



University of Kerbala  
College of Computer Science & Information Technology  
Computer Science Department

# **AUTOMATED DETECTION AND CLASSIFICATION OF HUMAN BRAIN TUMOR USING MACHINE LEARNING**

A Thesis

Submitted to the Council of the College of Computer Science & Information  
Technology / University of Kerbala in Partial Fulfillment of the Requirements  
for the Master Degree in Computer Science

**Written by**

Sara Ali Abd Al hussen Qamar

**Supervised by**

Assist. Prof. Dr. Elham Mohammed Thabit A. Alsaadi

2024 A.D.

1445 A.H.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ  
(فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَى إِلَيْكَ  
وَحْيُهُ وَقُلْ رَبِّ زِدْنِي عِلْمًا).

صدق الله العلي العظيم  
سورة طه (114)

## Supervisor Certification

I certify that the thesis entitled (**Automated detection and classification of human brain tumor using machine learning**) was prepared under my supervision at the department of Computer Science/College of Computer Science & Information Technology/ University of Kerbala as partial fulfillment of the requirements of the degree of Master in Computer Science.

Signature:



Supervisor Name: Assist. Prof. Dr. Elham Mohammed Thabit A. Alsaadi

Date: 27/ 2 /2024

## The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**Automated detection and classification of human brain tumor using machine learning**" for debate by the examination committee.

Signature:



Assist. Prof. Dr. Muhannad Kamil Abdulhameed

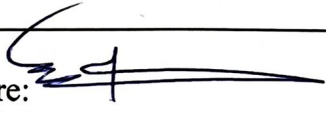
Head of Computer Science Department


Date: 27/ 2 /2024


## Certification of the Examination Committee


We hereby certify that we have studied the dissertation entitled (**Automated detection and classification of human brain tumor using machine learning**) presented by the student (**Sarah Ali Abd Alhussen**) and examined her in its content and what is related to it, and that, in our opinion, it is adequate with (**excellence**) standing as a thesis for the degree of Master in Computer Science.

---


Signature:   
Name: Baheeja Kudhair Shukur  
Title: Prof. Dr.  
Date: 27/ 2 / 2024  
(Chairman)

Signature:   
Name: Soukaena Hassan Hashem  
Title: Prof. Dr.  
Date: 28/ 2 / 2024  
(Member)

Signature:   
Name: Ashwan Anwer Abdulmunem  
Title: Asst. Prof.  
Date: 27/ 2 / 2024  
(Member)

Signature:   
Name: Elham Mohammed Thabit  
Title: Asst. Prof.  
Date: 27/ 2 / 2024  
(Member and Supervisor)

Approved by the Dean of the College of Computer Science & Information Technology, University of Kerbala.

Signature:   
Title: Assist Prof.Dr. Name: Mowafak khadom Mohsen  
Date: 29/ 2 / 2024  
(Dean of Collage of Computer Science & Information Technology)

## **Dedication**

First, I thank God deeply in my heart for facilitating my research. I would also like to dedicate my research to my deceased father, may God have mercy on him. I would also like to express my gratitude to my family who stood by me throughout my educational journey, especially my mother, husband, and children because they were the best support for me. I would also like to thank my sisters and brothers who supported me.

## **Acknowledgement**

First and for most, praises and thanks to Allah, the almighty, for his showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor; **Assist. Prof. Dr. Elham Mohammed Thabit A. Alsaadi**, for her efforts, scientific instructions and good advice, and her keenness to master every small and large, which had a great impact on enriching the message. In addition, show it at this scientific level. Working and studying under her supervision was a great privilege and honor. I am very grateful for what you have given me, may Allah reward her.

I extend my sincere thanks and gratitude to the Dean of the College of Computer Science and Information Technology, and all members of the teaching staff, and the head of the computer department, and all graduate studies staff and my colleagues are graduate students in the College of Computer Science and Information Technology / the University of Karbala

## **Abstract**

The automated system is important for helping doctors and radiologists to detect and classify brain tumors. Solid tumors inside the skull result from uncontrolled and abnormal cell division. The main challenge in detecting brain tumors is the difference in tumor location, shape, size, and the diversity and complexity of brain tumors. Machine learning and Deep learning are the perfect solution to this problem.

The proposed work includes data pre-processing and image segmentation, two segmentation techniques (edge-based segmentation and region-based segmentation) were applied to compare them, where the results using region were better than the results using edge. Morphological operations was applied after the segmentation process and includes the closing processes(dilation, and erosion).

Next, machine learning and deep learning algorithms were applied to classify MRI brain tumors into four types: pituitary, glioma, meningioma, and no tumor in normal cases, based on a specific set of features that improve classification accuracy, save time and cost.

In this work, two proposed models were implemented. The first is the combination Visual Geometry Group 16 (VGG16) with four traditional classifiers: Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Naive Bayes (NB). The combination is implemented due to its deep learning capabilities, as it can extract complex features such as details of brain tumors. The second proposed model is a convolutional neural network (CNN).

In this work, experimental results showed that the combination ((VGG16)with random forest) using region-based segmentation obtained the accuracy of 99.24%. The percentage is higher compared to the

combination ((VGG16)with random forest) when using edge-based segmentation where was the result 98.78%.

This dataset is originally a combination of three datasets: Figshare, the SARTAJ data set, and the Br35H dataset, containing MRI images of the four types of brain tumors was used.

Finally, our results were compared with existing research in the field of segmentation and classification on the same dataset, where our results proved to be the best. Our proposed model achieved an accuracy of 99.24%, while previous research results ranged from 97.12%, 95.73%, to 87.67%.



## **Declaration Associated with this Thesis**

1. S. A. Abd Al Hussen and E. M. T. A. Alsaadi, “Automated Identification and Classification of Brain Tumors Using Hybrid Machine Learning Models and MRI Imaging,” *Ingénierie des systèmes d Inf.*, vol. 28, no. 5, pp. 1299–1308, 2023, doi: 10.18280/isi.280518.
2. S. A. Abd Al Hussen and E. M. T. A. Alsaadi , “Machine learning for detection and classification of human brain tumor: A survey,”, 2023 International Conference on Information Technology Applied Mathematics Statistics (ICITAMS), 2023.

# Table of Contents

Dedication .....	i
Acknowledgement .....	ii
Abstract .....	iii
Declaration Associated with this Thesis.....	v
Table of Contents.....	vi
List of Tables .....	x
List of Figures.....	xii
List of Algorithms.....	xv
List of Abbreviations.....	xvi
<b>CHAPTER ONE: INTRODUCTION.....</b>	<b>1</b>
1.1 Overview .....	2
1.2 Problem statement.....	4
1.3 The aim of the thesis.....	5
1.4 Thesis Organization .....	6
<b>CHAPTER TWO: THEORETICAL BACKGROUND.....</b>	<b>7</b>
2.1 Overview .....	8
2.2 Brain Tumor Disease .....	8
2.2.1 Types of Brain Tumors.....	8
2.3 Imaging Techniques.....	11
2.4 Image Enhancement.....	11
2.4.1 Resize.....	11
2.4.2 Brightness.....	12
2.4.3 Contrast.....	12
2.4.3.1 Contrast-Limited Adaptive Histogram Equalization (CLAHE).....	12
2.4.4 Normalization.....	14
2.4.5 Sharpness.....	14
2.4.6 Segmentation Techniques.....	15
2.4.6.1 Edge-Based Segmentation.....	16
2.4.6.1.1 Canny Edge Detection.....	16

2.4.6.2 Region-Based Segmentation.....	18
2.4.6.2.1 Region Splitting and Merging.....	19
2.4.6.2.2 Region Growing .....	19
2.5 Morphology Operation .....	20
2.5.1 Dilation Operation.....	20
2.5.2 Erosion Operation.....	21
2.6 Machine Learning.....	22
2.6.1 Support Vector Machine (SVM).....	24
2.6.2 Decision Tree (DT).....	26
2.6.3 Random Forest(RF).....	28
2.6.4 Naïve Bayes(NB).....	30
2.7 Neural Networks.....	32
2.8 Deep Learning.....	34
2.8.1 New Train.....	34
2.8.1.1 Convolution Neural Network.....	34
2.8.2 Pre-trained.....	38
2.8.2.1 Visual Geometry Group 16 (VGG16).....	39
2.9 Transfer Learning.....	40
2.10 Activation Function types.....	40
2.10.1 Relu.....	40
2.10.2 Sigmoid.....	41
2.10.3 SoftMax.....	41
2.11 Loss Function.....	42
2.11.1 Binary Cross-Entropy .....	42
2.11.2 Categorical Cross-Entropy.....	43
2.12 Optimizer.....	43
2.12.1 Adaptive momentum estimation (Adam) .....	44
2.13 Evaluation Matrices.....	45
2.13.1 Confusion Matrix.....	45
2.13.2 Accuracy.....	46
2.13.3 Sensitivity(Recall).....	47
2.13.4 Precision.....	47

2.13.5 F1score.....	47
2.14 Related Work.....	47
<b>CHAPTER Three: PROPOSED METHODOLOGY.....</b>	<b>54</b>
3.1 Overview .....	55
3.2 Proposed Models.....	55
3.2.1 Combination Model .....	57
3.2.2 CNN Model.....	58
3.3 Splitting Dataset.....	60
3.4 Preprocessing.....	60
3.5 Segmentation for Brain Tumor.....	61
3.5.1 Segmentation for brain tumor using region based segmentation.....	61
3.5.2 Segmentation for brain tumor using edge based segmentation.....	62
3.6 Morphology Operation.....	63
3.7 Feature Extraction.....	65
3.8 VGG16.....	65
3.9 Classification for Brain Tumor.....	66
3.9.1 Classification Using Combination Model .....	66
3.9.2 Classification Using CNN Model.....	67
<b>CHAPTER FOUR: RESULTS AND DISCUSSION .....</b>	<b>71</b>
4.1 Overview .....	72
4.2 Experimental Setting .....	72
4.3 Dataset Description .....	72
4.4 Experimental Result.....	73
4.4.1 Experiment of Region-Based Segmentation.....	73
4.4.2 Experiment of Edge-Based Segmentation.....	74
4.5 Results and Discussion.....	75
4.5.1 Results of Combination Model .....	76
4.5.1.1 Results of Combination Model Using Region-Based Segmentation and Edge-Based Segmentation.....	76

4.5.2 Results of CNN System.....	84
4.5.2.1 Result of CNN System Using Region-Based Segmentation and Edge-Based Segmentation.....	84
4.6 Performance Comparison between Traditional Machine Learning Classifiers in combination model (VGG16 with ML Classifiers).....	88
4.7 Performance Comparison between Combination Model and CNN Model.....	89
4.8 Comparison with the most modern models .....	90
<b>CHAPTER FIVE: CONCLUSION AND FUTURE WORKS .....</b>	<b>92</b>
5.1 Overview.....	93
5.2 Conclusion.....	93
5.3 Future Work.....	94
<b>REFERENCES .....</b>	<b>95</b>

