concatenate to the corresponding paths this path need to read a BRATS's folder from first image to the last image and so on then move to another folder while also reading the truth images from the truths folder, we noticed that reading the data takes while the first time it executes so we decided to after reading each 3D image any type we put it in a list, we save it in a file with extension .pkl with library pickle the name of each file contain information about the data that it possess and saving the ground truths list in another file containing the same information called masks, which brain and what type of images does it contain, we keep in notice again that each slice in this list needs to have the exact truth mask for it at the same index. After saving the data, we upload any number of brains with a loop and.

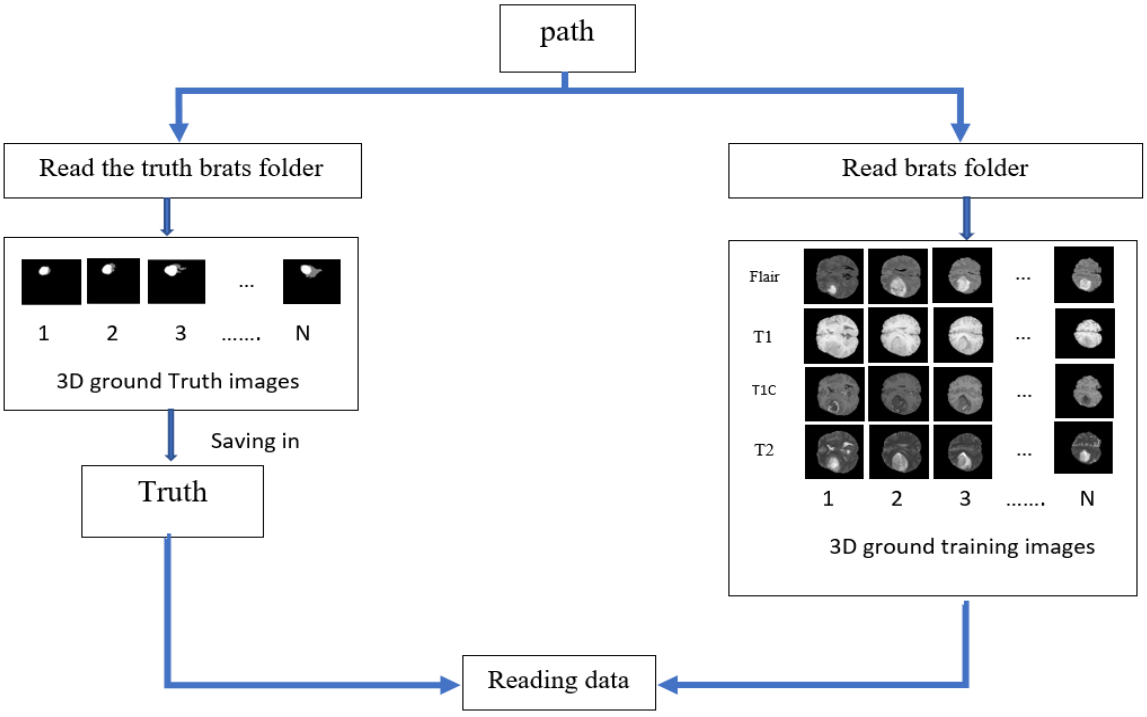


Figure 3.7: diagram that clarify how to read and load the data.

3.4 Data Pre-processing:

Our data is consisted of 3D image containing from bottom to the top tip of the brain , at first our images dimension varies from a 3D image to another, so the first thing we did was to crop the images at the same size so they can be fit to the U-net , loaded and cropped with cv2.Then after before feeding to the model we decide the size of the input, we tested a variant of dimensions to feed them to U-net (144,160) then (32,32), as it was said to be the best given dice coefficient and Jaccard distance but results weren't much satisfying, so we made an experimental size (64,64), right after loading the data , we crop many small pieces from our data would be 8 pictures of (64,64) we used an algorithm that move the cropping process a sliding window of (32x32) was applied to have more diverse data .Cropping the images into small pieces actually worked in our favor as our training data has 40826 pictures per each type of images and unbalanced a lot, cropping the images into sub images made sure we have 8 times the number more data to feed, and we can balance it with a small algorithm , we selected the sub images with only tumor or edema , we added other pictures that contains no tumor and edema in the ground truths images nor the data , last our test model couldn't understand the relation in dark images with masks containing fully dark image, so we fetched more picture pictures that contain a dark training data with a dark ground truths mask, this way we balanced the data to feed it to U-net .

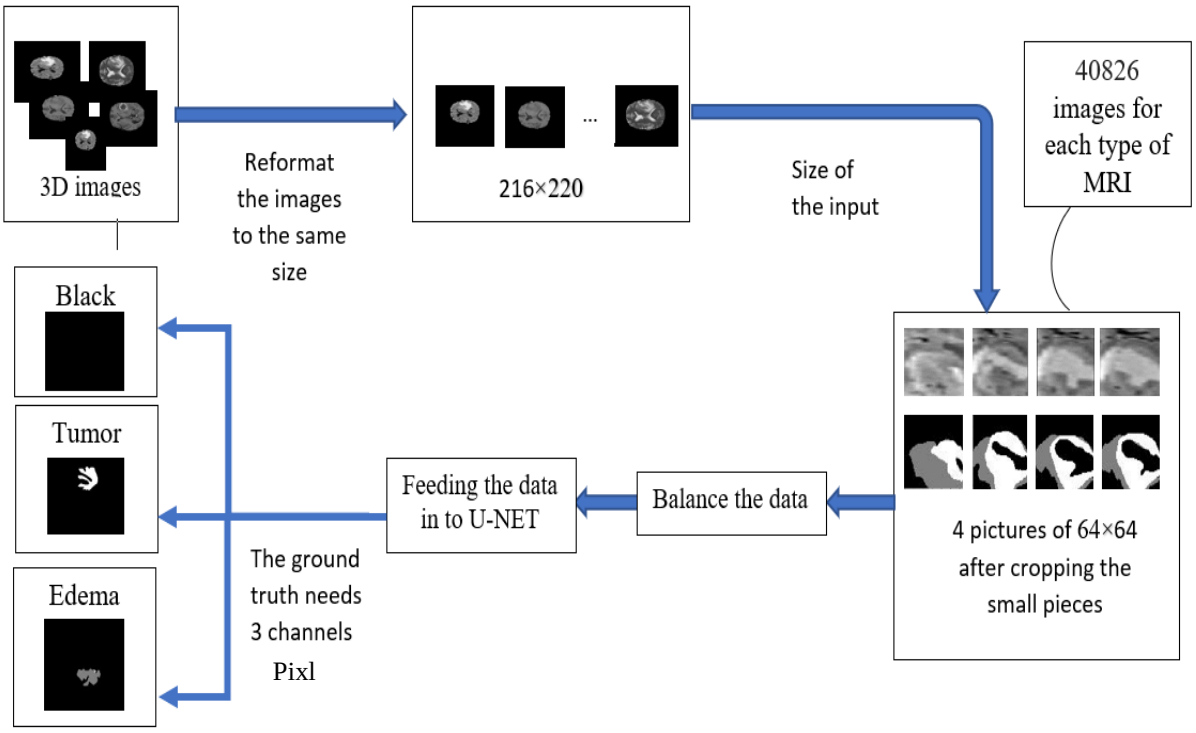


Figure 3.8: The different steps of data pre-processing

4. ***The proposed approach:***

The CNN architecture utilized for segmenting pictures of the corneal endothelium. It was created using the U-Net network. [45]

The U-Net uses an encoder-decoder design, in which the encoder decreases the spatial dimension of the input data while the decoder progressively restores it. As a result, instead of categorizing an input image as a whole, it generates a pixel-by-pixel probability map. In contrast to conventional CNN designs, the U-Net does not require a large number of training samples and may be trained well with only a few pictures. This was also true for the dataset used in this research. [51]

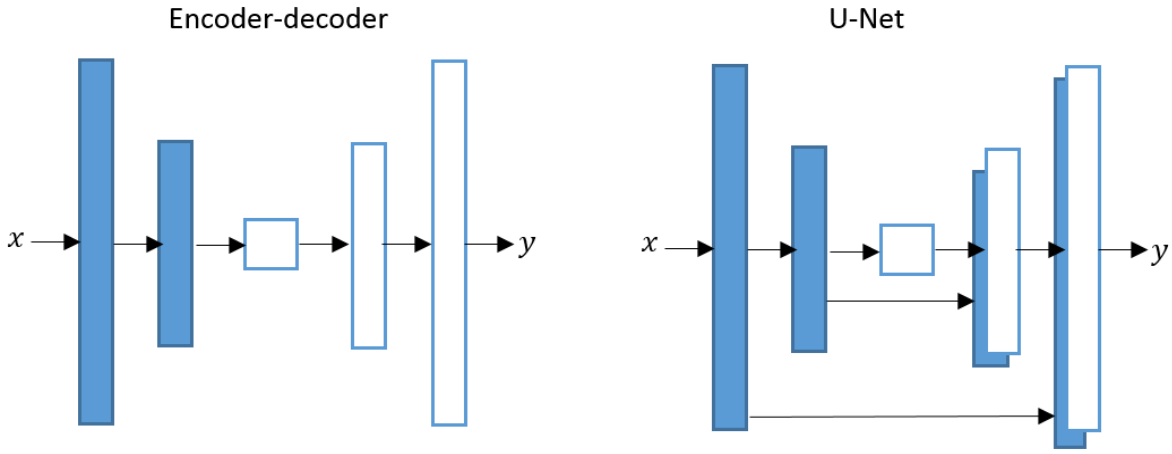


Figure 3.9: encoder-decoder design of the U-NET approach.

4.1 Network Architecture:

4.1.1 The proposed CNN network model:

For this segmentation work we've decided to work with convolutional neural network, combined with autoencoder using an encoder path and a decoder path.

The encoder path uses multiple layers of convolutional layers followed by RELU activation function each, along with some dropout layers to avoid over-fitting, there are 2x2 max pooling layer in the encoder path which picks the highest value in portion of the image which lead with the extraction of a feature map, along doubling channels after each maxpooling with a convolutional layer.

In the decoder path the instead of using maxpooling to downsample the dimension, here the up sampling happens, doing the opposite here where channels are divided by their half number after that's merged from different blocks of U-net , in fourth layer , third and second layer are merged , in the fifth layer the first and fourth block output are merged then again some convolutional layer with dropout followed by another convolutional layer with RELU and in the end a SoftMax after the last convolutional layer. **Figure 3.10** depicts layers of the model.

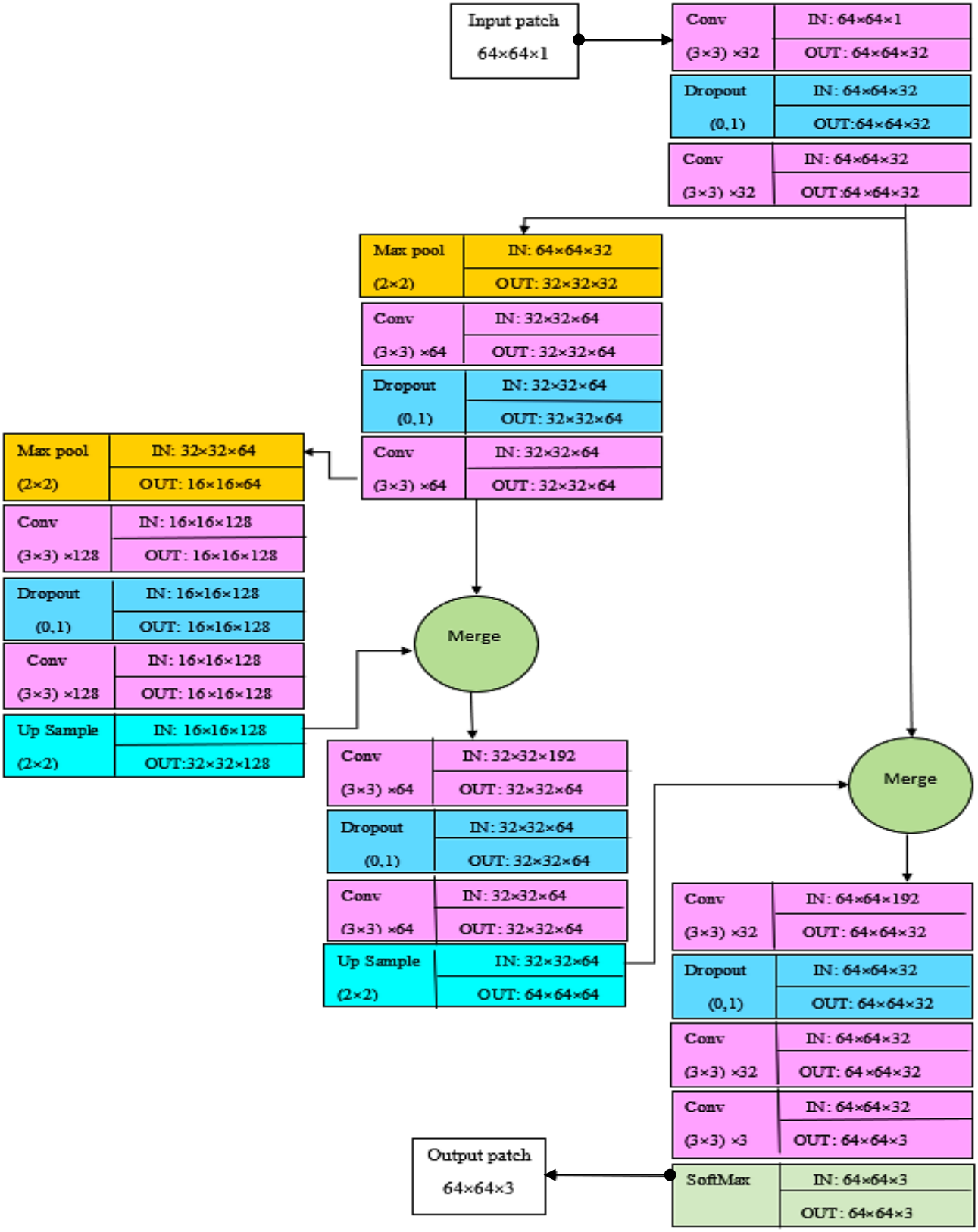


Figure 3.10: The architecture of the proposed CNN network for models without merge.

Table 3. 3: No merge Model: "U-NET" summary.