## List of Tables

Table 2. 1: The most commonly used fully connected layer activation functions are listed below.....	41
Table 2.2: Confusion Matrix.....	46
Table 2.3: Summary of related work.....	51
Table 3.1: Accuracy comparison between traditional machine learning classifiers using different Threshold.....	62
Table 4.1: The performance evaluation of Combination model (VGG16 with SVM) using region-based segmentation.....	76
Table 4.2: The performance evaluation of Combination model (VGG16 with SVM) using edge- based segmentation.....	76
Table 4.3: The performance evaluation of Combination model (VGG16 with DT) using region-based segmentation.....	78
Table 4.4: The performance evaluation of Combination model (VGG16 with DT) using region- based segmentation.....	78
Table 4.5: The performance evaluation of Combination model (VGG16 with RF) using region- based segmentation .....	80
Table 4.6: The performance evaluation of Combination model (VGG16 with RF) using edge - based segmentation.....	80
Table 4.7: The performance evaluation of Combination model (VGG16 with NB) using region- based segmentation.....	82
Table 4.8: The performance evaluation of Combination model (VGG16 with NB) using edge – based segmentation.....	82
Table 4.9: Accuracy of the CNN model with region-based segmentation and Edge-Based Segmentation .....	84
Table 4.10: The performance evaluation of CNN model using region-based segmentation.....	85
Table 4.11: performance evaluation of CNN model using edge-based segmentation.....	85
Table 4.12: : Accuracy comparison among traditional machine learning classifiers in	

combination model (VGG16 with ML Classifiers) by using region-based segmentation.....	88
Table 4.13: Accuracy comparison among traditional machine learning classifiers in combination model(VGG16 with ML Classifiers) by using edge-based segmentation.....	89
Table 4.14: Performance comparison between the Combination model and CNN model using region-based segmentation .....	89
Table 4.15: Performance comparison between the Combination model and CNN model using edge-based segmentation .....	90
Table 4.16: Comparison of proposed model and related research .....	90

## List of Figures

Figure 1.1: Examples of brain tumor variation.....	5
Figure 2.1: (a) Benign brain tumor, (b) Malignant brain tumor .....	9
Figure 2.2: Secondary brain tumor.....	10
Figure 2.3: Segmentation techniques.....	16
Figure 2.4: Example canny edge detection.....	16
Figure 2.5: Dilation Operation.....	21
Figure 2.6: Erosion Operation.....	22
Figure 2.7:SVM (a) linear separation and (b) Margin (c) SVM non-linear separation.....	25
Figure 2.8: Internal working of Decision Tree.....	27
Figure 2.9: Internal working of Random Forest.....	30
Figure 2.10: Figure (a) A Naive Bayes Classifier, where C represents the class node and X represents the feature node; (b) A simplified Graphical Model of Naive Bayes Classifier; (c)example Naive Bayes Classifier.....	32
Figure 2.11: A simple Artificial Neural Network.....	33
Figure 2.12: Typical architecture of convolutional neural network.....	34
Figure 2.13: Convolutional operation showed schematically. In order to fill the pixels in the destination layer, the convolutional kernel shifts over the source layer .....	35
Figure 2.14: Displaying pooling activities visually, including maximum and average pools.....	36
Figure 2. 15: Architecture of Fully Connected Layers.....	37
Figure 2.16: Structure of VGG-16 model.....	39
Figure 3.1: Block diagram of two proposed Models.....	67
Figure 3.2: Combination Model (VGG16+Machine Learning	



Algorithms).....	60
Figure 3.3: Architecture of the CNN Models.....	69
Figure 4.1: Sample for dataset.....	73
Figure 4.2: Displays an example segmented image using region-based segmentation.....	74
Figure 4.3: Displays an example segmented image using edge-based segmentation.....	75
Figure 4.4: confusion_ matrix of Combination model (VGG16 with SVM) using region-based segmentation .....	77
Figure 4.5: confusion_ matrix of Combination model (VGG16 with SVM) using edge-based segmentation .....	77
Figure 4.6: confusion_ matrix of Combination model (VGG16 with DT) using region-based segmentation .....	79
Figure 4.7: confusion_ matrix of Combination model (VGG16 with DT) using edge-based segmentation .....	79
Figure 4.8: confusion_ matrix of Combination model (VGG16 with RF) using region-based segmentation .....	81
Figure 4.9: confusion_ matrix of Combination model (VGG16 with RF) using edge-based segmentation .....	81
Figure 4.10: confusion_ matrix of Combination model (VGG16 with NB) using region-based segmentation .....	83
Figure 4.11: confusion_ matrix of Combination model (VGG16 with NB) using edge-based segmentation .....	83
Figure 4.12: Accuracy function of the CNN model using region-based segmentation .....	86
Figure 4.13: Loss function of the CNN model using region-based segmentation .....	86
Figure 4.14: Accuracy function of the CNN model using edge-based	

segmentation .....	86
Figure 4.15: Loss function of the CNN model using edge-based segmentation .....	86
Figure 4.16: Confusion_Matrix of CNN using region-based segmentation.....	87
Figure 4.17: Confusion_Matrix of CNN using edge-based segmentation.....	87

## List of Algorithms

Algorithm 2.1: Support Vector Machine .....	24
Algorithm 2.2: Decision Tree.....	28
Algorithm 2.3: Random Forest.....	29
Algorithm 2.4: Naive Bayes.....	31
Algorithm 3. 1: Combination Model.....	57
Algorithm 3. 2: CNN Model.....	58
Algorithm 3. 3: pre-processing .....	60
Algorithm 3. 4: Region Split and Merge Segmentation.....	61
Algorithm 3. 5: Canny edge detection for segmentation.....	63
Algorithm 3. 6: Morphological Operations .....	64

## List of Abbreviations

<b>Abbreviation</b>	<b>Description</b>
Adam	Adaptive Moment Estimation
BT	Brain Tumors
CLAHE	Contrast-Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
FC	Fully Connected
FN	False Negative
FP	False Positive
HSV	Hue Saturation Value
ML	Machine Learning
MRI	Magnetic Resonance Imaging
NB	Naive Byes
RELU	Rectified Linear Unit
RF	Random Forest
SVM	Support Vector Machine
TL	Transfer Learning
TN	True Negative
TP	True Positive
VGG16	Visual Geometry Group 16

**CHAPTER ONE**  
**INTRODUCTION**

## 1.1 Overview

The rising technology in image-processing medicine is drawing a lot of attention to brain tumors and their examination. The brain which has billions of cells and functions as one of the numerous organs that make up the human body is the most vital of all of them. One of the common reasons for brain dysfunction is a brain tumor. A tumor is nothing more than developing into aberrant cells, which are a group of cells in or around the brain that can harm healthy cells and impede normal brain activity. Brain failure is caused by the gradual depletion of all nutrients given to healthy cells and tissues. Brain tumors can be highly cancerous (malignant) or low-grade non-cancerous (benign). Benign tumors, which start in the brain and develop slowly, are not cancerous and do not progress. They are believed to be less aggressive. In addition, it is unable to spread throughout the body. On the other hand, malignant tumors have a never-ending growth cycle. Primary tumors are tumors that develop in the brain itself, while secondary tumors are tumors that begin elsewhere in the body and progress to the brain [1].

The traditional way to detect a tumor in an MRI image is a human examination. This process takes a long time. For a lot of data, it is not appropriate. Additionally, operator noise in an inaccurate classification may come from an MRI. Automated systems are necessary due to how much MRI data is being evaluated because they are more effective. Given that dealing with human life necessitates great resolution,

automated tumor detection in MRI images is crucial [2]. Early detection and prompt treatment unquestionably boost the chance of surviving a brain tumor[3].

Brain MRI is mostly used to track tumor development. The principal applications of this knowledge are in the diagnosis and management of cancers. Magnetic Resonance Imaging (MRI) offers more information on a particular medical imaging than a Computed Tomography (CT) or ultrasound image. The MRI image identifies anomalies in brain tissue and offers extensive information on the structure of the brain [4]. The segmentation of brain tumors in MRI images is one of the most difficult tasks in medical image processing, the borders of the tumors may be poorly delineated by soft tissue. Consequently, obtaining a correct segmentation of tumors from the human brain is a highly challenging mission [5].

Based on the mechanism of feature selection and learning, many strategies for the automatic classification of brain tumors have been recently developed. These technologies can be divided into machine learning (ML) and deep learning (DL) technologies. Feature selection and extraction are crucial for the classification of machine learning techniques. However, deep learning methods directly extract and detect image features. In particular, CNN, a high-resolution (DL) technique, is often used in medical image analysis. Moreover, machine learning can work with a small dataset, while deep learning works with a huge training dataset and requires expensive GPUs. In addition, choosing the right deep learning tool can be difficult because it requires

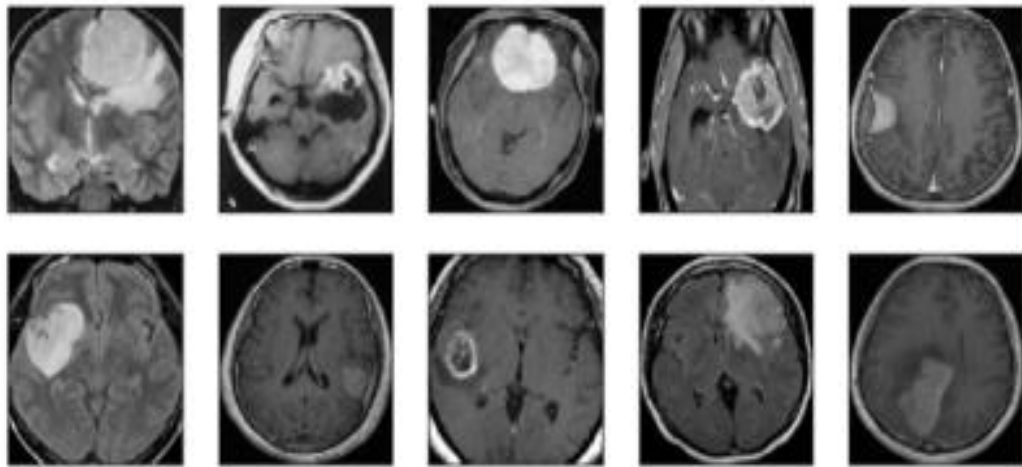
understanding many parameters, training strategies, and topology. However, in the field of medical imaging, machine learning and deep learning techniques are very important. Support Vector Machine (SVM), Artificial Neural Network (ANN), Sequential Minimum Optimization (SMO), Fuzzy C-Means (FCM), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT) and K-Nearest Neighbor (KNN) are some learning-based classifiers used for the classification and detection of brain tumors [6].

## 1.2 Problem Statement

The problem statement of this work includes the following:

1. Inaccurate detection of the brain tumor is due to the physicians locate a brain tumor at MRI of a patient's brain manually, thereby adversely affecting the accuracy of the detection, this also takes time consuming.
2. Due to the complexity of brain structure, tumor segmentation is a really challenge tasks.
3. The main difficulty is detecting brain tumors in variations of tumor location, shape, size, and intensities from patient to patient, also tumor boundaries are usually unclear and irregular. Example variations of brain tumors are shown in Figure (1.1) [7].





*Figure 1.1: Examples of brain tumor variation[7]*

### 1.3 The Aim of the Thesis

This work will model design to detect and classify brain tumor.

1. Tumor detection using segmentation strategies (region-based segmentation, edge-based segmentation) in MRI images of patients' damaged brains.
2. Tumors classification into four categories for human brain tumors (pituitary, meningioma, glioma, or no tumor in normal cases), two proposed models were implemented. The first is the combination (VGG16) with four traditional classifiers: Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Naive Bayes (NB). The second proposed model is a convolutional neural network (CNN).
3. To use it as a tool to assist clinicians and serve as a decision-support tool for classifying brain tumors. The goal of this work is to help implement appropriate preventive measures and make future diagnoses easier and more accurate.

## 1.4 Thesis Organization

The five Chapters of this thesis are as follows:

**Chapter One:** Provides a general overview of brain tumor detection, presents the problem statement and goal of this work.

**Chapter Two:** Explains segmentation, classification, machine learning techniques, and the use of metrics to evaluate the system, providing the theoretical underpinnings of the performance model, and related work.

**Chapter Three:** Describe the proposed technique for automatic detection and diagnosis of brain tumors in human MRI images.

**Chapter Four:** Introduces the experimental implementation and the results of this work.

**Chapter Five:** The conclusions of the work are shown and future work is discussed.

## 1.5 Summary

An overview of brain tumor, the problem description, the Aim of the Thesis, and thesis organization are included in this chapter.

**CHAPTER TWO**

**THEORETICAL BACKGROUND**

## 2.1 Overview

This chapter explains segmentation and classification of brain tumor. Additionally, it illustrates how the model's representation of the deep learning technique and the machine learning idea works. A design model approach, software testing and metrics are presented. Finally, related works are presented.

## 2.2 Brain Tumor Disease

Uncontrolled and abnormal cell division in the brain can lead to brain tumors. It exists in various parts such as neurons, glial cells, lymphoid tissue, pituitary gland, and skull. Cancer cells found in various organs can spread the tumor. Cancer is currently the most common cause of death, brain tumors may contribute to their development [8].

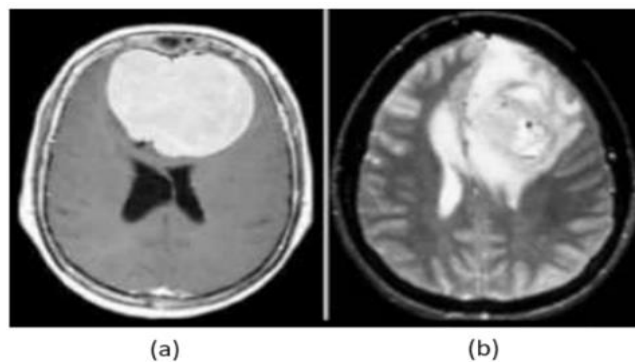
Due to the tumor's variable nature and co-resemblance with other brain regions, analyzing a tumor to identify malignant characteristic is a challenging task [9]. Many bodily functions, including speech, balance, breathing, movement, and heart function are impacted by brain tumors, because it affects healthy cells. Early detection of brain tumors is essential to understanding the condition and providing effective therapy [8].

### 2.2.1 Types of Brain Tumors

There are two main forms of brain tumors:

1. Primary brain tumor: This kind of tumor does not spread and develops right adjacent to brain tissue. These tumors come in both benign and malignant forms.

- Benign: These tumors are characterized by slow growth and sharp borders.
- Malignant: This category of tumor grows quickly and has an erratic border [9]. It propagates to surrounding areas of the brain. Figure (2.1) shows examples of benign and malignant brain tumors.



*Figure 2.1: (a) Benign brain tumors, (b) Malignant brain tumors [9]*

2. Secondary brain tumor: These tumors spread from other parts of the body to the brain where they first appear. This kind develops as a result of the bloodstream carrying malignant cells. More than 120 different forms of brain tumors exist [8].

In this work, it was performed on secondary brain tumor type.

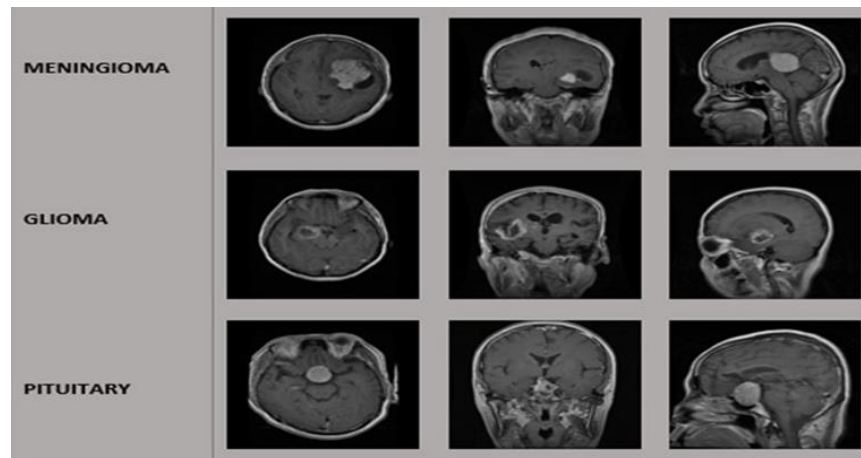


Figure 2.2: Secondary brain tumor[10]

The following includes some brain tumors with names based on the tissues they involve.

- Gliomas: Gliomas are tumors that develop in glial cells and affect the brain and spinal cord. It's crucial to classify this sort of tumor to assess how dangerous it is and then take the appropriate treatment measures[8].
- Meningioma: It is a tumor that develops from the membranes surrounding the brain, spinal cord, and meninges. The classification of this sort of tumor can assist in identifying the type of tumor and how much it affects nerves and important functions because it pressures on the nearby brain, nerves, and veins [8].
- Pituitary: Tumors are abnormal growths that develop in your pituitary gland; classification of this type of tumor can help determine its effects on hormones and bodily functions [8].
- No tumor: This category means that there is no tumor in the submitted image. This category may be important to ensure the accuracy of a negative classification and to ensure that no normal changes are converted into a misdiagnosis of tumors [8].

## 2.3 Imaging Techniques

There is a variety of imaging techniques available, including:

- X-rays: Create images using radiation. In the movie, when the X-rays move through the body, solid materials like bones appear white. These are frequently used to examine and detect infections, bone diseases, fractures, and cracks in the bone [11].
- CT scan: This procedure creates detailed, high quality body images. It creates precise images of the soft tissue, bones, organs, and blood vessels. It has greater power than X-rays [11].
- MRI: An MRI machine uses a powerful magnet to excite and polarize hydrogen nuclei. which allows the physical properties of the proton to be used to build a digital depiction of the brain tissue [11].

The MRI scan is utilized to thoroughly examine various bodily regions and also aids in the early detection of brain disorders than other imaging modalities [8].

## 2.4 Enhancement Image

It is a crucial stage in the processing of digital images. The fundamental idea behind image enhancement is to draw attention to the important parts of an image. To reveal of the image's details. The modification of brightness and contrast is the fundamental process used in image enhancement [12].

### 2.4.1 Resize

The dataset consists of thousands of images of different sizes. Taking into account the input of the image, the size of each image varies

in height and width, in this case will set a fixed size. In order to standardize the image sizes. All images in the group are resized to the appropriate size [13].

### **2.4.2 Brightness**

Brightness in an image refers to the overall lightness or darkness of the image. It determines how much light is present in the image as perceived by the human eye. Increasing the brightness makes the image appear brighter, while decreasing it makes the image appear darker. Pixel values in an image are directly related to brightness. You can change the pixel values based on the brightness of the image. There are several methods for adjusting image brightness: Addition/Subtraction, Multiplication/Division, and Gamma Correction [14].

### **2.4.3 Contrast**

Measure the image spatial frequency and Gray-Level Co-occurrence Matrix (GLCM) divergence moment. It is the variation in contrast between adjacent pixels that are at their highest and lowest levels. It measures how much local variation there is in the image. Contrast images are enhanced using Contrast-Limited Adaptive Histogram Equalization (CLAHE) [15].

#### **2.4.3.1 Contrast-Limited Adaptive Histogram Equalization (CLAHE)**

Adaptive graph equations are a generalization called CLAHE. The CLAHE algorithm partitions the input image into non-overlapping context regions (also known as sub images, boxes, and blocks). Then,



equalize the histogram for each context region, crop the original histogram to a specific value, and redistribute the cropped pixels to each gray level [16]. The steps in creating a histogram are: creating the histogram matrix; computing the Cumulative Distribution Function (CDF); and applying the intensity transformation to each pixel's intensity.

The CLAHE equation for boosting the pixel values in a specific tile can be written as:

$$H(V) = \text{round} \left( \frac{CDF(V) - CDF_{min}}{(M*N) - CDF_{min}} * (L - 1) \right) \quad (2.1)$$

H: is the histogram

V: is the intensity

M\*N: is the image matrix

L: no. of gray

CDF: is the cumulative distribution function

$$CDF_k = \sum_{v=-\infty}^v \frac{H_v}{M*N} \quad (2.2)$$

Where:

CDF (k): represents the cumulative distribution function at intensity level k

$H_{(v)}$ : represents the histogram value at intensity level v.

M: indicates how many columns there are in the image.

N: shows how many rows there are in the image.

If the CDF (k) exceeds the clip limit, the excess value is redistributed to other bins in the histogram, the clip limit is determined based on the experiment until satisfactory results are achieved. And then, the

histogram is equalized for each tile using the CDF (k) values. The pixel values are adjusted based on the CDF values, effectively spreading out the intensities and enhancing the contrast within each tile [16].

#### 2.4.4 Normalization

These images include significant values outside the range [0, 255], including negative values, as a result of reading the input brain images. By dividing the value of each pixel by the highest pixel value (255), which can be calculated using the min-max normalization equation (2.3), this step converts the brain image into one with an intensity range of [0, 1][17].

$$f(x, y) = \frac{f(x,y)-Vmin}{Vmax-Vmin} \quad (2.3)$$

The minimum and maximum values in image are represented by  $Vmin$  and  $Vmax$  respectively, where  $f(x,y)$  represents each pixel in the brain image.

#### 2.4.5 Sharpness

Edges are sharpened, and visual blur is eliminated. Spatial discrimination is the basis for sharpening filters. Among them are Laplace filters, Sobel filters, difference filters.

1. The Laplacian filter is a derivative filter used to identify edges in images that undergo rapid changes. The Laplacian operator takes partial derivatives along of an image  $f(x, y)$ . highlights sharp intensity

transitions and reduces the effect of regions having slowly varying gray levels, resulting in the following  $3 \times 3$  Laplacian filters[18].

$$k = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.4)$$

2. To detect edges, Sobel filters are generally used by converging image with  $K_x$  and  $K_y$  Sobel kernels, respectively[19].