Layer (type)	*Output Shape*	*Param#*	*Connected to*
input_2 (Input Layer)	[(None, 64, 64, 1)]	0	
input_3 (Input Layer)	[(None, 64, 64, 1)]	0	
conv2d_11 (Conv2D)	(None, 64, 64, 32)	320	input_2[0][0]
conv2d_19 (Conv2D)	(None, 64, 64, 32)	320	input_3[0][0]
dropout_5 (Dropout)	(None, 64, 64, 32)	0	conv2d_11[0][0]
dropout_9 (Dropout)	(None, 64, 64, 32)	0	conv2d_19[0][0]
conv2d_12 (Conv2D)	(None, 64, 64, 32)	9248	dropout_5[0][0]
conv2d_20 (Conv2D)	(None, 64, 64, 32)	9248	dropout_9[0][0]
re_lu_5 (ReLU)	(None, 64, 64, 32)	0	conv2d_12[0][0]
re_lu_9 (ReLU)	(None, 64, 64, 32)	0	conv2d_20[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 32)	0	re_lu_5[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 32, 32, 32)	0	re_lu_9[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 64)	18496	max_pooling2d_2[0][0]
conv2d_21 (Conv2D)	(None, 32, 32, 64)	18496	max_pooling2d_4[0][0]
dropout_6 (Dropout)	(None, 32, 32, 64)	0	conv2d_13[0][0]
dropout_10 (Dropout)	(None, 32, 32, 64)	0	conv2d_21[0][0]
conv2d_14 (Conv2D)	(None, 32, 32, 64)	36928	dropout_6[0][0]
conv2d_22 (Conv2D)	(None, 32, 32, 64)	36928	dropout_10[0][0]
re_lu_6 (ReLU)	(None, 32, 32, 64)	0	conv2d_14[0][0]
re_lu_10 (ReLU)	(None, 32, 32, 64)	0	conv2d_22[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 64)	0	re_lu_6[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 16, 16, 64)	0	re_lu_10[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 128)	73856	max_pooling2d_3[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 128)	73856	max_pooling2d_5[0][0]
dropout_7 (Dropout)	(None, 16, 16, 128)	0	conv2d_15[0][0]
dropout_11 (Dropout)	(None, 16, 16, 128)	0	conv2d_23[0][0]
conv2d_16 (Conv2D)	(None, 16, 16, 128)	147584	dropout_7[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 128)	147584	dropout_11[0][0]
re_lu_7 (ReLU)	(None, 16, 16, 128)	0	conv2d_16[0][0]
re_lu_11 (ReLU)	(None, 16, 16, 128)	0	conv2d_24[0][0]
up_sampling2d_2 (UpSampling2D)	(None, 32, 32, 128)	0	re_lu_7[0][0]
up_sampling2d_4 (UpSampling2D)	(None, 32, 32, 128)	0	re_lu_11[0][0]
concatenate_2 (Concatenate)	(None, 32, 32, 192)	0	re_lu_6[0][0] up_sampling2d_2[0][0]

concatenate_3 (Concatenate)	(None, 32, 32, 192)	0	re_lu_10[0][0] up_sampling2d_4[0][0]
conv2d_17 (Conv2D)	(None, 32, 32, 64)	110656	concatenate_2[0][0]
conv2d_25 (Conv2D)	(None, 32, 32, 64)	110656	concatenate_3[0][0]
dropout_8 (Dropout)	(None, 32, 32, 64)	0	conv2d_17[0][0]
dropout_12 (Dropout)	(None, 32, 32, 64)	0	conv2d_25[0][0]
conv2d_18 (Conv2D)	(None, 32, 32, 64)	36928	dropout_8[0][0]
conv2d_26 (Conv2D)	(None, 32, 32, 64)	36928	dropout_12[0][0]
re_lu_8 (ReLU)	(None, 32, 32, 64)	0	conv2d_18[0][0]
re_lu_12 (ReLU)	(None, 32, 32, 64)	0	conv2d_26[0][0]
up_sampling2d_3 (UpSampling2D)	(None, 64, 64, 64)	0	re_lu_8[0][0]
up_sampling2d_5 (UpSampling2D)	(None, 64, 64, 64)	0	re_lu_12[0][0]
concatenate_4 (Concatenate)	(None, 64, 64, 192)	0	re_lu_5[0][0] up_sampling2d_3[0][0] re_lu_9[0][0] up_sampling2d_5[0][0]
conv2d_27 (Conv2D)	(None, 64, 64, 32)	55328	concatenate_4[0][0]
dropout_13 (Dropout)	(None, 64, 64, 32)	0	conv2d_27[0][0]
conv2d_28 (Conv2D)	(None, 64, 64, 32)	9248	dropout_13[0][0]
re_lu_13 (ReLU)	(None, 64, 64, 32)	0	conv2d_28[0][0]
conv2d_29 (Conv2D)	(None, 64, 64, 3)	99	re_lu_13[0][0]
softmax_1 (Softmax)	(None, 64, 64, 3)	0	conv2d_29[0][0]
Total params: 932,707 **Trainable params: 932,707** **Non-trainable params: 0**			

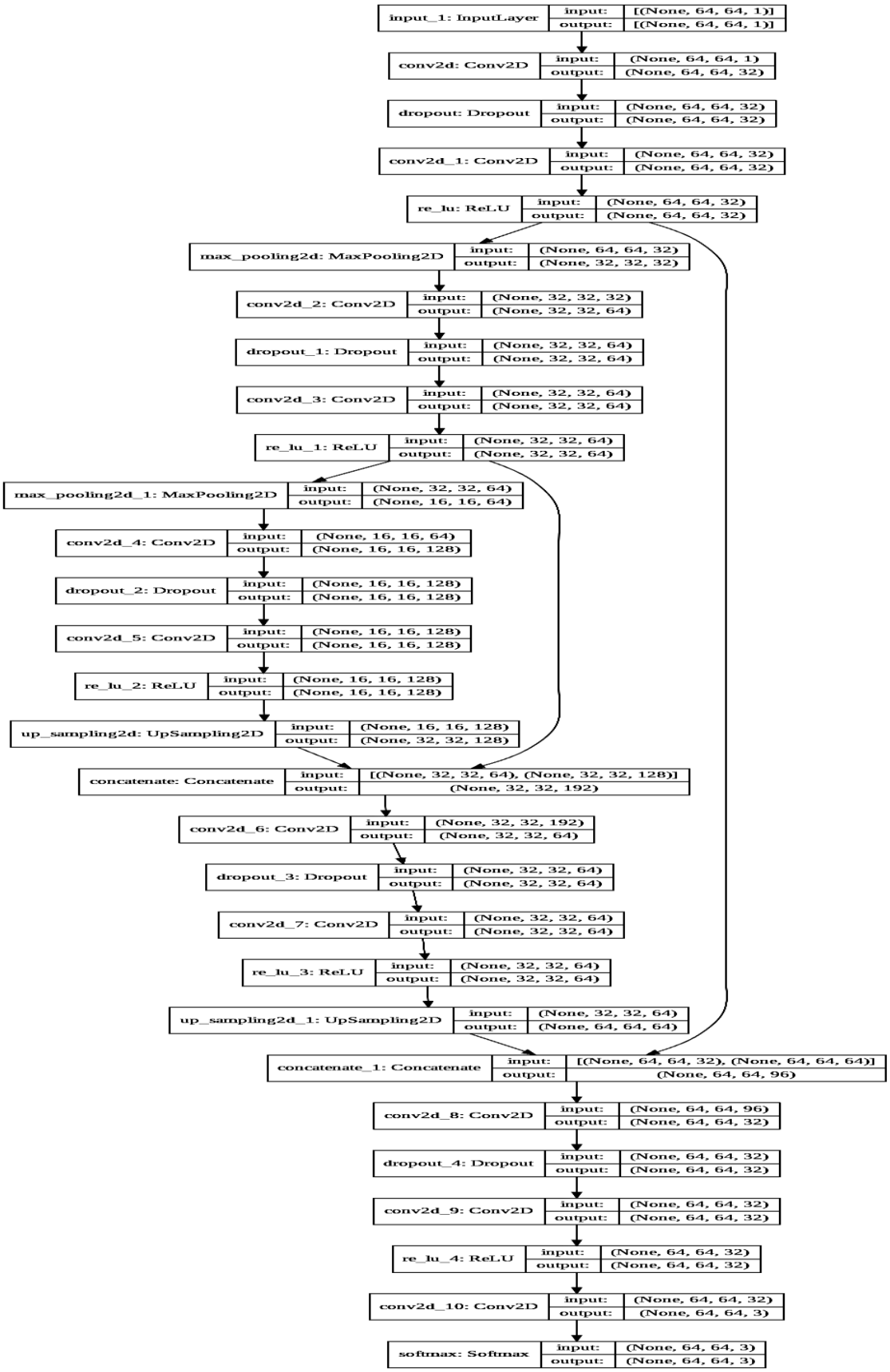


Figure 3. 11:plot model of no merge U-NET architecture

4.1.2 ***The Merge models:***

Merge models are non-other than U-net model built with double the inputs and double blocks till block 4 , merge models take two different kinds of images , example T1C, FLAIR or T1 ,T2 and so on these images are fed separately in merge models and they concatenate as output in 5th block input , as shown in the **figure 3.12** , this cloned sub model has the same layers as well as the same activation function and channels, convolutional layers followed by dropouts and maxpooling and up sampling for contracting our data.

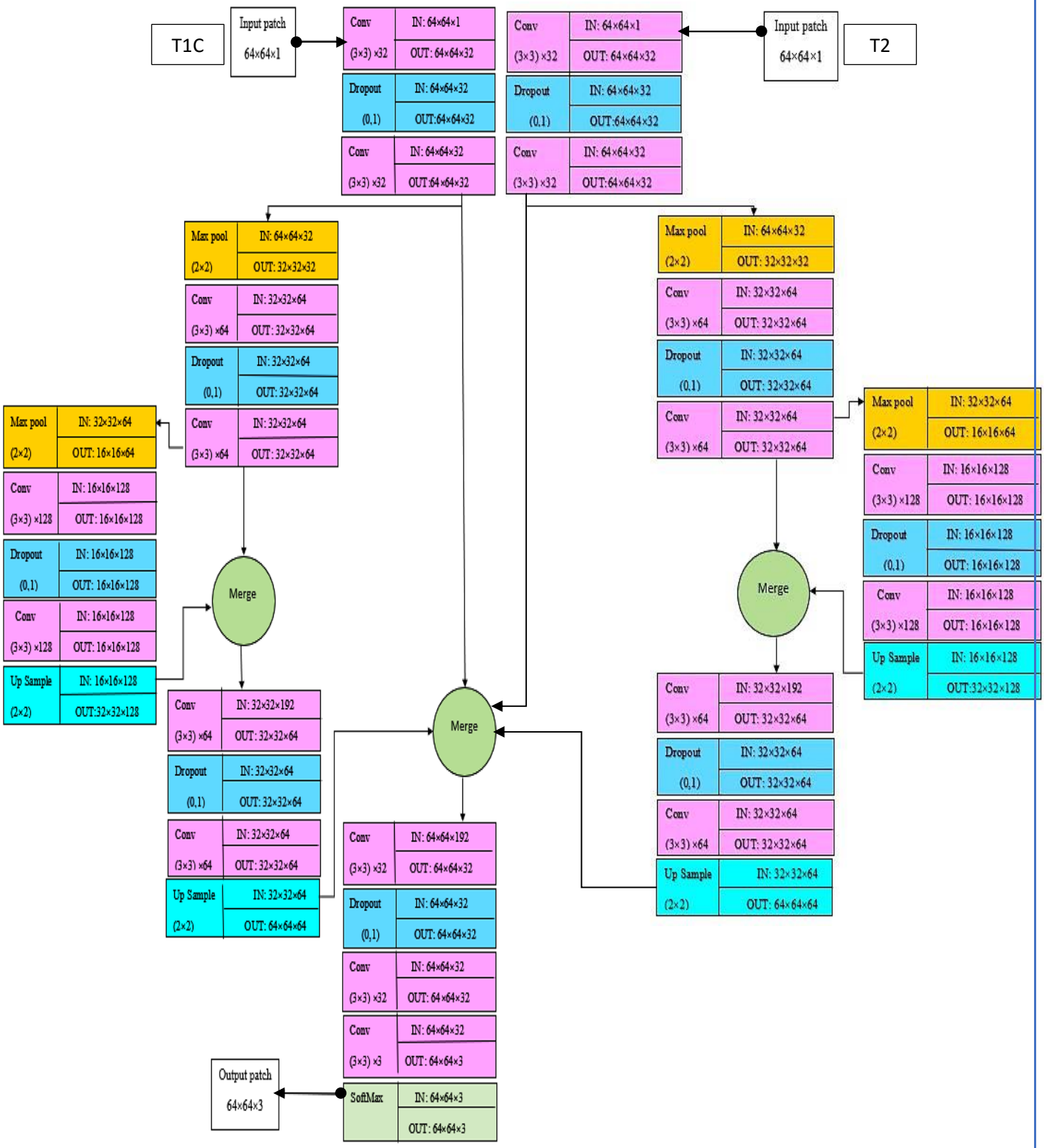


Figure 3. 12: The architecture of the proposed CNN network for merge models.

Table 3. 4: Merge model "U-NET" summary.