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.6)$$

3. Different filters highlight details in a way that is distinctive to the chosen mask [19].

### 2.4.6 Segmentation Techniques

In segmentation, an image is divided into several regions. These regions contain the same qualities in terms of texture, color, intensity, contrast, and gray level. Therefore, the primary goal of segmentation is to division of the items present in an image that are connected in some way. Segmentation methods include threshold- based segmentation, edge- based segmentation, clustering- based segmentation, and region-based segmentation [20]. In our work, two techniques were used: edge-based segmentation and region-based segmentation.

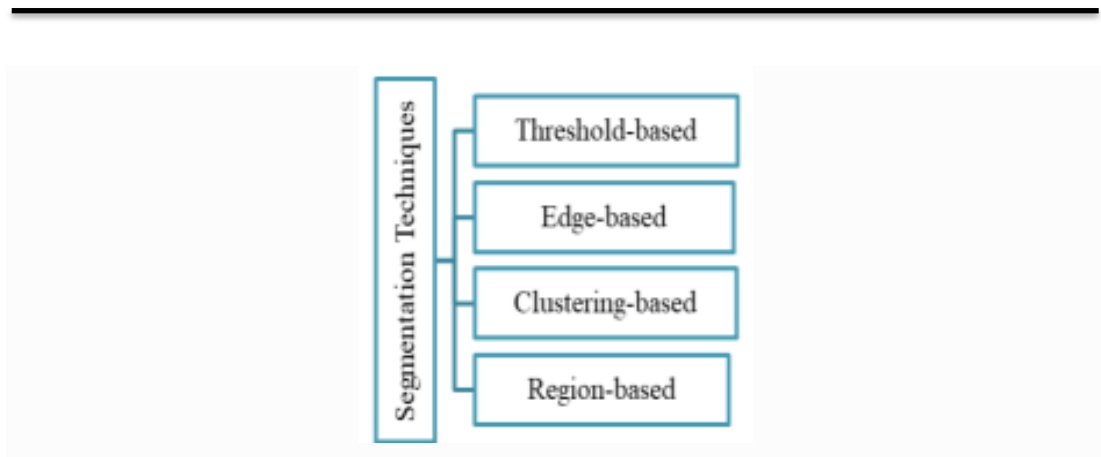


Figure 2.3: Segmentation techniques[20]

### 2.4.6.1 Edge-Based Segmentation

This method divided the image based on a abrupt changes in the brightness of the pixels that are close to the edges. This method results in the creation of a binary image. The components' boundaries can be seen in this image [21].

#### 2.4.6.2.1 Canny Edge Detection

The canny edge detection algorithm consists of five steps [22]:

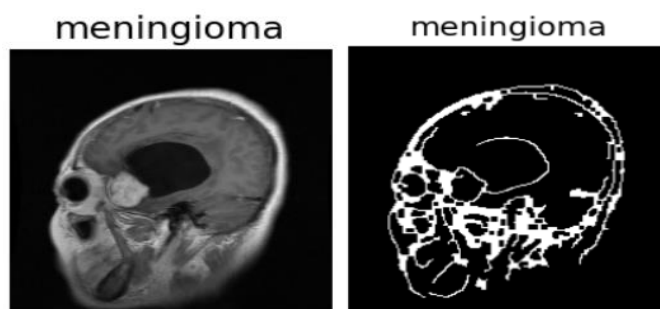


Figure 2.4: Example canny edge detection

#### : 1. Noise reduction

Applying Gaussian Blur to an image to smooth it is a way to remove noise from an image. A Gaussian kernel with image convolution technique is used to achieve this equation (2.7) is used to express it [22].

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (2.7)$$

Where  $y$  is the distance from the origin in the vertical axis,  $x$  is the distance from the origin in the horizontal axis and  $\sigma$  is the standard deviation of the Gaussian distribution.

### 3. Gradient Calculation

The gradient computation step determines the strength and direction of edges by computing image gradients using edge detection operators. Filters were applied that emphasize this intensity contrast in both the horizontal ( $x$ ) and vertical ( $y$ ) axes. It can be used by converging image with  $K_x$  and  $K_y$  Sobel kernels, respectively [23].

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.9)$$

The gradient's magnitude  $G$  and slope are then determined as follows:

$$|G| = \sqrt{I_x^2 + I_y^2} \quad (2.10)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{y}{x}\right) \quad (2.11)$$

#### 4. Non-Maximum Suppression

The Final image should have sharp edges. To straighten edges, use Maximum Suppression. The algorithm traverses all points of the gradient intensity matrix and selects the pixel with the highest edge direction value [22].

#### 5. Double Threshold

Strong pixels, weak pixels and irrelevant pixels are the three types of pixels to be identified in the dual threshold step. Use a high threshold to identify strong pixels (intensities greater than the high threshold). Use the lower threshold to identify irrelevant pixels (pixels with intensity below the lower threshold).

All pixels whose intensity is between the two thresholds are marked as weak, and the hysteresis process helps us distinguish those pixels that can be considered strong from those that are relevant in the next stage [23].

#### 6. Edge Tracking by Hysteresis

Hysteresis involves converting a weak pixel into a strong pixel based on a threshold result if at least one pixel around the processed pixel is a strong pixel [22].

##### 2.4.6.2 Region-Based Segmentation

It is a method for accurately identifying and locating the desired location. It combines the individual pixels in an input image into groupings of pixels called regions, which could be associated with an

item or a significant portion of one. Based on the similarity between the pixels in the given MRI, Tanuja and Subhangi developed a technique to separate malignancies [24].

#### **2.4.6.2.1 Region Splitting and Merging**

The split method begins with the entire image, and repeatedly splits each segment into quarters if the homogeneity criterion is not satisfied. These splits can sometimes divide portions of one object. The merge method joins adjacent segments of the same object. It is important to distinguish the separate regions for intensity based segmentation so that over segmentation and under-segmentation of regions can be differentiated. Task of this kind can be performed using split segmentation or merge segmentation. If a region is not segmented fully, correction can be made by adding boundaries to, or splitting, certain regions that contain parts of different objects. If a region is segmented more than is necessary, correction can be made by eliminating false boundaries and merging adjacent regions if they belong to the same object or feature[25].

In this work, region splitting and merging techniques were employed.

#### **2.4.6.2.3 Region Growing**

Region growing connects neighboring points to make bigger region. The process of region growing is dictated by certain condition associated with the selection of a threshold value. Seeded region growing starts with one or more seed points and then grows within the region to form a larger region satisfying some homogeneity constraint.

The homogeneity of a region can be dependent upon any characteristic of the region in the image: texture, color or average intensity[25].

## 2.5 Morphology Operation

The focus is on pattern analysis and extraction of boundary regions from images. The reordering of pixel values is called a morphological process.

- **Opening and closing operate** on features smaller than the width of the structuring element.

- ❖ Opening is defined as erosion followed by dilation within the same structuring element. The opening operation removes bright features (e.g. peaks and ridges) [26].

- ❖ Closing is dual to opening, i.e. dilation followed by erosion within the same structuring element. The closing operation fills in dark regions (e.g. troughs, holes and gaps) [26].

### 2.5.1 Dilation Operation

The dilation determines the maximum value by comparing all pixel values near the input image[27]. It is just the opposite of erosion. Here, a pixel is '1' if at least one pixel below the kernel is '1'. So the white area in the image increases or the size of the foreground object increases. Typically, in cases such as noise removal, corrosion is followed by dilation. Because abrasion removes white noise, but it also causes the body to shrink and expand the area. Since the noise has disappeared, it will not return, but the area of the object increases. It is also useful in binding broken parts of the body[26].



The following equations explain the morphological operations for tumor areas detection. The dilation of A by B, denoted  $A \oplus B$ , is defined as[28]:

$$A \oplus B = \{Z \mid (\hat{B})_Z \cap A \neq \Phi\} \quad (2.12)$$

Where  $\Phi$  is the empty set and B is the structure element. In words, the dilation of A by B is the set consisting of all the structure element origin locations where the reflected and translated B overlaps at least some portion of A[28][27]. Figure 2.5 illustrates how dilation works.

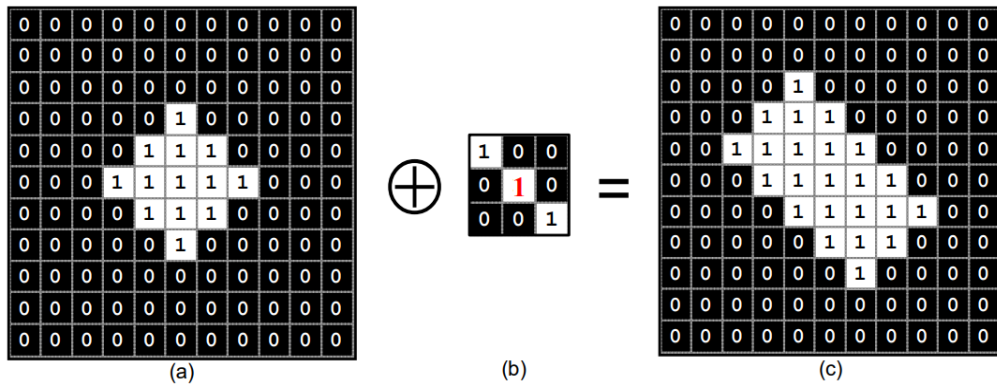


Figure 2.5: Dilation Operation

### 2.5.2 Erosion Operation

The erosion determines the minimum value by compare all pixel values near the input image[27]. The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object (Always try to keep foreground in white).The kernel slides through the image (as in 2D convolution). A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded (made to zero) . So all the pixels near boundary will be discarded depending upon the size of kernel[26].

It is stated that the morphological function has eroded as follows [28]:

$$A \ominus B = \{Z \mid (B)_Z \cap A^c \neq \Phi\} \quad (2.13)$$

In other words, erosion of A by B is the set of all structure element origin locations where the translated B has no overlap with the background of A. Algorithmically we can define erosion as: the output image  $A \ominus B$  is set to zero. B is placed at every black point in A. If A contains B (that is, if  $A \text{ AND } B$  is not equal to zero) then B is placed in the output image. The output image is the set of all elements for which B translated to every point in A is contained in A [28]. Figure (2.6) illustrates how erosion works.

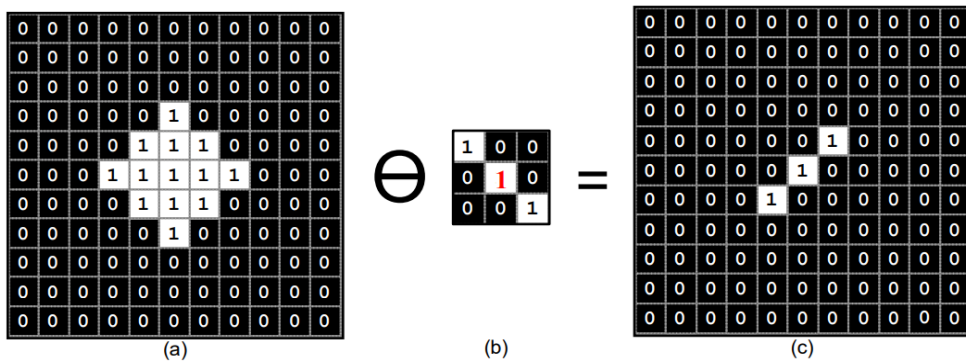


Figure 2.6: Erosion Operation

## 2.6 Machine Learning

The scientific field of (ML) enables computers to "learn" from data. To help and/or support decision and predictions, algorithms are employed to identify natural patterns in data. Thinking about the objective and the type of data [29].

(ML) is used to train computers to process data more efficiently. When looking at data, the information is not understood as to what it contains, so machine learning is applied. The need for machine learning has increased due to the accessibility of huge data sets. Many sectors employ

machine learning to collect persistent data. (ML) aims to get knowledge from data. Many mathematicians and programmers use various techniques to leverage large amounts of data to solve this problem. Algorithms can be categorized into different types based on the learning model, such as supervised learning, unsupervised learning, and reinforcement learning[30].

### **1. Supervised Learning**

It is trained based on the given input and its expected output, i.e., the label of the input. The model creates a mapping equation based on the inputs and outputs and predicts the label of the inputs in the future based on that mapping equation[31].

### **2. Unsupervised Learning**

It is trained only on the inputs, without their labels. The model classifies the input data into classes that have similar features. The label of the input is then predicted in the future based on the similarity of its features with one of the classes[31].

### **3. Reinforcement Learning**

It is taking the best possible action in a given situation to maximize the total profit. The model learns by getting feedback on its past outcomes[31].

The following are examples of machine learning algorithms, which focus more on classification: Decision trees, Random Forests, Neural Networks, Naive Bayes, Support Vector Machines, and other algorithms [32].

### 2.6.1 Support Vector Machine (SVM)

SVM is a powerful classifier used for classification tasks (binary and multiclass) and is characterized by its ability to handle complex data sets and separate them into different classes. SVM overhauls the level of class separation by creating margins between data of different classes. SVM can handle complex classification problems and is able to handle nonlinear data using space transformations. Algorithm (2.1) illustrate Support Vector Machine[33].

<b>Algorithm 2.1 Support Vector Machine</b>
<p><b>Input:</b> Determine the various training and testing data  <b>Output:</b> Predicated Class <math>Y</math></p>
<p><b>Begin</b>  candidateSV = {closest pair from opposite classes}  while there are violating points do      Find a violator      candidateSV = candidateSV U violator      <b>if</b> any <math>\alpha p &lt; 0</math> due to addition of <math>c</math> to <math>S</math> <b>then</b>          candidateSV = candidateSV \p          repeat till all such points are pruned      <b>end if</b>  <b>End</b></p>

SVM is originally designed for binary classification, it can be adapted to handle multiclass classification using various strategies such as one-vs-all, one-vs-one.

#### 1. One-vs-All (OvA) Approach:

- For each class in the dataset, a separate binary SVM classifier is trained to distinguish that class from all other classes combined.

- During prediction, each classifier predicts the likelihood of the instance belonging to its corresponding class.
- The class with the highest predicted likelihood becomes the predicted class for the instance.

2. One-vs-One (OvO) Approach:

- In this approach, instead of training a single classifier for each class, a binary classifier is trained for every pair of classes.
- During prediction, each classifier votes for its predicted class.
- The class with the most votes is chosen as the predicted class for the instance.

It creates a hyperplane in the infinitely large space of dimensions. The hyperplane aids in achieving the greatest distance from any training data point to the nearest of any class. With a greater functional margin, the generalization error can be reduced. SVM use the kernel trick to do this[34]. Figure (2.7) (a, b, c) shows the linear and nonlinear separation with margin.

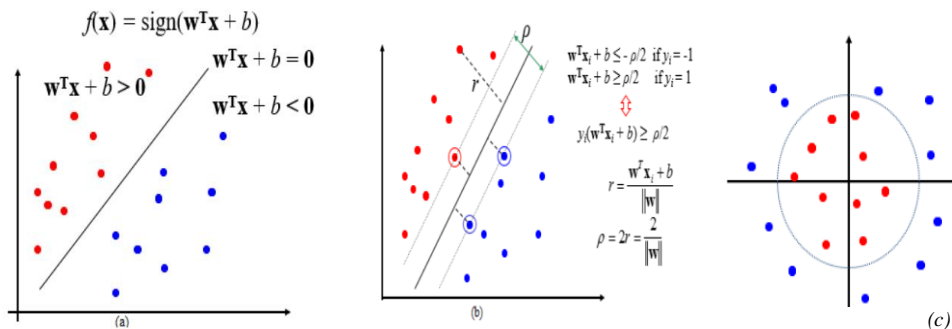


Figure 2.7: (a) SVM linear separation and (b) Margin(c) SVM non-linear separation[34]

Consider a hyperplane defined where  $w$  is its normal by

$$x \cdot w^T + b = 0 \quad (2.14)$$

The information is labeled and linearly divided[34]:

$$\{x_i, y_i\}, x_i \in R, y_i \in \{-1, 1\}, i = 1, 2, \dots, N \quad (2.15)$$

In this case,  $y_i$  is Class naming for two SVM classes. The objective function is minimized by a maximum margin.

$$x_i \cdot w^T + b \geq 1 \quad \text{for } y_i = +1 \quad (2.16)$$

$$x_i \cdot w^T + b \leq -1 \quad \text{for } y_i = -1 \quad (2.17)$$

$x_i$ : input,  $w^T$ : weight transport,  $b$ : bias.

### 2.6.2 Decision Tree (DT)

The decision tree is constructed starting at the root and working down to the sub-nodes. The information is categorized as a result of the division of the nodes by the predicted value of the class, which reflects characteristic features that represent decision points. The nodes are connected at various levels by branches that reflect various choices made by evaluating the state of the node's attributes. It applies to data with intricate architecture. It can lead to tree fusion, which causes a weakening of the classifier. A supervised machine learning approach known as a decision tree operates by combining an if/else expression with a tree data structure.

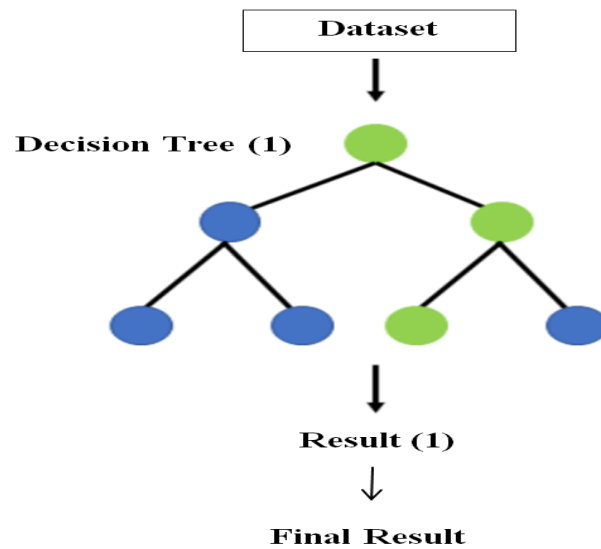


Figure 2.8: Internal working of Decision Tree [23]

Shannon entropy, a measure of how much information an event contains, is the foundation of the decision tree method. The information carried by the probability distribution  $P$ , expressed as  $P = (p_1, p_2, p_3, \dots, p_n)$ , with data  $S$  as a sample, called entropy  $P$ , expressed by the equation below[35].

$$Entropy(X) = -\sum_{i=1}^n P(x) \log(P(x)) \quad (2.18)$$

Where:

$Entropy(X)$  : is the entropy of the random variable  $X$ .

$x$  : represents each possible outcome of the random variable.

$$Gain(P, T) = Entropy(p) - \sum_{j=1}^n Entropy(P, T) \quad (2.19)$$

Where:

$P$ : is the dataset ,  $T$ : attribute.

The gain is used to determine the root node of the tree. The root node represents the attribute that best splits the data. Algorithm (2.2) illustrate

Decision Tree[33].

<b>Algorithm 2.2 Decision Tree</b>
<b>Input:</b> Training dataset $S$ , Classes $X$ , attribute $A$ . Value $v$ <b>Output:</b> Decision Tree Predictor $T$
<b>Begin</b> Calculate Entropy by applying equation (2.18) Calculate InformationGain by applying equation (2.19) Function BuildTree ( $S$ , $A$ split) <b>For</b> $i$ in attributeList Compute InformationGain ( $S, i$ ) Append InformationGain ( $S, i$ ) to IGLIST $A$ max = attributemax ( IGLIST) <b>If</b> InformationGain ( $S$ , $A$ max) > InformationGain ( $S$ , $A$ split) <b>then</b> <b>For all</b> $v \in$ to val ( $A$ max) Subset = ( $x \in S$   $X$ max = $v$ ) Build (Subset, $A$ max) <b>End</b>

### 2.6.3 Random Forest (RF)

An ensemble can be described as a composition of multiple weak learners to form one with (expected) higher predictive performance (strong learner), such that a weak learner is loosely defined as a learner that performs slightly better than random guessing. There are several types of ensemble learning methods, such as Bagging (Bootstrap), Boosting (AdaBoost ), and Stacking [36].

To enhance classification performance, numerous decision trees are combined to create Random Forest, an ensemble classifier. To determine the final classification, random forest is a popular ensemble learning



algorithm that uses bagging with decision trees as base models. It computes the outcome using the majority voting method[37]. Algorithm (2.3) illustrate Ensemble Random Forest[33].

<b>Algorithm 2.3 Random Forest</b>
<p><b>Input:</b> Training set S, features F, Class Y, number of trees B, the Weight H.</p> <p><b>Output:</b> Predictor of learned Tree F</p>
<p><b>Begin</b></p> <p>Function RandomForest (S,F)</p> <p style="padding-left: 2em;">H=0</p> <p style="padding-left: 2em;">For i = 0 to B</p> <p style="padding-left: 4em;">si = sample subset of S</p> <p style="padding-left: 4em;">hi = RandomizedTreeLearn (si,F)</p> <p style="padding-left: 4em;">H=H+hi</p> <p style="padding-left: 2em;">End for</p> <p style="padding-left: 2em;">Return H</p> <p>End function.</p> <p>Function RandomaizedTreeLearned (S. F)</p> <p style="padding-left: 2em;">At each Node</p> <p style="padding-left: 4em;">f= small subset of F</p> <p style="padding-left: 4em;">spilt on best feature of f</p> <p style="padding-left: 4em;">return learned tree classifier</p> <p>End function</p> <p><b>End</b></p>

Random forests avoid overfitting, which increases bias and reduces variance, thereby improving performance by using various parts of the same training data set in different trees. This allows them to average many decision trees and avoid overfitting[37]. Figure (2.9) example of Random Forest [6].