Layer (type)	*Output Shape*	*Param#*	*Connected to*
input_4 (InputLayer)	[(None, 64, 64, 1)]	0	
input_5 (InputLayer)	[(None, 64, 64, 1)]	0	
conv2d_30 (Conv2D)	(None, 64, 64, 32)	320	input_4[0][0]
conv2d_38 (Conv2D)	(None, 64, 64, 32)	320	input_5[0][0]
dropout_14 (Dropout)	(None, 64, 64, 32)	0	conv2d_30[0][0]
dropout_18 (Dropout)	(None, 64, 64, 32)	0	conv2d_38[0][0]
conv2d_31 (Conv2D)	(None, 64, 64, 32)	9248	dropout_14[0][0]
conv2d_39 (Conv2D)	(None, 64, 64, 32)	9248	dropout_18[0][0]
re_lu_14 (ReLU)	(None, 64, 64, 32)	0	conv2d_31[0][0]
re_lu_18 (ReLU)	(None, 64, 64, 32)	0	conv2d_39[0][0]
max_pooling2d_6 (MaxPooling2D)	(None, 32, 32, 32)	0	re_lu_14[0][0]
max_pooling2d_8 (MaxPooling2D)	(None, 32, 32, 32)	0	re_lu_18[0][0]
conv2d_32 (Conv2D)	(None, 32, 32, 64)	18496	max_pooling2d_6[0][0]
conv2d_40 (Conv2D)	(None, 32, 32, 64)	18496	max_pooling2d_8[0][0]
dropout_15 (Dropout)	(None, 32, 32, 64)	0	conv2d_32[0][0]
dropout_19 (Dropout)	(None, 32, 32, 64)	0	conv2d_40[0][0]
conv2d_33 (Conv2D)	(None, 32, 32, 64)	36928	dropout_15[0][0]
conv2d_41 (Conv2D)	(None, 32, 32, 64)	36928	dropout_19[0][0]
re_lu_15 (ReLU)	(None, 32, 32, 64)	0	conv2d_33[0][0]
re_lu_19 (ReLU)	(None, 32, 32, 64)	0	conv2d_41[0][0]
max_pooling2d_7 (MaxPooling2D)	(None, 16, 16, 64)	0	re_lu_15[0][0]
max_pooling2d_9 (MaxPooling2D)	(None, 16, 16, 64)	0	re_lu_19[0][0]
conv2d_34 (Conv2D)	(None, 16, 16, 128)	73856	max_pooling2d_7[0][0]
conv2d_42 (Conv2D)	(None, 16, 16, 128)	73856	max_pooling2d_9[0][0]
conv2d_42 (Conv2D)	(None, 16, 16, 128)	73856	max_pooling2d_9[0][0]
dropout_16 (Dropout)	(None, 16, 16, 128)	0	conv2d_34[0][0]
dropout_20 (Dropout)	(None, 16, 16, 128)	0	conv2d_42[0][0]
conv2d_35 (Conv2D)	(None, 16, 16, 128)	147584	dropout_16[0][0]
conv2d_43 (Conv2D)	(None, 16, 16, 128)	147584	dropout_20[0][0]
re_lu_16 (ReLU)	(None, 16, 16, 128)	0	conv2d_35[0][0]
re_lu_20 (ReLU)	(None, 16, 16, 128)	0	conv2d_43[0][0]
up_sampling2d_6 (UpSampling2D)	(None, 32, 32, 128)	0	re_lu_16[0][0]
up_sampling2d_8 (UpSampling2D)	(None, 32, 32, 128)	0	re_lu_20[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 192)	0	re_lu_15[0][0] up_sampling2d_6[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 192)	0	re_lu_19[0][0] up_sampling2d_8[0][0]
conv2d_36 (Conv2D)	(None, 32, 32, 64)	110656	concatenate_5[0][0]
conv2d_44 (Conv2D)	(None, 32, 32, 64)	110656	concatenate_6[0][0]
dropout_17 (Dropout)	(None, 32, 32, 64)	0	conv2d_36[0][0]
dropout_21 (Dropout)	(None, 32, 32, 64)	0	conv2d_44[0][0]
conv2d_37 (Conv2D)	(None, 32, 32, 64)	36928	dropout_17[0][0]
conv2d_45 (Conv2D)	(None, 32, 32, 64)	36928	dropout_21[0][0]
re_lu_17 (ReLU)	(None, 32, 32, 64)	0	conv2d_37[0][0]

re_lu_21 (ReLU)	(None, 32, 32, 64)	0	conv2d_45[0][0]
up_sampling2d_7 (UpSampling2D)	(None, 64, 64, 64)	0	re_lu_17[0][0]
up_sampling2d_9 (UpSampling2D)	(None, 64, 64, 64)	0	re_lu_21[0][0]
concatenate_7 (Concatenate)	(None, 64, 64, 192)	0	re_lu_14[0][0] up_sampling2d_7[0][0] re_lu_18[0][0] up_sampling2d_9[0][0]
conv2d_46 (Conv2D)	(None, 64, 64, 32)	55328	concatenate_7[0][0]
dropout_22 (Dropout)	(None, 64, 64, 32)	0	conv2d_46[0][0]
conv2d_47 (Conv2D)	(None, 64, 64, 32)	9248	dropout_22[0][0]
re_lu_22 (ReLU)	(None, 64, 64, 32)	0	conv2d_47[0][0]
conv2d_48 (Conv2D)	(None, 64, 64, 3)	99	re_lu_22[0][0]
softmax_2 (Softmax)	(None, 64, 64, 3)	0	conv2d_48[0][0]

Total params: 932,707
Trainable params: 932,707
Non-trainable params: 0

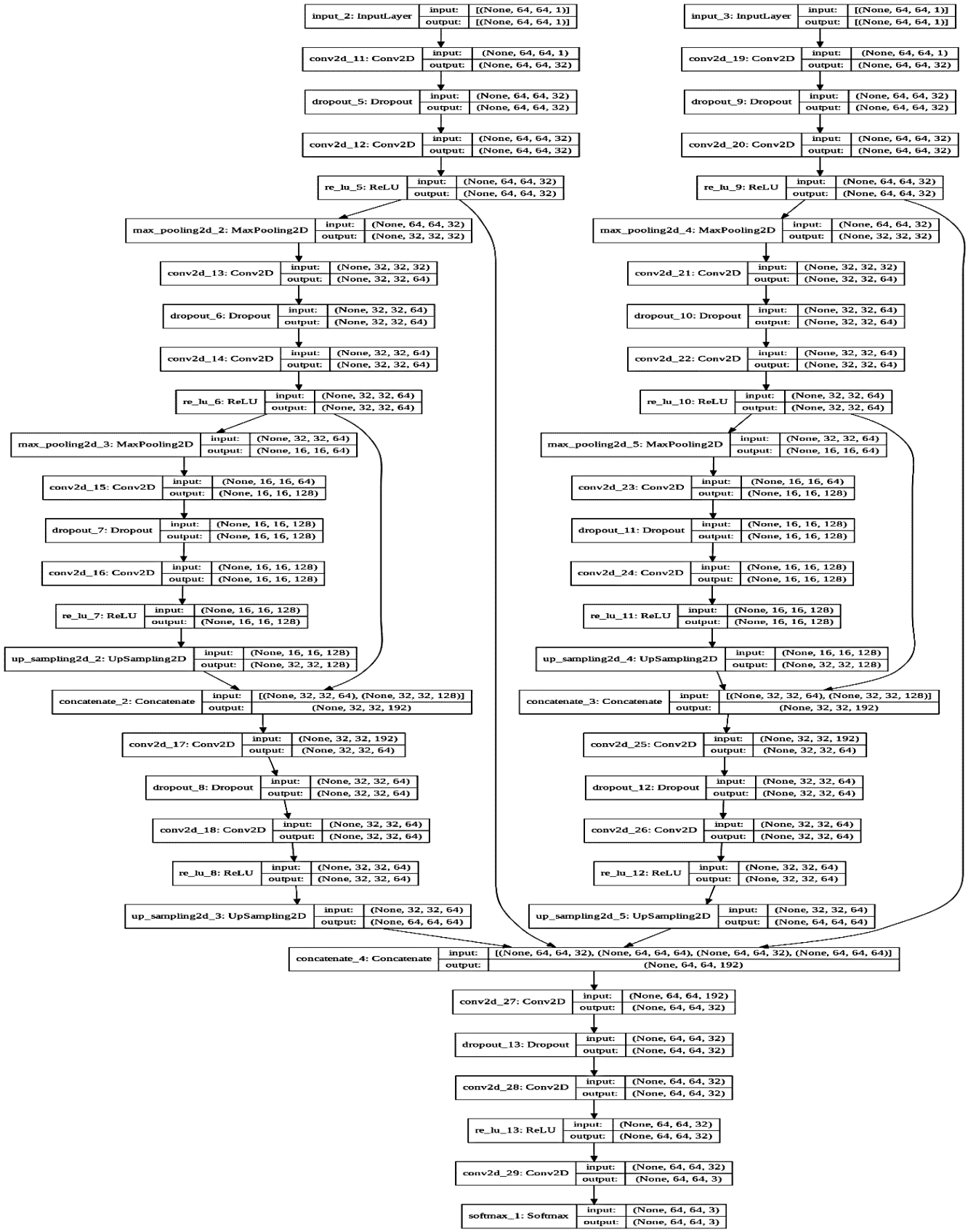


Figure 3.13: plot model of merge U-NET architecture

4.2 Training:

U-net was trained on patch size of 64x64 slices, these images provided the best dice and Jaccard index, as it was selected experimentally from various tests, 124x124, 32x32, 64x64. These patches were extracted from our training images using and walking algorithm that extract slices of 64x64 by moving every time with 32 pixels in x and y axis, 40826 images were extracted from images for each type of 3D image Sample images are shown in the figure as well as the ground truths.

The network was trained on 100 epochs with a batch size of 32 using the categorical loss entropy, Adam optimizer was used to minimize objective, with learning rate of 0.001 and decay of 1-e^-6).

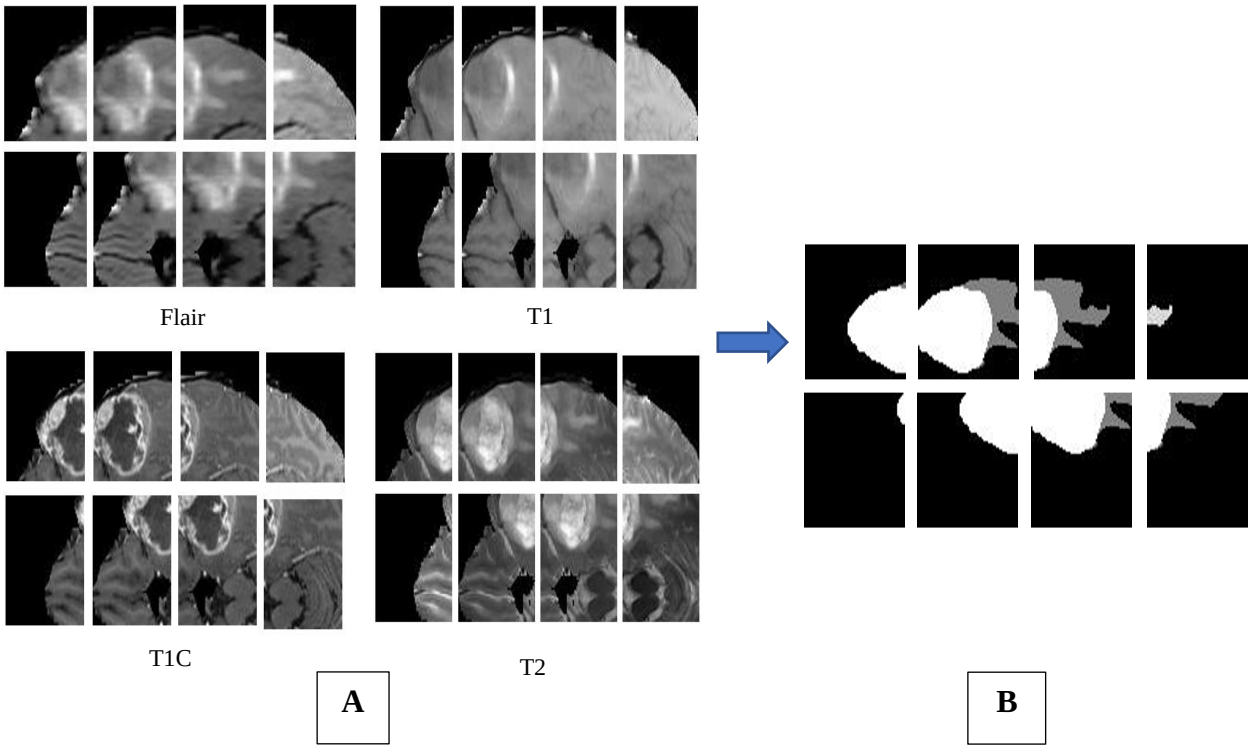


***Figure* 3. 14:** The sample training patches of brain tumor and edema from MRI images (a) and the corresponding ground truths (b).

4.3 Prediction:

We load each data, do the same process as preprocessing then we feed it and see our predictions by plotting the images alongside the ground truths to compare it, we also use evaluate method and preprocess the ground truths to turn them into 3 channels and see the dice coefficient and Jaccard distance.

5. Understanding Boxplots

A boxplot is a graph that gives us a good indication of how the values in the data are spread out. Although boxplots may seem primitive in comparison to a histogram or density plot, they have the advantage of taking up less space, which is useful when comparing distributions between many groups or datasets. It can tell us about outliers and what their values are. It can also tell us if the data is symmetrical, how tightly data is grouped, and if the data is skewed. [52]

Boxplots are a standardized way of displaying the distribution of data based on a five-number summary: Median, Inter-quartile range, Upper quartile, Lower quartile, Whiskers.

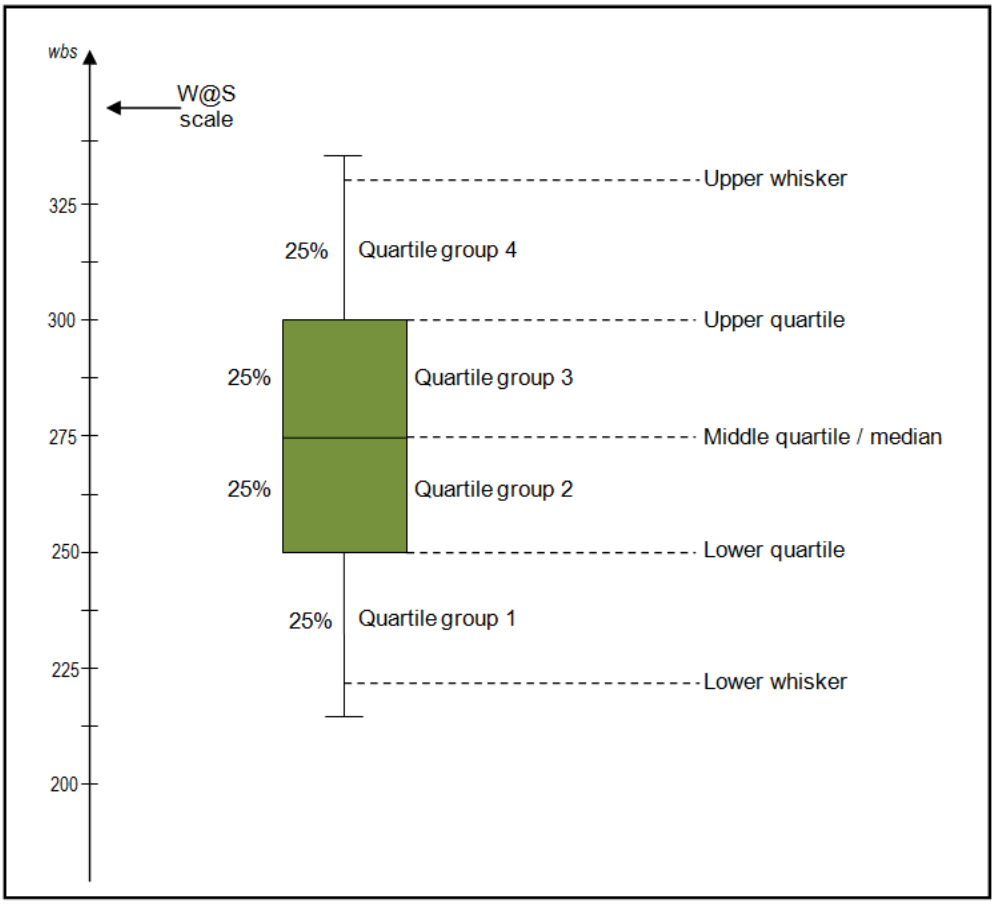


Figure 3.15: Different parts of a boxplot

5.1 Median

The median (middle quartile) marks the mid-point of the data and is shown by the line that divides the box into two parts. Half the scores are greater than or equal to this value and half are less. [53]

5.2 Inter-quartile range

The middle "box" represents the middle 50% of scores for the group. The range of scores from lower to upper quartile is referred to as the inter-quartile range. The middle 50% of scores fall within the inter-quartile range.[53]

5.3 Upper quartile

Seventy-five percent of the scores fall below the upper quartile.