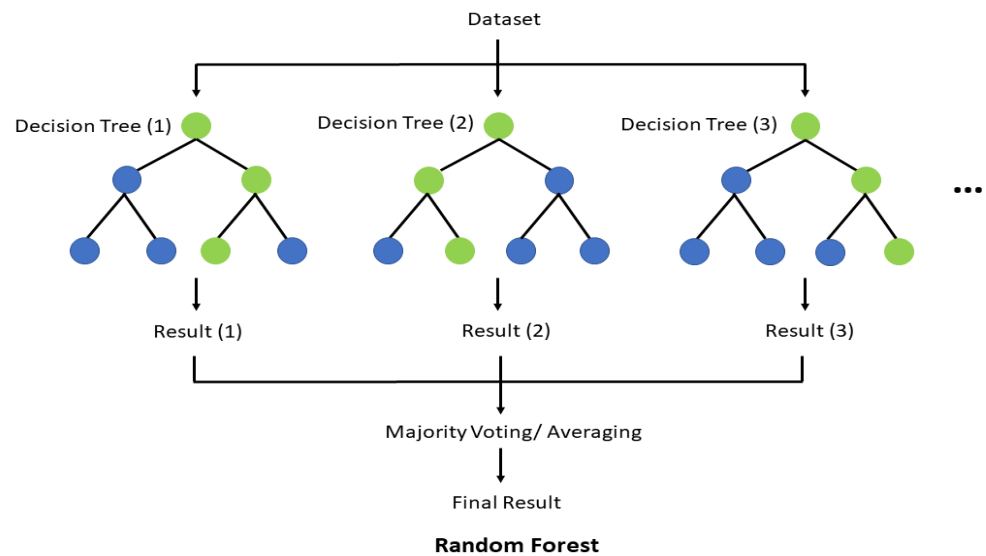


Figure 2.9: Internal working of Random Forest [23]

Majority voting is the most popular and intuitive combination, the predictions for each class are summed, and the class with the majority vote is returned a prediction[37].

#### 2.6.4 Naïve Bayes (NB)

Works on class probabilities and features to make a rating decision. Works well for simple data and can be used for quick classification. It is based on specific assumptions regarding the independence between features which causes a weakening of the classifier. Naive Bayes classification is a machine learning-based supervised classification technique that makes use of the Bayesian probability theorem and probabilistic approaches. Because it presumes that feature occurrences are unrelated to one another, the Nave Bayes algorithm is known as "nave."[38]. Algorithm (2.4) illustrate Naive

Bayes[33].

<b>Algorithm 2.4 Naive Bayes</b>	
<b>Input:</b>	Training/testing dataset T, F= (f1, f2, f3.., fn)
<b>Output:</b>	Estimated class K
<b>Begin</b>	
<b>Step 1:</b> Read the training dataset T.	
<b>Step 2:</b> Calculate the mean and standard deviation of the predictor variables in each class.	
<b>Step 3:</b> Repeat Calculate the probability of fi using the gauss density equation in each class;	
Until the probability of all predictor variables (f1, f2, f3.,, fn) has been calculated.	
<b>Step 4:</b> Calculate the likelihood for each class.	
<b>Step 5:</b> Get the greatest likelihood;	
<b>End</b>	

The following Bayes formula serves as the foundation for the Naive Bayes theorem.

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)} \quad (2.20)$$

Where:

$x$ : attributes,  $c$ : class

$(c | x)$ : The probability of  $c$  occurring given  $x$

$(x | c)$ : The probability that only  $x$  occurs, given  $c$

$(c)$ : probability of event  $c$ ,  $(x)$ : probability of event  $x$

When  $x$  is replaced, Bayes' formula can be expressed as follows [39]:

$$P(c|x_1, x_2, \dots, x_n) = \frac{P(c)P(x_1, x_2, \dots, x_n|c)}{P(x_1, x_2, \dots, x_n)} \quad (2.21)$$

Where:

$x : x_1, x_2, x_3, \dots, x_n$

$P(c|x_1, x_2, \dots, x_n)$ : The probability of  $c$  occurring given  $x$

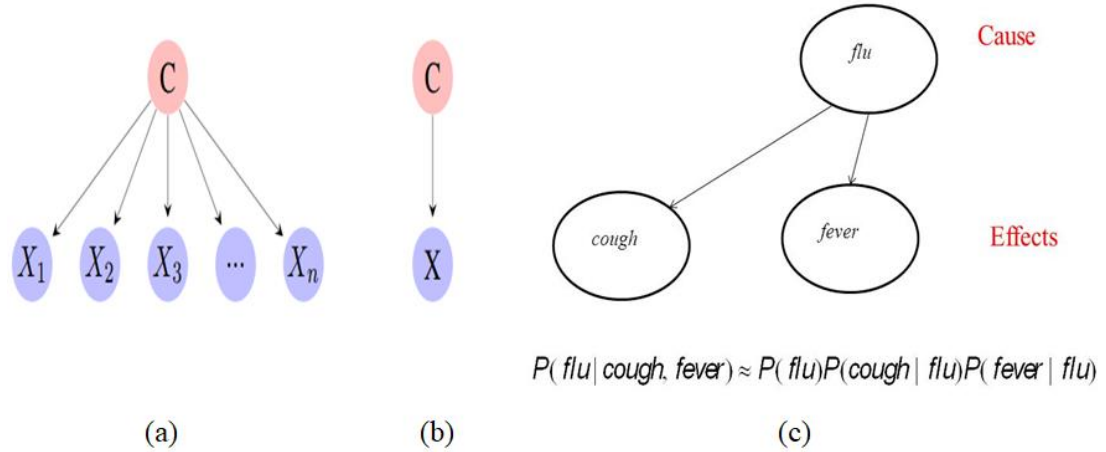


Figure 2.10:(a) A Naive Bayes Classifier, where  $C$  represents the class node and  $X$  represents the feature node; (b) A simplified Graphical Model of Naive Bayes Classifier; (c) example Naive Bayes Classifier [40]

## 2.7 Neural Networks

Artificial neurons, which resemble the neurons in the human brain, are the fundamental building block of neural networks[41]. With weighted input signals and an activation function-based output signal, these artificial neurons are potent computing units. In a neural network, these neurons are dispersed among the many layers. Typically, an artificial neural network includes three layers [42].

- **Input Layer:** This layer consists of neurons and they just receive the input neuron multiplied by a weight value. Which can be adjusted during training time and pass it to the next layer [41].
- **Hidden Layer:** These are layers that lie between the input layer and the output layer. Neurons in this layer modify the input before sending it to the next layer [41].

- **Output Layer:** This layer utilizes an activation function to solve a classification, the anticipated feature or class is the output layer [46].
- **Bias:** additional inputs to neurons; always have the value 1, and have a connection weight of 1. This guarantees that the neuron will be activated even if all of the inputs are zeros (all 0s) [41].

The weights are adjusted to do this. The signal between each connection in a neural network is often controlled by a weight. If the output is satisfactory, no modification is required; however, if production is below average, the system adjusts to increase production by changing weights. The system compares the outputs with the original outputs previously provided in training mode a technique to assess how well the outputs perform [41]. A basic artificial neural network is shown in Figure (2.11).

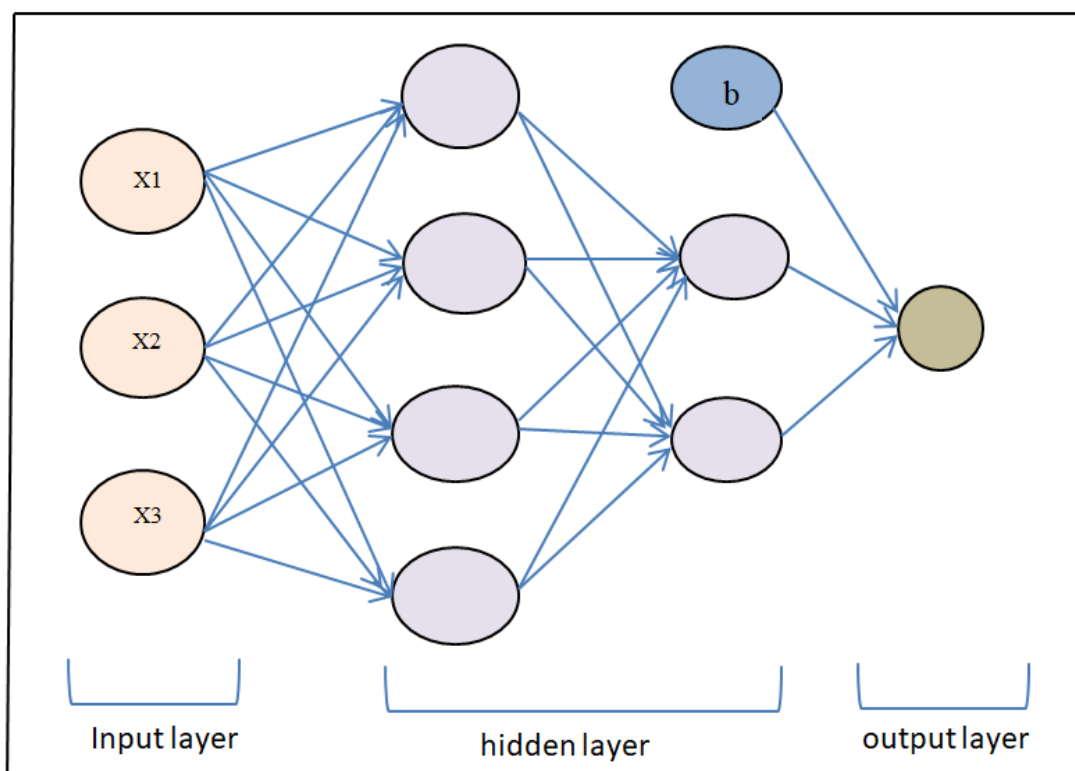


Figure 2.11: A simple Artificial Neural Network

## 2.8 Deep Learning

One of the Artificial Intelligence (AI) innovations that have transformed the market and how machines operate is deep learning. We have advanced in many areas of daily life as a result of this innovation, paving the way for how we will engage with services and technology in the future. The field of artificial intelligence, known as deep learning, is the search for a set of rules and procedures that allow robots to behave and make decision based on data, rather than being explicitly programmed to do so, while looking at a lot of data, they can find patterns and use models to come to conclusions. As a result, machines are now equipped to generate predictions via data processing [43].

### 2.8.1 New Train

Is the process of optimizing the parameters of a machine learning model using a specific data set.

#### 2.8.1.1 Convolution Neural Network

It content type of layers convolution layer, pooling layer, fully connected layer and other layers [44].