5.4 Lower quartile

Twenty-five percent of scores fall below the lower quartile.

5.5 Whiskers

The upper and lower whiskers represent scores outside the middle 50%. Whiskers often (but not always) stretch over a wider range of scores than the middle quartile groups.[53]

6. Measures of Evaluating the performance of the models:

We used various evaluation measures to assess the results, and these measures are described below:

6.1 Dice:

The Dice coefficient is the similarity/overlap between two images [54], it's compares the area of overlap of the ground truth and predicted regions with the total area of both regions. A Dice score of 1.0 indicates a perfect segmentation. Dice score is calculated using the formula:

$$\text{Dice}(P,T) = \frac{2|P_1 \cap T_1|}{|P_1| + |T_1|}$$

Where ∩ is the logical AND operator, | | is the size of the set (i.e., the number of voxels belonging to it). P1 and T1 represent the numbers of voxels belonging to algorithm's prediction and ground truth, respectively. The Dice score normalizes the number of true positives to the average size of predicted and ground truth-segmented area. It also gives us the voxel wise overlap between the result and ground truth [55].

6.2 Jaccard coefficient

The Jaccard coefficient measures the similarity between two images and can be defined as the size of intersection divided by the size of union of two sets. Jaccard coefficient is also known as Jaccard index and can be measured as:

$$\text{Jaccard}(P,T) = \frac{P_1 \cap T_1}{P_1 \cup T_1}$$

The two measures are graphically explained in **Figure 16:**

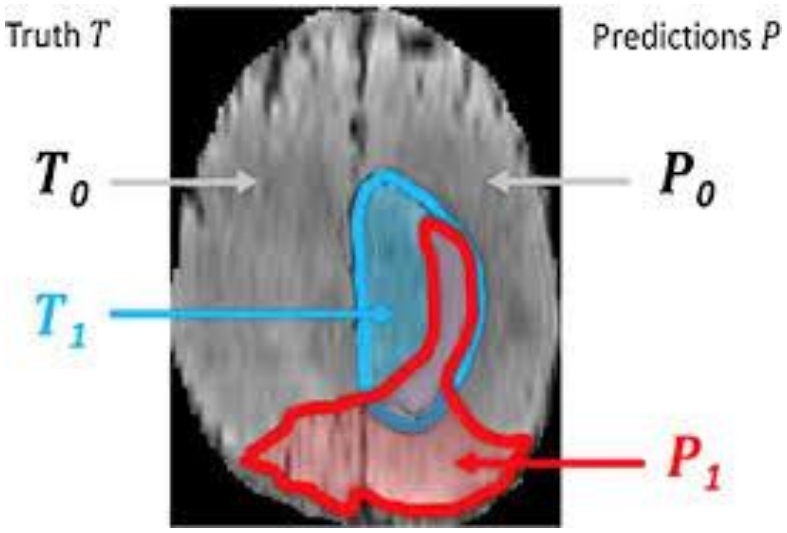


Figure 3.16: T1 is the ground truth lesion, and T0 is the area outside T1 within the brain. P1 is the algorithm's predicted lesion, and P0 is the algorithm's predicted area outside P1 within the brain. Overlapped area between T1 and P1 gives us the true positive. [49]

7. ***Conclusion:***

In this chapter, we present our proposed approach to detect and to segments the brain tumor and edema, in addition, we explained the models of our approach with detailed diagrams, then we presented the data set used with its description; at the end we defined the evaluation metrics we used.

Chapter 04:

Implementation and Evaluation

1 Introduction:

In this chapter we present the implementation of the proposed approach. We start with the presentation of the hardware and software development tools as well as the configuration parameters.

The obtained results are presented in detail, compared and discussed with examples on images of the datasets used.

2 Presentation of tools:

2.1 Hardware:

We used several methods (Pre-Processing, Detection and Recon- naissance), the latter require high performance which is why we used Google Colab, a tool that offers 12 hours a day. Our system is based on image processing, it requires the GPU (Graphics Processing Unit) that Google Colab allows us to use in a free online 'Jupyter' notebook environment.

- GPU: 1xTesla K80, having 2496 CUDA cores, compute 3.7, 12 GB (11.439 GB usable).
- CPU: 1x single hyper threaded core, i.e. (1 core, 2 threads) Xeon @ 2.3GHz processors (NonTurbo Boost), 45MB cache.
- RAM: 12 Gb available.
- Disk: 64 GB available.



Figure 4. 1: Google Collab logo.

2.2 Development environment:

We present in this section, the programming languages used, also the development environment:

2.2.1 Python:

Python is a general-purpose high-level dynamic programming language created by the developer Guido Van Rossum in 1991, itself as one of the most popular languages for scientific computing. [56]

The Python is widely used in bigger organizations because of its multiple programming paradigms, it can be used for different application (console app, Desktop app, web app, Mobile app, Machine learning, Deep learning...).

They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features. [57]



Figure 4. 2:python logo.

2.2.2 Python's features:

Python has many features which the other programming language does not have. It is easier to learn options such as Python and is less time consuming to write a program.

Below we have some unique features which make python stand out from another programming language:

2.2.2.1 Easy-to-learn:

Python is easy to learn since the code is not complex and the language appears to be plain English. This makes it easy to read and create a program.

Easy to code: Anyone can code with python because:

- Popular (scripting/extension) language.
- clear and easy syntax.
- no type declarations.
- automatic memory management.

Easy to Read:

- high-level data types and operations.
- design to read (more English like syntax) and write (shorter code compared to C, C++, and Java which allows the user to invest less time and effort to code a programme.) fast. [58]

2.2.2.2 Free and Open Sources:

Python is publicly available open source software, anyone can use source code and we can do whatever we want to download, change, use and distribute it. This is called FLOSS which means Free/Libre and Open Source Software. [58]

2.2.2.3 High-Level Programming Language:

Python is a high-level programming language (closer to human) refers to the higher level of concept from machine language (for example assembly languages). which means there is no need to memorize the system architecture. There is also no requirement to manage memory. This feature makes Python more programmers friendly.[58]

2.2.2.4 Expressive:

Python helps us focus on the solution rather than investing the mind in syntax, because it provides us with a myriad of constructs. This is an amazing feature that can enlighten us on "why should you learn this particular system?". [58]

2.2.2.5 Portable:

High level languages are portable, which means they are able to run across all major hardware and software platforms with few or no change in source code. Python is portable and can be used on Linux, Windows, Macintosh, Solaris, FreeBSD, OS/2, Amiga, AROS, AS/400 and many more.[58]

2.2.2.6 Object-Oriented:

Python is known as a full-featured object-oriented programming language because it can model the real world. basically, focuses on an object and combines functions and data, with features such as classes, inheritance, and overloading.[58]

2.2.2.7 Interactive:

Python has an interactive console where you get a Python prompt (command line) and interact with the interpreter directly to write and test your programs. This is useful for mathematical programming.[58]

2.2.2.8 Interpreted:

A Python program are interpreted, takes source code as input, and then compiles (to portable byte-code) each statement and executes it immediately. No need to compiling or linking. This means the Python code can be run without stressing over connecting to libraries and other different things.[58]

2.2.2.9 Extendable:

Python can be extended to other languages which make it extensible language. This feature allows you to write some of the Python codes ion other languages C++ or Java.[58]

2.2.2.10 Large Standard Library:

Python has a large built-in library that can be used while we're coding a program so that we don't have to write our own code for everything. In Python, the library involves a grouping of modules where we will locate all kinds of things. Modules are records, just like books that

contain functions that need to be imported into our program. For example, Databases, web services, networking, numerical packages, graphical user interfaces,3D graphics, others.[58]

2.2.2.11 Embeddable:

Embeddable means we can also put our Python code in different languages source code, like C++. This enables the users to coordinate scripting capabilities into the programmer of other languages. [58]

2.2.2.12 Supports:

Support from online Python community. [59]

2.2.2.13 GUI Programming:

In Python, we can create basic GUI's by using Tk. [58]

2.2.2.14 Dynamically Typed:

Python is dynamically-typed. This implies the Type for esteem is chosen at runtime, not progress of time. This is the reason we don't have to determine the sort of information while pronouncing it. [58]

2.2.3 Python's used modules (Environment):

In our project we used the following modules:

2.2.3.1 TensorFlow:

TensorFlow is an open source Python library for fast numerical computing created and released by Google. This is a simple library which can be used to construct Deep Learning models either directly or with the help of wrapper libraries that make the process easier.

Google built and maintains it, and its open source under the Apache 2.0 license. Although there is access to the underlying C++ API, the API is ostensibly for the Python programming language.

Unlike other numerical libraries for Deep Learning, like as Theano, TensorFlow was created with the goal of being used in both research and production systems, including Google's Rank Brain and the fun Deep Dream project.

It can run on single-processor computers, GPUs, mobile devices, and large-scale distributed systems involving hundreds of machines. [60]

This allows us to implement and manipulate all architectures in a simple and easy way



Figure 4. 3: TensorFlow logo.

2.2.3.2 ***Keras:***

Keras is a high-level API for building and training deep learning models, capable of running on TensorFlow, it is used for rapid prototyping, advanced research and production. Keras models are made by connecting configurable building blocks together, so Keras allows us to create new layers, functions and develop state-of-the-art models with few restrictions in a few lines of code. [61]

It has all the layers of our model u-net.



Figure 4. 4: keras logo.

2.2.3.3 Backend:

Keras is a model-level library, offers high-level building blocks that are useful to develop deep learning models. Instead of supporting low-level operations such as tensor products, convolutions, etc. itself, it depends upon the backend engine that is well specialized and optimized tensor manipulation library. It doesn't pick just one library of a tensor to implement Keras tied to that particular library. It handles the situation in a modular way by seamlessly plugging many distinct back-end engines to Keras. [62]

We used Backend to create function like dice and Jaccard because it has mathematical methods.

2.2.3.4 Glob:

The glob module in Python is used to find files/pathnames that match a pattern. The glob pattern rules are the same as the Unix path expansion rules. It is also expected that, based on benchmarks, it would match pathnames in directories quicker than previous techniques. Apart from exact text search, we may utilize wildcards ("*,?, [ranges]) with glob to make path retrieval more easy and convenient. [63]

We used Glob to extracts the path of each image.

2.2.3.5 OS:

The OS module in Python provides functions for interacting user with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality. The Python OS module lets us work with the files and directories. [64]

We used OS to work with files (create, delete).

2.2.3.6 Sys:

The python sys module provides functions and variables which are used to manipulate different parts of the Python Runtime Environment. It lets us access system-specific parameters and functions. [65]

We used Sys to change the configuration system (a full NumPy array).

2.2.3.7 NumPy:

NumPy is a general-purpose array-processing package, is the fundamental package for scientific computing with Python. It contains among other things:

- A powerful and high-performance multidimensional array object, and tools for working with these arrays.
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities
- Can also be used as an efficient multi-dimensional container of generic data. [66]

We used NumPy for all processing of data and shaping.

2.2.3.8 Pickle:

The pickle module supports binary serialization and de-serialization protocols for Python object structures. Pickling is the process of converting a Python object hierarchy into a byte stream, while unpickling is the process of converting a byte stream back into an object hierarchy.

We used pickle to save the data in file with extension .Pkl and later loaded into a program..

2.2.3.9 Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell (MATLAB or Mathematica), web application servers, and various graphical user interface toolkits. [67]

We used matplotlib for plotting image and data visualization.

2.2.3.10 CV2:

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. cv2. imread () method loads an image from the specified file. Is a library for computer vision tasks. [68]

We used cv2 to read and to upload all image of data.

2.2.3.11 ***Json:***

JSON stands for JavaScript Object Notation is a format for structuring data. It is mainly used for storing and transferring data between the browser and the server. Python too supports JSON with a built-in package called json. This package provides all the necessary tools for working with JSON Objects including parsing, serializing, deserializing, and many more. [69]

We used json to convert our data to format json.

2.2.4 ***Version of modules used:***

Table 4. 1: versions of the python modules used

Modules	Version used
TensorFlow	2.6.0
Keras	2.6.0
Backend	2.6.0
Glob	Basic Version
OS	Basic Version
Json	2.0.9
NumPy	1.19.5
Pickle	Basic Version
Matplotlib	3.2.2
Sys	Basic Version
CV2	4.1.2

3 Results and Discussions:

In this part, we will present the results of our models based on the CNN architecture with U-NET model specialized in the segmentation and detection of medical images.

We have created ten (10) models that have been used for the detection of edemas and brain tumors through the different MRI modalities, such as four (04) among them are T1, T2, T1C and FLAIR and the other six (06) are a fusion of the two modalities that are T1/FLAIR, T1/T1C, T1/T2, T1C/FLAIR, T2/FLAIR andT1C/T2. We took from our dataset twenty-four (24) patients to train the models (we present four (04) patients like examples in tables of our results) and six (06) patients for testing. Finally, the evaluation of the performance was measured by the two coefficients dice and Jaccard.

3.1 Result of no merge models:

After the evaluation of the detection stage, we summarize the tumor and edema tables showing the results obtained for no merge models as well as the training and testing data set.

such as for the tumor results (**Table 4.2**) the first model **T1** has dice=0.960, Jaccard=0.936, the second model **T2** has dice=0.932, Jaccard=0.897, the third model **T1C** has dice=0.704, Jaccard=0.626, and the last model **FLAIR** has dice=0.632, Jaccard=0.518.

and concerning the edema results (**Table 4.3**) we have the **T1** model dice=0.953, Jaccard=0.927, **T2** model dice=0.863, Jaccard=0.897, **T1C** model dice=0.623, Jaccard=0.532, and **FLAIR** model dice=0.446, Jaccard=0.571.

3.1.1 Comparison of results

The comparison between the results shows that **T1** presents better results than the other models in the two cases tumor and edema. In fact, this modality distinguishing healthy tissues, in contrast to the **FLAIR** model gave us very weak and insufficient results. knowing that each model took 1H and 10min of running time; like it shows in the boxplot of no merging models figures (**Figure 4.5,4.6**).

In **figure 4.7,4.8,4.9,4.10** we present some images to show how similar are the truth slices made by the experts and the predict slices made by the models no merging (T1, T2, T1C and FLAIR), such as the white area describes the tumor and the gray area reflects the edema.

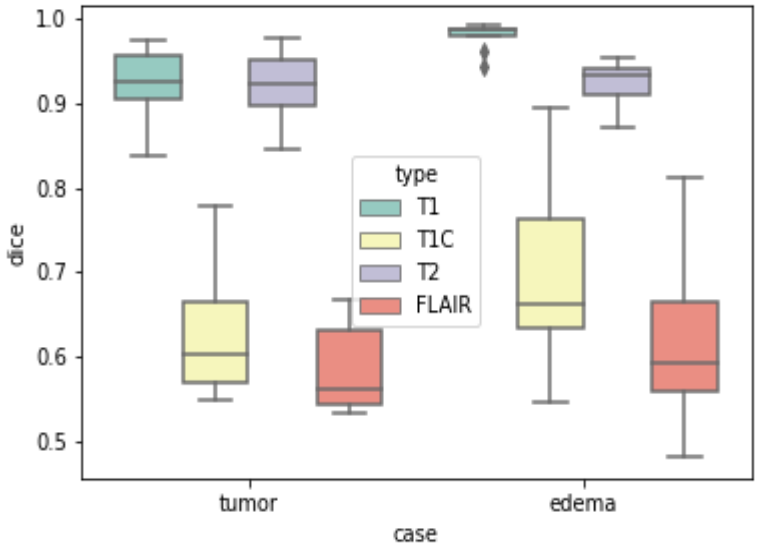


Figure 4. 5:Figure represent the Boxplot tumor and edema prediction results for no merging models of dice coefficient.

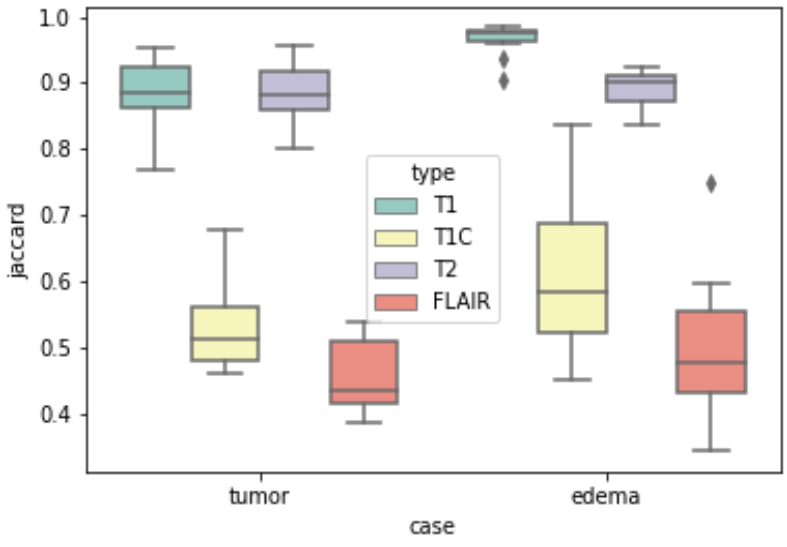


Figure 4. 6: Figure represent the Boxplot tumor and edema prediction results for no merging models of Jaccard coefficient.

Table 4. 2: Table represent the tumor prediction results for no merging models (T1, T2, T1C,FLAIR)

Patient N°	Train /Test	T1		T2		T1C		Flair	
		Jaccard	Dice	Jaccard	Dice	Jaccard	Dice	Jaccard	Dice
HG0001	Train	0.975	0.987	0.902	0.934	0.704	0.784	0.558	0.673
HG0015		0.969	0.983	0.871	0.906	0.634	0.696	0.486	0.598
LG0006		0.979	0.988	0.911	0.939	0.588	0.672	0.449	0.574
LG0008		0.981	0.990	0.910	0.942	0.837	0.893	0.748	0.813
Mean		0.976	0.987	0.898	0.930	0.690	0 .761	0.560	0.664
HG0025	Test	0.866	0.908	0.865	0.921	0.479	0.548	0.425	0.541
HG0026		0.884	0.927	0.880	0.924	0.557	0.651	0.490	0.631
HG0027		0.849	0.890	0.855	0.889	0.562	0.669	0.387	0.532
LG0013		0.953	0.974	0.956	0.975	0.515	0.569	0.532	0.632
LG0014		0.931	0.962	0.957	0.977	0.508	0.617	0.425	0.551
LG0015		0.904	0.944	0.869	0.920	0.757	0.839	0.597	0.721
Mean		0.897	0.934	0.897	0.934	0.563	0.648	0.476	0.601
Mean globale		0.936	0.960	0.897	0.932	0.626	0.704	0.518	0.632

Table 4. 3: Table represent the edema prediction results for no merging models(T1,T2,T1C,FLAIR)

Patient N°	Train /Test	T1		T2		T1C		Flair	
		Jaccard	Dice	Jaccard	Dice	Jaccard	Dice	Jaccard	Dice
HG0001	Train	0.974	0.986	0.835	0.871	0.539	0.655	0.399	0.532
HG0015		0.961	0.979	0.730	0.907	0.519	0.630	0.426	0.553
LG0006		0.976	0.987	0.902	0.931	0.466	0.582	0.344	0.481
LG0008		0.984	0.991	0.916	0.949	0.451	0.547	0.548	0.638
Mean		0.973	0.985	0.845	0.914	0.493	0.603	0.429	0.551
HG0025	Test	0.770	0.836	0.801	0.846	0.488	0.589	0.413	0.547
HG0026		0.896	0.936	0.902	0.939	0.472	0.561	0.446	0.571
HG0027		0.859	0.905	0.854	0.893	0.462	0.569	0.400	0.538
LG0013		0.886	0.923	0.885	0.922	0.654	0.724	0.538	0.666
LG0014		0.944	0.969	0.924	0.955	0.677	0.778	0.515	0.648
LG0015		0.936	0.962	0.922	0.954	0.582	0.645	0.470	0.589
Mean		0.881	0.921	0.881	0.918	0.571	0.644	0.463	0.592
Mean globale		0.927	0.953	0.863	0.916	0.532	0.623	0.446	0.571

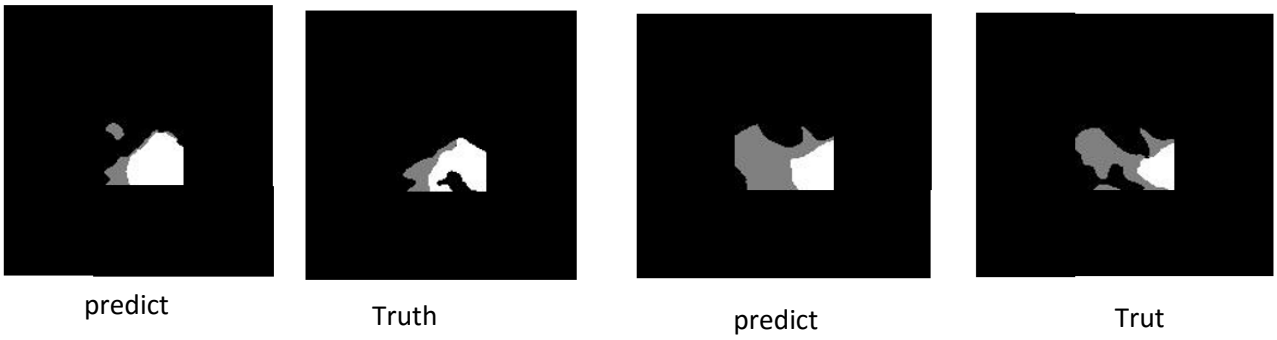


Figure 4. 7: Examples of the prediction result of model T1

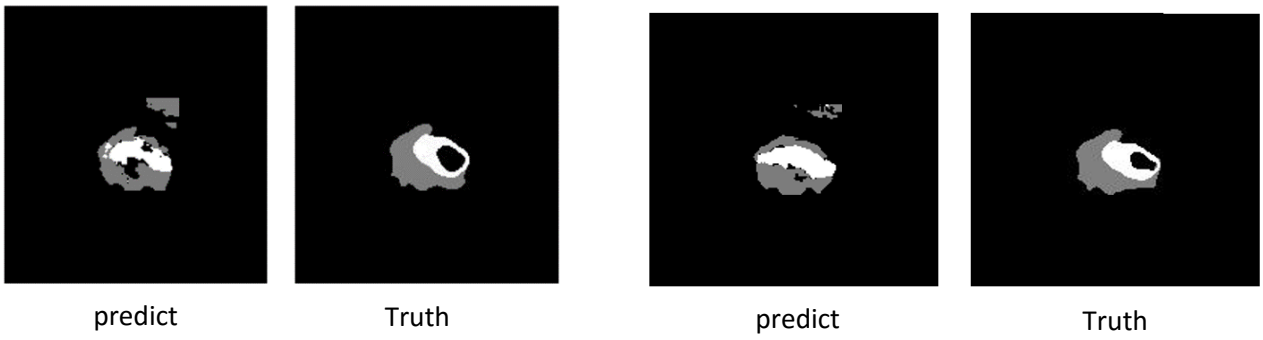


Figure 4. 8: Examples of the prediction result of model T2

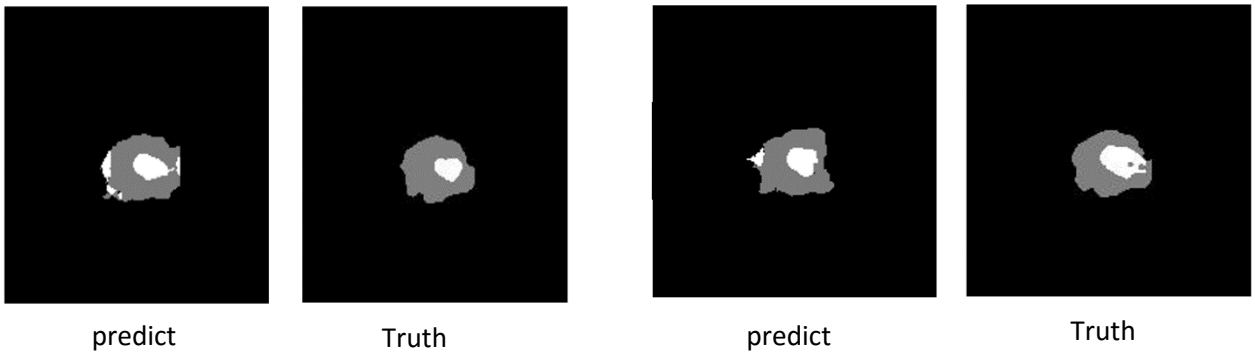


Figure 4. 9: Examples of the prediction result of model T1C

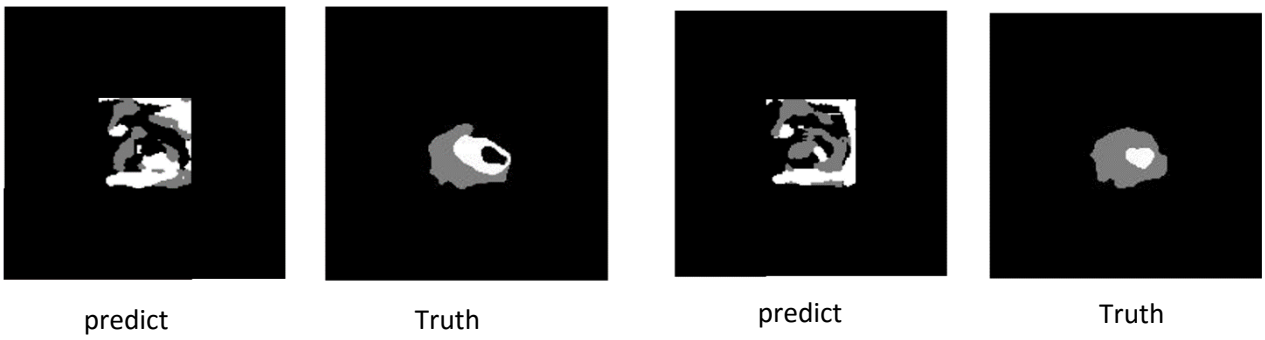


Figure 4. 10: Examples of the prediction result of model FLAIR

3.2 Result of merge models:

In this part we will also present the evaluation of the tumor and edema tables showing the results obtained for merge models as well as the training and testing data set.

such as for the tumor results (**Table 4.4,4.6**) the first merge model **T1/FLAIR** has dice=0.950, Jaccard=0.828, the second merge model **T1/T1C** has dice=0.797, Jaccard=0.741, the third merge model **T1/T2** has dice=0.705, Jaccard=0.609, and the fourth merge model **T1C/FLAIR** has dice=0.578, Jaccard=0.485,then fifth merge model **T2/FLAIR** has dice=0.587, Jaccard=0.495, and the sixth merge model **T1C/T2** has dice= 0.791, Jaccard=0.752

and concerning the edema results (**Table 4.5,4.7**) for merge models we have the **T1/FLAIR** model dice=0.894, Jaccard=0.841, **T1/T1C** model dice=0.821, Jaccard=0.729, **T1/T2** model dice=0.629, Jaccard=0.532, and **T1C/FLAIR** model dice=0.483, Jaccard=0.380, then model **T2/FLAIR** has dice=0.507, Jaccard=0.309, and the model **T1C/T2** has dice=0.749, Jaccard=0.737.

3.2.1 Comparison of results

The comparison between the results of merge models in tumor and edema shows that **T1/FLAIR** merge model presents better results than the other merge models. In fact, the modality T1 highlights the whole healthy tissues and FLAIR aids in identifying the edema area from the Cerebrospinal Fluid. considering that each model took 2 hours and 20 minutes to complete execution; like it shows in the boxplot of merging models figures (**Figure 4.5,4.6**). (**Figure 4.10**).

In **figures 4.13,4.14,4.15,4.16,4.17,4.18** we present also some images of the truth slices and the predict slices made by the merging models (T1/FLAIR, T1/T1C, T1/T2, T1C/FLAIR, T2/FLAIR and T1C).