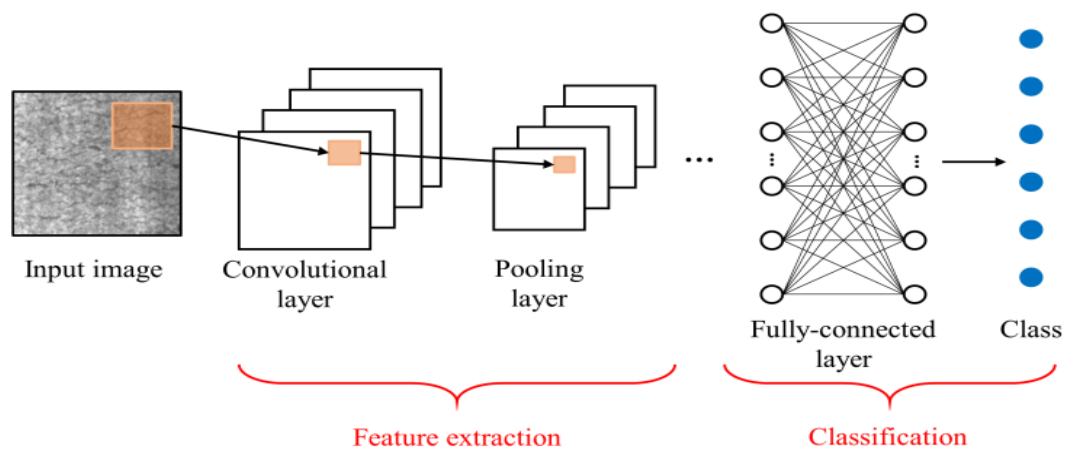


Figure 2.12: Typical architecture of convolutional neural network[44]

## 1. Convolution Layer

The weight matrix is taken into account when assigning convolutional layers to filters, because the size of the neuron is determined by the input image of each filter. After calculating the dot product of the weight matrix and filter matrix, add the offset value to the dot product. Filter then moves forward with each step specified in steps, tracking the complete input matrix [45]. Figure (2.13) shows the number of weights used by the filter.

The number of parameters used in the convolutional layer is given by Eq. (2.22)

$$P = ((M \times N \times K + 1) \times \text{number of filters}) \quad (2.22)$$

The height, width, and number of input filter channels are represented by M, N and K respectively [45].

The output size of this layer can be obtained using the formula below [46].

$$O = ((\text{inputsize} - \text{filtersize} + 2 \times \text{padding}) \setminus \text{stride}) + 1 \quad (2.23)$$

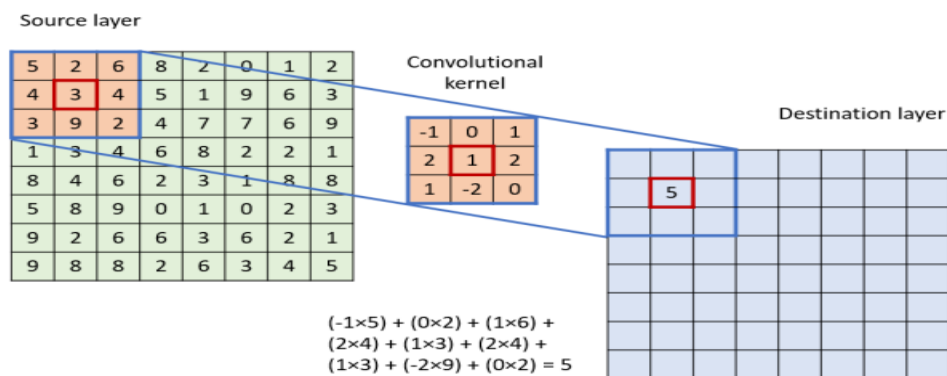


Figure 2.13: Convolutional operation showed schematically. In order to fill the pixels in the destination layer, the convolutional kernel shifts over the source layer[46]

## 2. Pooling Layers

Pooling layers aim to reduce the dimensionality of feature maps in order to reduce computational complexity. It is a sample estimation procedure. This function is often implemented using one of two methods, as shown in Figure (2.14). The maximum sampling output of the previous convolutional layer depends on the maximum value of each sub-patch. The average value of each sub-region of the feature maps is calculated using another method called mean pooling. The representative part of the input data is produced by grouping the layers. Each feature map undergoes a separate clustering procedure. With minor modifications from previous layers, it is not delicate. When collecting features, the feature map often does not have any overlap, and more than one step is required to achieve dimensionality reduction [47].

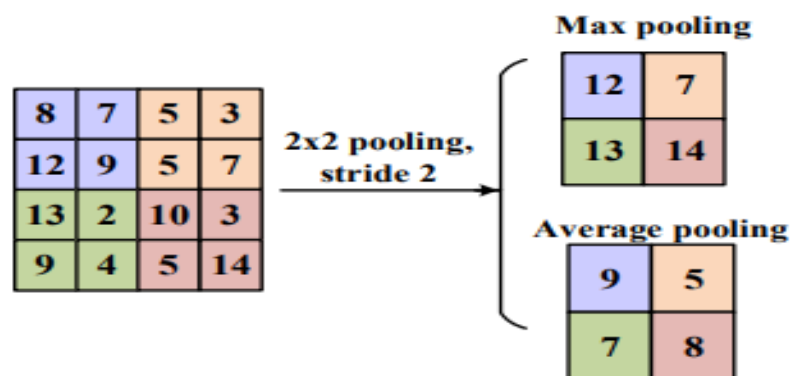


Figure 2.14: Displaying pooling activities visually, including maximum and average pools[47]

There are a few symmetric aggregation functions in the pooling layer, including:[48].

- Max Pooling: It provides the highest value from its immediate, rectangular area of the feature map.



- Average Pooling: This method calculates the average of elements present in feature map area covered by the filter[48].

### 3. Flattening Layer

In flatten reduces the three-dimensional matrix into one-dimensional vector so that it can be easily given as an input to the next layer [49].

### 4. Fully Connected Layer

Every neuron in the previous layer is connected to every neuron in the next layer in the fully connected (FC) layer. In the output layer, this layer often makes use of the Softmax activation function. The FC layer develops an N-dimensional feature vector, where N is a number of output classes, from the high-level characteristics of the previous layers. The output vector's values each correspond to a class's probability [50].

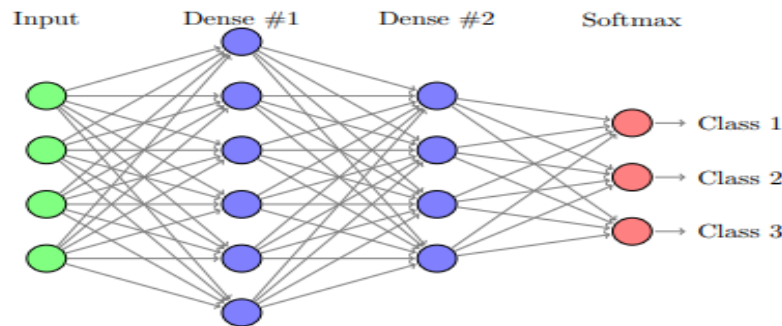


Figure 2. 15: Architecture of Fully Connected Layers[51]

$$U_i^l = \sum_j w_{ji}^{l-1} y_j^{l-1} \quad (2.24)$$

$$y_i^l = f(u_i^l) + b^{(l)} \quad (2.25)$$

where  $U_i^l$  is the value of the output layer,  $y_i^l$  is the value created in the

proposed output layer,  $wji^{l-1}$  is weight value of the hidden layer,  $yj^{l-1}$  is value of the input neurons, and  $b^{(l)}$  is value of deviance [10].

### 5. Batch Normalization Layer

Normalize the output produced by the proposed convolution layers using a batch normalization layer. Proposed shortening the training time of the model by normalization, leading to a more rapid and effective learning process [10]. The batch normalization process is given in Equations (26–28).

$$Yi = \frac{xi - \mu^{\text{batch}}}{\sqrt{\sigma^2_{\text{batch}} + \epsilon}} \quad (2.26)$$

$$\sigma^2_{\text{batch}} = \frac{1}{M} (Xi - \mu^{\text{batch}})^2 \quad (2.27)$$

$$\mu^{\text{batch}} = \frac{1}{M} \sum_{i=1}^M xi \quad (2.28)$$

$Yi$  represents the output of a neuron,  $xi$  represents the input to that neuron.  $\mu$  is the mean of the batch,  $\beta$  is a learnable scaling parameter,  $\sigma$  is the standard deviation of the batch,  $\epsilon$  is a small constant (usually added to avoid division by zero).

### 6. Classification Layer

It is a fundamental task in machine learning and statistics that involves categorizing data points into predefined classes depending on the specific task, such as binary classification, and multiclass classification.

#### 2.8.2 Pre-trained

Is the process of initializing the parameters of a machine learning

model using weights learned from another task or dataset. There are different types of Pre-trained models including VGG (such as VGG16 or VGG19), GoogLeNet (such as InceptionV3), the remaining network (such as ResNet50), DenseNet, and MobileNet[52].

### 2.8.2.1 Visual Geometry Group 16 (VGG16)

The network architecture of VGG is a CNN model. A unique variant of VGG is called VGG-16 and has 16 weighted layers. Convolutional layer, max pooling layer, activation layer and fully connected layer constitute its layers [53]. Figure (2.16) illustrates the structure of VGG-16 model.

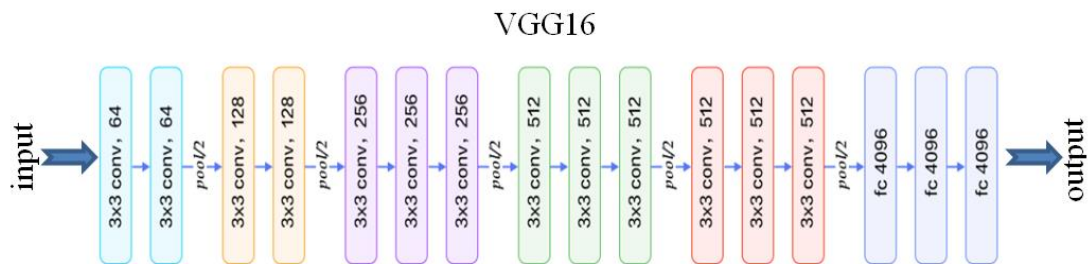


Figure 2.16: Structure of VGG-16 model[54]

- The architecture has a total of 21 layers, including 13 convolutional layers, 5 pooling layers, and 3 dense layers.
- There are only 16 weighted categories. The first convolutional layer has 64 filters, the second layer has 128 filters, the third layer has 256 filters, and the fourth and fifth layers have 512 filters.

Pre-Trained the VGG-16 network using the ImageNet dataset containing over 14 million images and 1000 labels and achieved 92.7% accuracy[13][53].

## 2.9 Transfer Learning

Transfer learning transfers weights of a network that has been previously trained on large amounts of data to another model designed to solve a related problem. The basic idea here is that models pre trained on large, diverse datasets such as ImageNet can be used to extract general features suitable for specific tasks [55]. This strategy becomes crucial when there is not enough training data for the problem at hand. Networks may face the problem of overfitting limited data and thus fail to generalize well. If the data set for the training model is large enough, the transmitted network parameters can ensure the accurate classification of small amounts of data. Train only the classifiers for the last part of the new model and pass them the estimated weights of the previously trained model. According to research on medical image analysis, transfer learning is useful when only a few images are available [56].

## 2.10 Activation Function types

In a neural network, activation functions are mathematical operations that are applied to the outputs of each neuron. They give the network nonlinearity, enabling it to learn intricate patterns and perform nonlinear transformations on the input data [57].