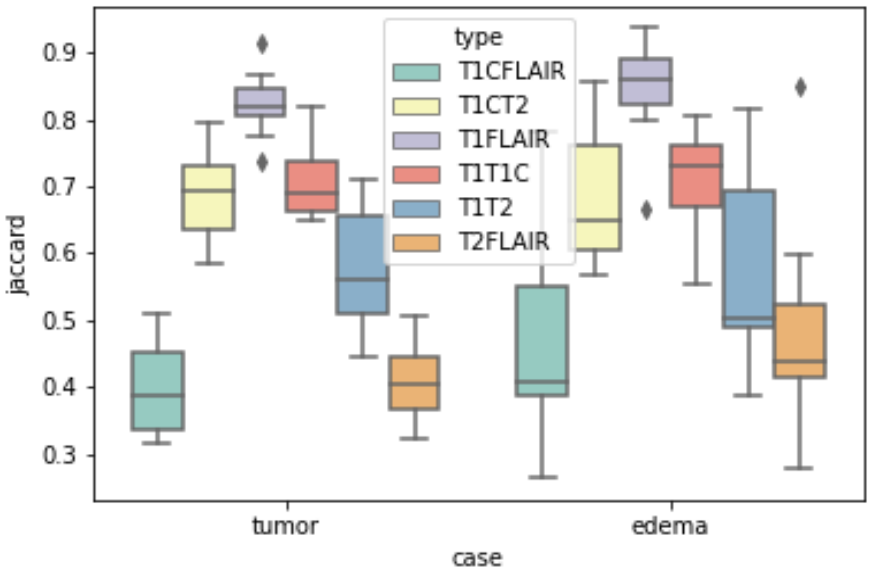


Figure 4. 11: Figure represent the Boxplot tumor and edema prediction results for merging models of Jaccard coefficient.

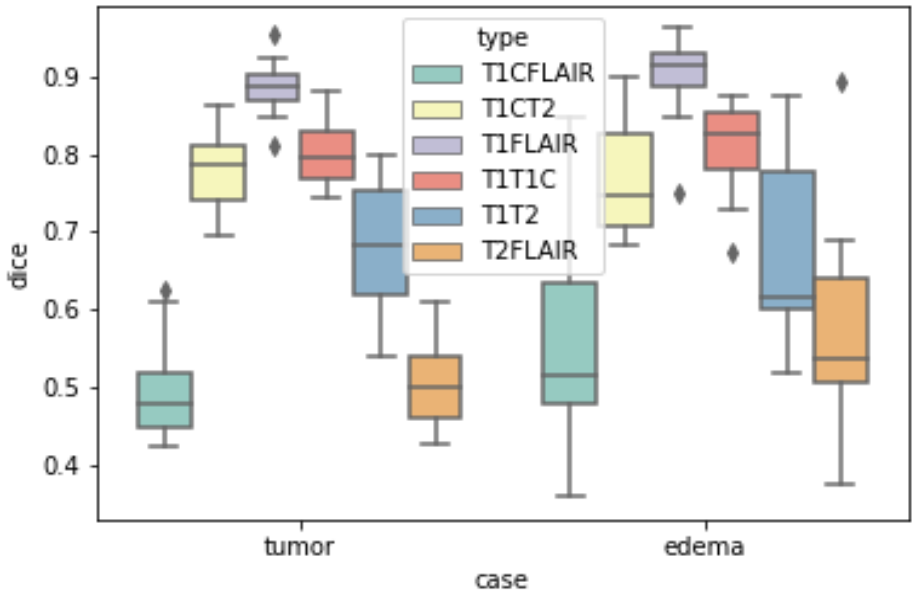


Figure 4. 12: Figure represent the Boxplot tumor and edema prediction results for merging models of dice coefficient.

Table 4. 4: Table represent the tumor prediction results for merging models (T1/FLAIR, T1/T1C, T1/T2)

Patient N°	Train/ Test	T1/FLAIR		T1/T1C		T1/T2	
		Jaccard	Dice	Jaccard	Dice	Jaccard	Dice
HG0001	Train	0.852	0.908	0.725	0.824	0.676	0.756
HG0015		0.665	0.749	0.555	0.672	0.487	0.593
LG0006		0.875	0.919	0.732	0.826	0.491	0.600
LG0008		0.896	0.932	0.659	0.778	0.814	0.874
Mean		0.822	0.877	0.666	0.775	0.617	0.705
HG0025	Test	0.774	0.848	0.786	0.862	0.479	0.555
HG0026		0.812	0.886	0.720	0.862	0.530	0.646
HG0027		0.802	0.863	0.660	0.818	0.575	0.703
LG0013		0.914	0.952	0.817	0.769	0.663	0.757
LG0014		0.864	0.924	0.656	0.880	0.631	0.738
LG0015		0.843	0.909	0.607	0.730	0.732	0.832
Mean		0.834	0.897	0.817	0.820	0.601	0.705
Mean globale		0.828	0.950	0.741	0.797	0.609	0.705

Table 4. 5: Table represent the edema prediction results for merging models (T1/FLAIR, T1/T1C, T1/T2)

Patient N°	Train/ Test	T1/FLAIR		T1/T1C		T1/T2	
		Jaccard	Dice	Jaccard	Dice	Jaccard	Dice
HG0001	Train	0.797	0.848	0.696	0.787	0.488	0.601
HG0015		0.901	0.939	0.765	0.854	0.499	0.621
LG0006		0.814	0.880	0.806	0.874	0.388	0.518
LG0008		0.936	0.961	0.784	0.870	0.506	0.513
Mean		0.862	0.907	0.762	0.846	0.470	0.563
HG0025	Test	0.736	0.809	0.650	0.744	0.444	0.541
HG0026		0.810	0.883	0.741	0.834	0.501	0.608
HG0027		0.823	0.879	0.700	0.799	0.547	0.666
LG0013		0.850	0.902	0.663	0.770	0.709	0.797
LG0014		0.838	0.895	0.678	0.791	0.677	0.776
LG0015		0.867	0.924	0.747	0.847	0.697	0.785
Mean		0.820	0.882	0.696	0.797	0.595	0.695
Mean globale		0.841	0.894	0.729	0.821	0.532	0.629

Table 4. 6: Table represent the tumor prediction results for merging models (T1C/FLAIR, T2/FLAIR, T1C/T2)

Patient N°	Train/ Test	T1C/FLAIR		T2/FLAIR		T1C/T2	
		Jaccard	Dice	Jaccard	Dice	Jaccard	Dice
HG0001	Train	0.575	0.661	0.598	0.688	0.770	0.833
HG0015		0.418	0.521	0.418	0.528	0.597	0.693
LG0006		0.383	0.479	0.414	0.498	0.649	0.746
LG0008		0.782	0.847	0.849	0.893	0.855	0.899
Mean		0.539	0.627	0.569	0.651	0.717	0.792
HG0025	Test	0.381	0.456	0.379	0.458	0.607	0.707
HG0026		0.408	0.519	0.418	0.541	0.722	0.806
HG0027		0.336	0.463	0.321	0.441	0.658	0.762
LG0013		0.464	0.520	0.444	0.510	0.707	0.790
LG0014		0.394	0.496	0.443	0.532	0.734	0.813
LG0015		0.613	0.725	0.526	0.665	0.783	0.871
Mean		0.432	0.529	0.421	0.524	0.787	0.791
Mean globale		0.485	0.578	0.495	0.587	0.752	0.791

Table 4. 7: Table represent the edema prediction results for merging models (T1/FLAIR, T2/FLAIR, T1C/T2)

Patient N°	Train/ Test	T1C/FLAIR		T2/FLAIR		T1C/T2	
		Jaccard	Dice	Jaccard	Dice	Jaccard	Dice
HG0001	Train	0.396	0.512	0.418	0.543	0.645	0.747
HG0015		0.345	0.456	0.367	0.478	0.630	0.746
LG0006		0.264	0.361	0.279	0.377	0.567	0.683
LG0008		0.394	0.480	0.516	0.572	0.588	0.686
Mean		0.349	0.452	0.198	0.492	0.781	0.715
HG0025	Test	0.327	0.447	0.360	0.470	0.585	0.694
HG0026		0.341	0.427	0.391	0.490	0.680	0.780
HG0027		0.315	0.426	0.322	0.428	0.626	0.734
LG0013		0.503	0.609	0.504	0.609	0.748	0.831
LG0014		0.508	0.626	0.493	0.606	0.795	0.862
LG0015		0.474	0.555	0.459	0.534	0.735	0.808
Mean		0.411	0.515	0.421	0.522	0.694	0.784
Mean globale		0.380	0.483	0.309	0.507	0.737	0.749

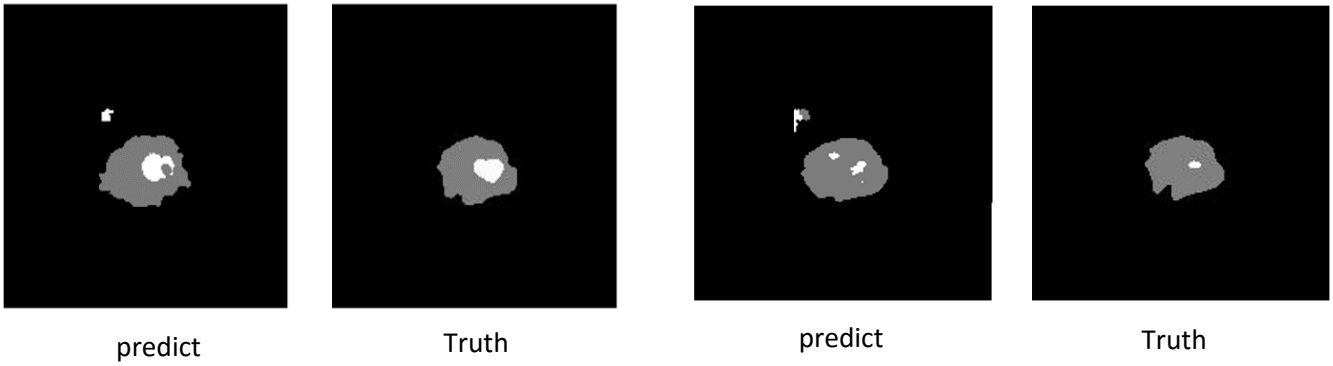


Figure 4. 13: Examples of the prediction result of model T1/FLAIR.

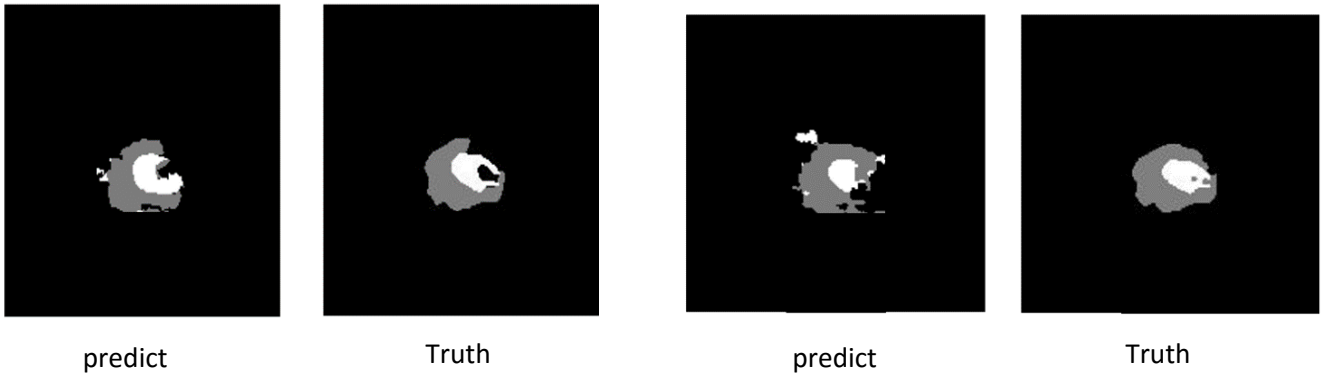


Figure 4. 14: Examples of the prediction result of model T1/T1C.

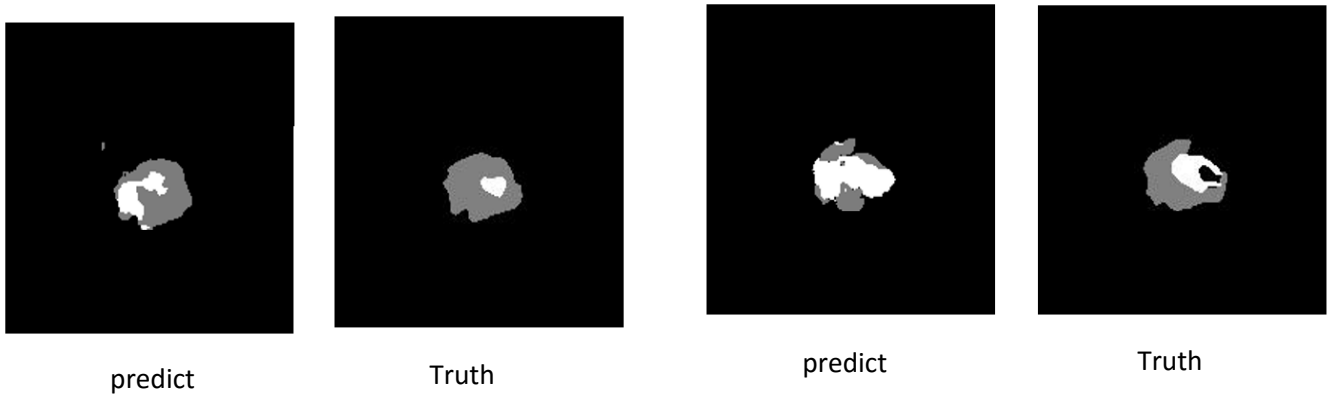


Figure 4. 15: Examples of the prediction result of model T1/T2.

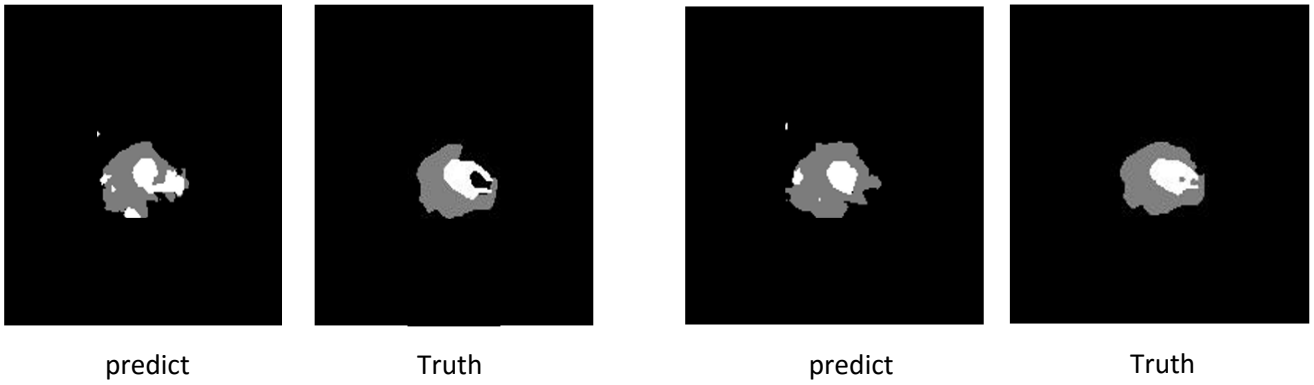


Figure 4. 16: Examples of the prediction result of model T1C/FLAIR.

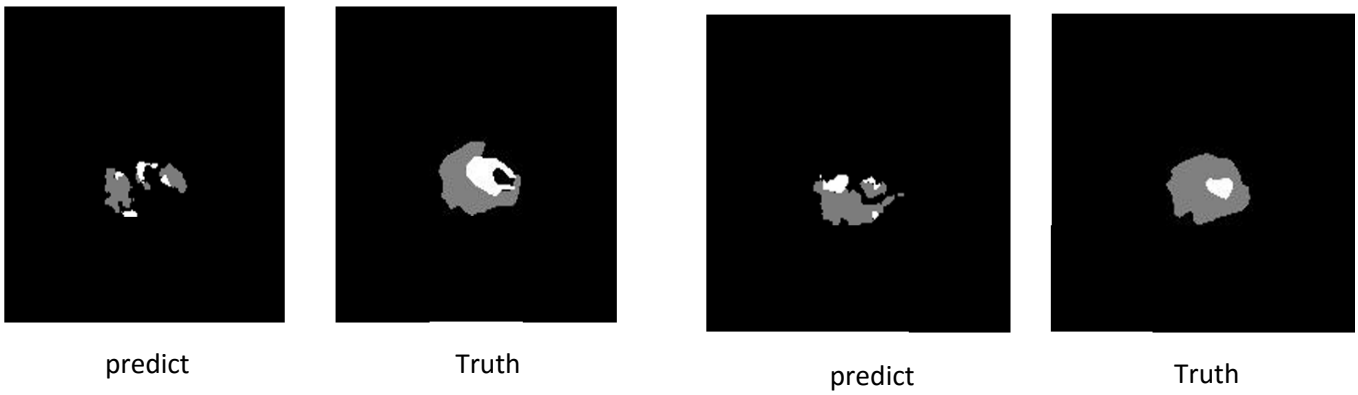


Figure 4. 17: Examples of the prediction result of model T2/FLAIR.

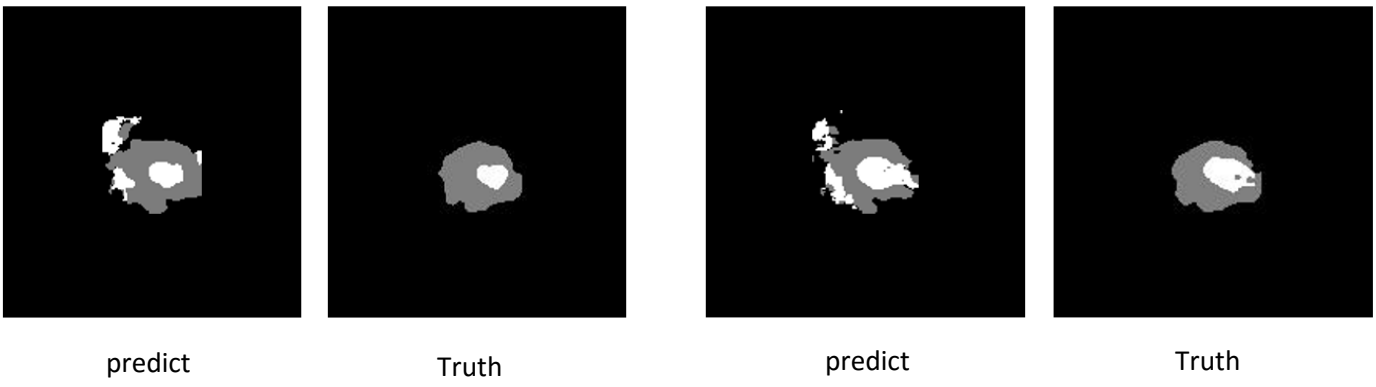


Figure 4. 18: Examples of the prediction result of model T1C/T2

4 Global evaluation:

The previous sections evaluated a comprehensive assessment of each phase of our work with comparisons of results. We conclude that the best results obtained for the detection of brain tumor in the non-fusion models and the fusion models are **T1, T1 / FLAIR**, successively, and for the detection of brain edema in non-fusion models and fusion models are **T1, T1/FLAIR**, successively.

In this section, depending on the conclusion that we have, we notice that merging models are better than no merge models, which means they predicted the edema and tumor brain with a large percentage near from what the experts make; like it shows in the boxplot of merging and no merging models figures (**Figure 4.19**)

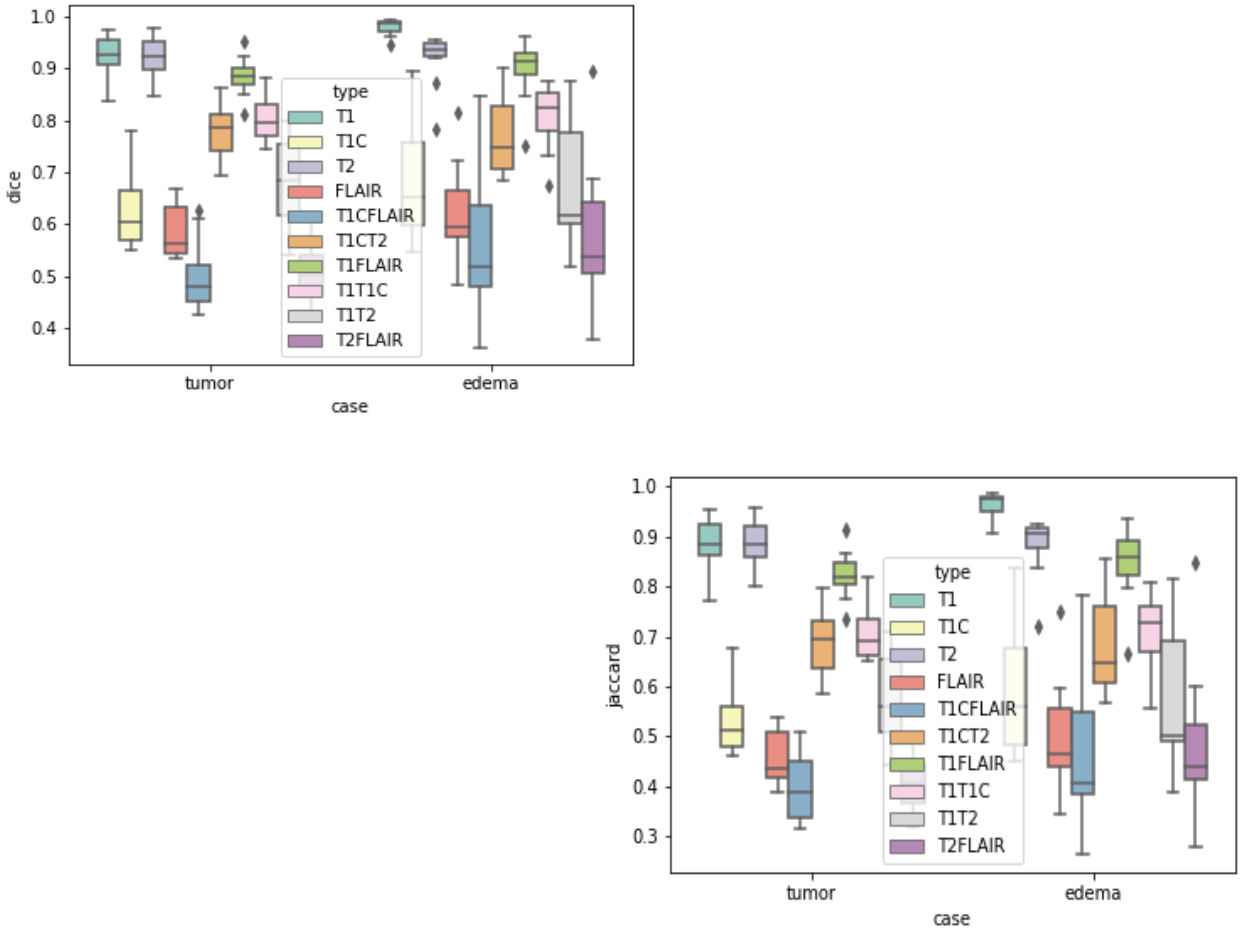


Figure 4. 19: Figure represent the Boxplot tumor and edema prediction results for merging and no merging models of Jaccard and dice coefficient.

In **figures 4.20** we present some examples of the prediction result of low-grade for merge models with truth slices.

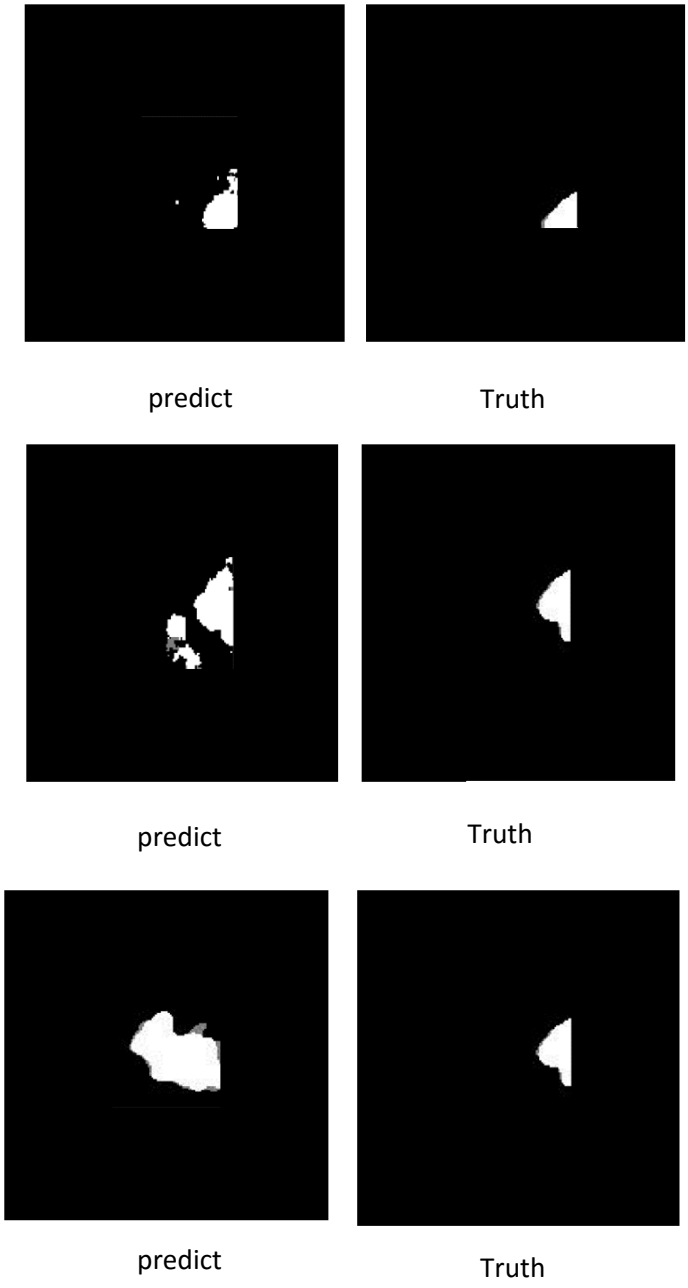


Figure 4. 20: Examples of the prediction result of low-grade for merge models.

5 Discussion

The overall evaluation of this system, for the detection of brain tumor and edema illustrate that it can detect with a large percentage both of tumor and edema in High-grade and low-grade classification in the case of merging models better than no merging models, which means they predicted the edema and tumor brain with a large percentage near from what the experts make. If we compare our results with the related work that we present in chapter 01 we found that we complete the messing elements in each work like using the fourth modalities for detection both tumor and edema brain in the tow cases low-grade and high-grade, it remains an honorable result which stimulates research and development in this field. So, with this system we might save more people's lives.

6 conclusion:

In this chapter we have seen the implementation of the proposed approach. The results for each phase have been presented, compared and discussed.

General conclusion and perspectives

General conclusion and perspective

In this thesis, we presented a new approach of brain edema and tumor detection using hybrid intelligent algorithms. For a better understanding of the problem and the difficulties First, an introduction to brain tumors and methods for brain tumor and edema segmentation is given, we also defined the MRI scanner with its different modalities and how we used them in our work, their classifications as well as the challenges that present.

After having identified the problems, we are at the end of proposing an effective solution. We have presented the most used and efficient methods and techniques that allow the detection of both brain edema and tumor with segmentation of medical image using MRI slices image. the U-NET architectures was selected for detection and segmentation. first, we used the four MRI modalities, after we starting to merge two modalities one after one, we took the results of both cases for brain tumor and edema detection. Also, we used dice and Jaccard coefficient to evaluate the performance of our models.

The results we obtained allow us to conclude that: for the detection of both tumor and edema the merge models are better than the no merge models. Brain tumor and edema segmentation isn't an easy task. Due to the genetics of cancer, it remains being a task for which the doctor's help is needed. However, they don't have to do all by themselves, since with tools like the one that we have designed in this work, we can help them and we can contribute in the fight against cancer.

Prospects The methods proposed in this thesis have shown very good performance. In perspective, it is planned to improve the detection system and extend it to all types of MRI image. Another perspective would be the implementation of the method on MRI's machine for validation in a real environment.

REFERENCES

[1] A. Işin, C. Direkoğlu, and M. Şah, "Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods," *Procedia Comput. Sci.*, vol. 102, no. August, pp. 317–324, 2016, doi: 10.1016/j.procs.2016.09.407.

[2] "Algeria." https://gco.iarc.fr/today/data/factsheets/populations/12-algeria-fact-sheets.pdf

[3] A. Wadhwa, A. Bhardwaj, and V. Singh Verma, "A review on brain tumor segmentation of MRI images," *Magn. Reson. Imaging*, vol. 61, no. May, pp. 247–259, 2019, doi: 10.1016/j.mri.2019.05.043.

[4] I. Njeh *et al.*, "3D multimodal MRI brain glioma tumor and edema segmentation: A graph cut distribution matching approach," *Comput. Med. Imaging Graph.*, vol. 40, pp. 108–119, 2015, doi: 10.1016/j.compmedimag.2014.10.009.

[5] M. Amian and M. Soltaninejad, "Multi-resolution 3D CNN for MRI Brain Tumor Segmentation and Survival Prediction," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11992 LNCS, pp. 221–230, Oct. 2019, doi: 10.1007/978-3-030-46640-4_21.

[6] "Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods | Elsevier Enhanced Reader." https://reader.elsevier.com/reader/sd/pii/S187705091632587X?token=13278B5D02F1F607ECC09A4EE60E1AD734490097BE0EEBF7E65DBB7ACF54CC7F91D1AB34F05332AE99C12B9A9DD0D20D&originRegion=eu-west-1&originCreation=20210824234724 (accessed Aug. 25, 2021).

[7] M. S. Alam *et al.*, "Automatic human brain tumor detection in mri image using template-based k means and improved fuzzy c means clustering algorithm," *Big Data Cogn. Comput.*, vol. 3, no. 2, pp. 1–18, 2019, doi: 10.3390/bdcc3020027.

[8] N. Kesav and M. G. Jibukumar, "Efficient and low complex architecture for detection and classification of brain tumor using rcnn with two channel CNN," *J. King Saud Univ. - Comput. Inf. Sci.*, 2021, doi: 10.1016/j.jksuci.2021.05.008.

[9] M. K. Islam, M. S. Ali, M. S. Miah, M. M. Rahman, M. S. Alam, and M. A. Hossain, "Brain tumor detection in MR image using superpixels, principal component analysis and template based K-means clustering algorithm," *Mach. Learn. with Appl.*, vol. 5, no. January, p. 100044, 2021, doi: 10.1016/j.mlwa.2021.100044.

[10] Z. X. Lin, "Glioma-related edema: New insight into molecular mechanisms and their clinical implications," *Chin. J. Cancer*, vol. 32, no. 1, pp. 49–52, 2013, doi: 10.5732/cjc.012.10242.

[11] I. C. Sorribes, M. N. J. Moore, H. M. Byrne, and H. V. Jain, "A Biomechanical Model of Tumor-Induced Intracranial Pressure and Edema in Brain Tissue," *Biophys. J.*, vol. 116, no. 8, pp. 1560–1574, 2019, doi: 10.1016/j.bpj.2019.02.030.

[12] D. S. S. K. Vipin Y. Borole, Sunil S. Nimbhore, "Image Processing Techniques for Brain Tumor Detection: A Review," *Int. J. Emerg. Trends Technol. Comput. Sci. Web*, vol. 4, no. 5, pp. 28–32,

2015, [Online]. Available: http://www.ijettcs.org/Volume4Issue5(2)/IJETTCS-2015-10-01-7.pdf.

[13] "MRI - Mayo Clinic." https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768 (accessed Aug. 27, 2021).

[14] V. Kumar *et al.*, "Brain Tumor Segmentation and Survival Prediction using Automatic Hard mining in 3D CNN Architecture."

[15] M. Ghaffari, A. Sowmya, and R. Oliver, "Automated Brain Tumor Segmentation Using Multimodal Brain Scans: A Survey Based on Models Submitted to the BraTS 2012-2018 Challenges," *IEEE Rev. Biomed. Eng.*, vol. 13, no. October, pp. 156–168, 2020, doi: 10.1109/RBME.2019.2946868.

[16] S. Alqazzaz, X. Sun, X. Yang, and L. Nokes, "Computational Visual Media Automated brain tumor segmentation on multi-modal MR image using SegNet," vol. 5, no. 2, pp. 209–219, 2019, doi: 10.1007/s41095-019-0139-y.

[17] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, 2016, doi: 10.1109/TMI.2016.2538465.

[18] T. Wang, I. Cheng, and A. Basu, "Fluid vector flow and applications in brain tumor segmentation," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 781–789, 2009, doi: 10.1109/TBME.2009.2012423.

[19] V. Harati, R. Khayati, and A. Farzan, "Fully automated tumor segmentation based on improved fuzzy connectedness algorithm in brain MR images," *Comput. Biol. Med.*, vol. 41, no. 7, pp. 483–492, 2011, doi: 10.1016/j.compbiomed.2011.04.010.

[20] B. N. Saha, N. Ray, R. Greiner, A. Murtha, and H. Zhang, "Quick detection of brain tumors and edemas: A bounding box method using symmetry," *Comput. Med. Imaging Graph.*, vol. 36, no. 2, pp. 95–107, 2012, doi: 10.1016/j.compmedimag.2011.06.001.

[21] Y. Zhu *et al.*, "Semi-Automatic Segmentation Software for Quantitative Clinical Brain Glioblastoma Evaluation," *Acad. Radiol.*, vol. 19, no. 8, pp. 977–985, 2012, doi: 10.1016/j.acra.2012.03.026.

[22] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja, "A novel content-based active contour model for brain tumor segmentation," *Magn. Reson. Imaging*, vol. 30, no. 5, pp. 694–715, 2012, doi: 10.1016/j.mri.2012.01.006.

[23] J. Rexilius, H. K. Hahn, J. Klein, M. G. Lentschig, and H.-O. Peitgen, "Multispectral brain tumor segmentation based on histogram model adaptation," *Med. Imaging 2007 Comput. Diagnosis*, vol. 6514, p. 65140V, 2007, doi: 10.1117/12.709410.

[24] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille, "Efficient multilevel brain tumor segmentation with integrated bayesian model classification," *IEEE Trans. Med. Imaging*, vol. 27, no. 5, pp. 629–640, 2008, doi: 10.1109/TMI.2007.912817.

[25] S. Ruan, S. Lebonvallet, A. Merabet, and J. M. Constans, "Tumor segmentation from a multispectral MRI images by using support vector machine classification," *2007 4th IEEE Int. Symp. Biomed. Imaging From Nano to Macro - Proc.*, vol. 2, no. 1, pp. 1236–1239, 2007, doi:

10.1109/ISBI.2007.357082.

[26] I. Mehmood, N. Ejaz, M. Sajjad, and S. W. Baik, “Prioritization of brain MRI volumes using medical image perception model and tumor region segmentation,” *Comput. Biol. Med.*, vol. 43, no. 10, pp. 1471–1483, 2013, doi: 10.1016/j.compbiomed.2013.07.001.

[27] M. Soltaninejad *et al.*, “Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in FLAIR MRI,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 2, pp. 183–203, 2017, doi: 10.1007/s11548-016-1483-3.

[28] J. Marquez, “Multimodal Brain Tumor Segmentation,” *J. Dev. Econ.*, vol. 24, no. 1, pp. 1–27, 1986, [Online]. Available: https://www.cbica.upenn.edu/sbia/Spyridon.Bakas/MICCAI_BraTS/MICCAI_BraTS_2018_proceedings_shortPapers.pdf.

[29] N. B. Bahadure, A. K. Ray, and H. P. Thethi, “Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM,” *Int. J. Biomed. Imaging*, vol. 2017, 2017, doi: 10.1155/2017/9749108.

[30] M. Soltaninejad *et al.*, “Supervised learning based multimodal MRI brain tumour segmentation using texture features from supervoxels,” *Comput. Methods Programs Biomed.*, vol. 157, pp. 69–84, 2018, doi: 10.1016/j.cmpb.2018.01.003.

[31] “(1) What is deep learning? Why is this a growing trend in machine learning? Why not use SVMs? - Quora.” https://www.quora.com/What-is-deep-learning-Why-is-this-a-growing-trend-in-machine-learning-Why-not-use-SVMs (accessed Jun. 05, 2021).

[32] “Deep Learning: Definition, Benefits, and Challenges.” https://www.linkedin.com/pulse/deep-learning-definition-benefits-challenges-oksana-maslovska (accessed Jun. 05, 2021).

[33] Y. Bengio, “Learning deep architectures for AI,” *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–27, Nov. 2009, doi: 10.1561/2200000006.

[34] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015, doi: 10.1038/nature14539.

[35] Y. Li, “Deep Reinforcement Learning: An Overview,” Jan. 2017, Accessed: Jun. 05, 2021. [Online]. Available: http://arxiv.org/abs/1701.07274.

[36] S. Palit, “Studies on Ozone-oxidation of Dye in a Bubble Column Reactor at Different pH and Different Oxidation-reduction Potential.,” *Int. J. Environ. Sci. Dev.*, vol. 1554, pp. 341–346, 2010, doi: 10.7763/ijesd.2010.v1.67.

[37] L. Deng and D. Yu, “Deep learning: Methods and applications,” *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013, doi: 10.1561/2000000039.

[38] D. Of, V. Pathology, and B. Y. Deep, “Acknowledgment,” 2019.

[39] M. Z. Alom *et al.*, “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches,” 2018, [Online]. Available: http://arxiv.org/abs/1803.01164.

[40] “Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville - Google Livros.” https://books.google.dz/books/about/Deep_Learning.html?hl=pt-

BR&id=Np9SDQAAQBAJ&redir_esc=y (accessed Jun. 05, 2021).

[41] C. C. Springs and C. Science, “DEEP LEARNING FOR BRAIN TUMOR SEGMENTATION by MARC MORENO LOPEZ B . S ., Polytechnical University of Catalonia , Barcelona , 2015,” 2017.

[42] “Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | upGrad blog.” https://www.upgrad.com/blog/basic-cnn-architecture/ (accessed Jun. 05, 2021).

[43] D. Dessí *et al.*, “Deep Learning Adaptation with Word Embeddings for Sentiment Analysis on Online Course Reviews,” pp. 57–83, 2020, doi: 10.1007/978-981-15-1216-2_3.

[44] J. Risch and R. Krestel, “Toxic Comment Detection in Online Discussions,” pp. 85–109, 2020, doi: 10.1007/978-981-15-1216-2_4.

[45] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” Accessed: Jun. 05, 2021. [Online]. Available: http://lmb.informatik.uni-freiburg.de/.

[46] P. Zhang, Y. Ban, and A. Nascetti, “Learning U-Net without forgetting for near real-time wildfire monitoring by the fusion of SAR and optical time series,” *Remote Sens. Environ.*, vol. 261, p. 112467, 2021, doi: 10.1016/j.rse.2021.112467.

[47] G. Park, J. Hong, B. A. Duffy, J.-M. Lee, and H. Kim, “White matter hyperintensities segmentation using the ensemble U-Net with multi-scale highlighting foregrounds,” *Neuroimage*, vol. 237, no. April, p. 118140, 2021, doi: 10.1016/j.neuroimage.2021.118140.

[48] “Python Deep Learning: Next generation techniques to revolutionize computer vision, AI, speech and data analysis (English Edition) eBook: Spacagna, Gianmario, Slater, Daniel, Zocca, Valentino, Roelants, Peter: Amazon.fr.” https://www.amazon.fr/Python-Deep-Learning-Gianmario-Spacagna-ebook/dp/B071RCDRKD (accessed Jun. 05, 2021).

[49] B. H. Menze *et al.*, “The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS),” *IEEE Trans. Med. Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015, doi: 10.1109/TMI.2014.2377694.

[50] S. Bakas *et al.*, “Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge,” 2019.

[51] A. Fabijańska, “Segmentation of corneal endothelium images using a U-Net-based convolutional neural network,” *Artif. Intell. Med.*, vol. 88, pp. 1–13, 2018, doi: 10.1016/j.artmed.2018.04.004.

[52] “Understanding Boxplots. The image above is a boxplot. A boxplot… | by Michael Galarnyk | Towards Data Science.” https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51 (accessed Aug. 30, 2021).

[53] “Understanding and interpreting box plots | Wellbeing@School.” https://www.wellbeingatschool.org.nz/information-sheet/understanding-and-interpreting-box-plots (accessed Aug. 30, 2021).

[54] “Measures of the Amount of Ecologic Association Between Species on JSTOR.” https://www.jstor.org/stable/1932409 (accessed Aug. 30, 2021).

References

[55] K. Usman and K. Rajpoot, "Brain tumor classification from multi-modality MRI using wavelets and machine learning," *Pattern Anal. Appl.*, vol. 20, no. 3, pp. 871–881, 2017, doi: 10.1007/s10044-017-0597-8.

[56] F. Pedregosa FABIANPEDREGOSA *et al.*, "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011, Accessed: Aug. 17, 2021. [Online]. Available: http://scikit-learn.sourceforge.net.

[57] "Advantages and Disadvantages of Python Programming Language | by Mindfire Solutions | Medium." https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121 (accessed Aug. 17, 2021).

[58] "13 Unique Features of Python Programming Language - Web Development Tutorial." http://www.webdevelopmenthelp.net/2019/05/13-unique-features-of-python-programming-language.html (accessed Aug. 20, 2021).

[59] "Python Tutorial - w3resource." https://www.w3resource.com/python/python-tutorial.php (accessed Aug. 20, 2021).

[60] "Introduction to the Python Deep Learning Library TensorFlow." https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow/ (accessed Aug. 20, 2021).

[61] "Keras: the Python deep learning API." https://keras.io/ (accessed Aug. 21, 2021).

[62] "Keras backends - Javatpoint." https://www.javatpoint.com/keras-backends (accessed Aug. 24, 2021).

[63] "How to use Glob() function to find files recursively in Python? - GeeksforGeeks." https://www.geeksforgeeks.org/how-to-use-glob-function-to-find-files-recursively-in-python/ (accessed Aug. 19, 2021).

[64] "Page Not Found - GeeksforGeeks." https://www.geeksforgeeks.org/os-module-python-examples/ OS Module in Python with Examples/ (accessed Sep. 06, 2021).

[65] "Python Sys Module - Javatpoint." https://www.javatpoint.com/python-sys-module (accessed Aug. 24, 2021).

[66] "Overview — NumPy v1.21 Manual." https://numpy.org/doc/stable/ (accessed Aug. 22, 2021).

[67] "Overview — Matplotlib 3.4.3 documentation." https://matplotlib.org/stable/contents.html (accessed Aug. 24, 2021).

[68] "Image Processing in Python_Final.ipynb - Colaboratory." https://colab.research.google.com/github/xn2333/OpenCV/blob/master/Image_Processing_in_Python_Final.ipynb#scrollTo=OU4AAstR0HG6 (accessed Aug. 25, 2021).

[69] "Python JSON - GeeksforGeeks." https://www.geeksforgeeks.org/python-json/?ref=lbp (accessed Aug. 24, 2021).