For nonlinear transformation processes, DL-based models frequently employ activation functions. The most popular and extensively adopted activation functions created in the past were Sigmoid, Tanh, and ReLU. There are different types of Activation Function which are [10]:

### 2.10.1 ReLU:

The Rectified Linear Unit (ReLU), uses a nonlinear pixel-wise

function . It returns  $x$  if  $x$  is positive; else, it returns 0, using the equation formula[58].

$$f(x)_{ReLU} = \max(0, x) \quad (2.29)$$

### 2.10.2 Sigmoid

The sigmoid function is a function that maps the input values between 0 and 1. It is used in binary classification tasks[59].

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.30)$$

Where:

$x$  : is the input to the sigmoid function

### 2.10.3 SoftMax

A softmax function is a function that takes a vector of value as input and outputs a probability distribution over multiple classes. It is used in multi-class classification tasks[10].

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^N x_j} \quad (2.31)$$

Where:

$N$ : is the total number of classes

$x_i$ : is the element of the input vector  $x_i$ .

Table 2.1 lists the most common fully connected layer activation functions used for various purposes.

*Table 2.1: The most commonly used fully connected layer activation functions are listed below [60]*

Last layer Activation Function	Task
--------------------------------	------

---

Linear	Continuous regression
Sigmoid	Multi-label, single node, pair classification nodes and binary classification
Softmax	Divided into single category and multiple categories

---

## 2.11 Loss Function

Neural network models compute error using the difference between the actual output and expected output. Numerous features of loss functions have recently been created to calculate error in neural networks. There are different types of loss functions which are: (Binary Cross-entropy, Categorical Cross-Entropy, hinge, squared hinge, and Jensen-Shannon divergence are loss functions) [61].

### 2.11.1 Binary Cross-Entropy

When a situation calls for binary classification, this strategy is employed by default and is the most effective one. When the goal values are either 0 or 1, it is presumptive and functional. The calculated value encapsulates the mean discrepancy between the calculated value probability distribution and the expected value probability distribution[62].

$$S_{cross}(y_j, \hat{y}_j) = -\frac{1}{n} \sum_{k=1}^n y_{jk} \log \hat{y}_{jk} \quad (2.32)$$

Where  $y_j$  actual target vector,  $\hat{y}_j$  the output predicted vector,  $k$  vector length.

### 2.11.2 Categorical Cross-Entropy

This loss function is employed in multi-class classification problems. In some cases, the target may only belong to one of several possible categories, and the model must determine which one. This function is widely used to compare two probability distributions since it is employed in networks that employ the Soft-max activation function [63].

$$S_{cross} - entropy(y_j, \hat{y}_j) = -\sum_{k=1}^n y_{jk} \log \hat{y}_{jk} \quad (2.33)$$

This loss function calculates the anticipated error in the observed events using the distribution  $\hat{y}_j$  while the events' information is valued in accordance with  $y_j$ [61].

### 2.12 Optimizer

Gradient descent is used by the majority of neural network-based methods, including CNN, to reduce error rates during training and parameter review. Gradient descent is the name of the first-order optimization method, and its derivatives show patterns and rising or falling error functions. Based on this knowledge, the error function is adjusted to a local minimum. The traditional batch gradient descent method takes a long time to process because it calculates the gradient for all training data. Common optimization algorithms are adaptive momentum, root mean spread propagation, and stochastic gradient descent [64]:

### 2.12.1 Adaptive momentum estimation (Adam)

Adam calculates an adaptive learning rate for each gradient training parameter. Low-memory first-order gradients are a very simple and computationally efficient stochastic optimization method. In machine learning issues involving high-dimensional parameter spaces and enormous data sets, this technique independently calculates learning rates for various parameters based on estimates of first- and second-order moments [64].

The mathematical notation for Adam is as follows:

$$x_{t_{new}} = \delta 1 * x_{t-1} + (1 - \delta 1) * g_t \quad (2.34)$$

$$y_{t_{new}} = \delta 2 * y_{t-1} + (1 - \delta 2) * g^2_t \quad (2.35)$$

$$\Delta \omega_t = -\eta \frac{x_{t_{new}}}{\sqrt{y_{t_{new}} + \epsilon}} * g_t \quad (2.36)$$

$$\omega_{t+1} = \omega_t + \Delta \omega_t \quad (2.37)$$

- $\eta$ : initial learning rate
- $g_t$ : gradient along time  $t$
- $x_t$ : first momentum
- $y_t$ : second momentum
- $\delta 1, \delta 2$ : hyperparameters
- $\omega_{t+1}$ : updated weight
- $\omega_t$ : current weight
- $\Delta \omega_t$ : is the change to the weight vector at time

Adam is computationally cheaper, uses less memory at runtime. Covers large data sets, hyperparameters, noisy data, insufficient gradients, and



non-stationary problems that require careful tuning. Adam's configuration options include alpha, which is the learning rate or step size[64].

### 2.13 Evaluation Matrices

The following metrics (accuracy, precision, recall, F1 score, and confusion matrix) are used to evaluate the effectiveness of the system's performance in detecting and classifying.

The following measures are used to determine how well the suggested system performs [65]:

#### 2.13.1 Confusion Matrix

Is an important tool for comprehensively evaluating classifiers, in which the relationships between correct and false classifications of the model. The confusion matrix consists of four concepts: (True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN)) are the abbreviations for actual correct positive and negative examples [48].

In a confusion matrix for a binary classifier, the actual values are labeled "true" (1) and "false" (0), while the predicted values are labeled "positive" (1) and "negative" (0). Multiclass classification is a classification that has more than two classes [66]. Table (2.2) illustrate of multiclass classification.

Table 2.2: Confusion Matrix [67].

		True Class			
		A	B	C	D
Predicted class	A	$TP_A$	$E_{BA}$	$E_{CA}$	$E_{DA}$
	B	$E_{AB}$	$TP_B$	$E_{CB}$	$E_{DB}$
	C	$E_{AC}$	$E_{BC}$	$TP_C$	$E_{DC}$
	D	$E_{AD}$	$E_{BD}$	$E_{CD}$	$TP_D$

The confusion matrix consists of four parameters as shown below:

- True Positive (TP): Number of correctly classified tumor images.
- True Negatives (TN): Number of correctly classified non-tumor images.
- False Positives (FP): Number of tumor images incorrectly classified as non-tumor.
- False Negatives (FN): Number of non-tumor images that were incorrectly classified as tumor images [48].

Diagonal elements: These elements represent the correctly classified instances for each class. They show how well the classifier is doing in identifying each class correctly.

Off-diagonal elements: These elements show misclassifications [68].

### 2.13.2 Accuracy

The first evaluation method used in this study is the accuracy parameter; it measures the proportion of correctly classified data points out of the total number of instances in the dataset. In other words,

accuracy is how often the model makes correct predictions [69].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.38)$$

### 2.13.3 Sensitivity (Recall)

The likelihood that the test will identify the anomalous example among all anomalous cases [70].

$$Sensitivity = \frac{TP}{TP+FN} \quad (2.39)$$

### 2.13.4 Precision

The formula for precision is found in Eq. (2.40). Precision is a measure of how well a classification eliminates false positives [53].

$$Precision = \frac{TP}{TP+FP} \quad (2.40)$$

### 2.13.5 F1<sub>Score</sub>

It balances both criteria by combining precision and recall into a single measurement. The F1 score, which measures a model's performance more holistically than accuracy alone this is the harmonious meaning of precision and recall[71].

$$F1_{Score} = \frac{2*(Recall)*(Precision)}{(Recall+Precision)} \quad (2.41)$$

## 2.14 Related Work

Last few years, many researchers have become interested in the segmentation and classification of brain MRI data. A brief summary of some of recent curricula is provided.

1. The authors (Minz and Mahobiya, 2017) developed a successful technique using the AdaBoost algorithm for the automatic classification of brain images. Three essential parts make up the suggested system. Pre-processing was used to reduce dataset noise, utilizing threshold segmentation and averaging filtering. Feature extraction using Grey Level Co-occurrence Matrix (GLCM), and Classification Using Adaboost Algorithm. Their model has Accuracy 89.90%, Sensitivity 88.23%, Specificity 62.5% [72].
2. In the article, (Megha and Sushma, 2019) segmentation is turned into a binary image using a threshold value, followed by the application of morphological operations to the image, such as erosion, dilation, or region filling, and classification using SVM, with an accuracy rate of 83.3 % [73].
3. The authors (Arunkumar et al., 2019) employed magnetic resonance imaging (MRI) data to train artificial neural network (ANN) model. Brain tumor segmentation method using K-means clustering, first naming the regions based on grayscale. In the second step, they used an artificial neural network (ANN) to select the appropriate object based on the training phase. Third, tissue features of the peripheral brain tumor region that had reached the mitotic stage were eliminated. Brain tumors are identified, evaluated, and diagnosed using grayscale features to determine whether they are benign or malignant. Their model evaluates and compares the segmentation and classification results between ANN and SVM. Their model has a 94.07% accuracy, a 90.09% sensitivity, and a 96.78% specificity [74].

4. The authors (Pravitasari et al., 2020) suggested deploying a new ROI and region of interest (ROI) classification architecture along with the UNet-VGG16 fully convolutional network. Transfer learning is used to streamline the U-Net design in this model or architecture, which combines U-Net with VGG16. In the learning data set, this approach has a high accuracy of roughly 96.1%. By calculating the correct classification ratio (CCR) and contrasting the segmentation outcome with the actual data, validation is carried out. The brain tumor location can be located using UNet-VGG16, according to the correct classification ratio (CCR) value, which averages 95.69% [75].

5. The authors (Sameer et al., 2020) proposed an idea to improve image contrast in MRI images using adaptive histogram equation (AHE). In addition, the U-NET algorithm was used to create a fully automated segmentation system that provides powerful and efficient segmentation. The segmentation model is then output as input to the 3D CNN utilized for classification, with 96% and 98.5% accuracy rates utilizing 5-fold and 10-fold, respectively, due to the high resolution approach through the usage of MRI images [76].

6. The authors (Kumar et al., 2021) proposed K-nearest neighbor (KNN) An automated system for classifying brain tumor in Magnetic resonance imaging (MRI) images, and a technique employed to categorize MRI images as abnormal or normal. The fuzzy C-means clustering strategy and pooling algorithms are used to separate tumor region. The trial makes use of the MICCAI and BRATS dataset, whose

results are evaluated with 96.5 % accuracy, 100% sensitivity, and 93% specificity [77].

7. In the article, (Garg and Garg, 2021) authors Automatic diagnosis of brain tumors using a majority voting based hybrid ensemble classifier (KNN-RF-DT). The technique utilized in KNN-RF-DT hybrid ensemble approach. Used Otsu threshold method for brain tumor segmentation. Feature extraction is carried out using the stationary wavelet transform (SWT), principal component analysis (PCA), and grey level co-occurrence matrix (GLCM). This approach has a 97.305% accuracy rate, which is quite high. However, the suggested approach offers good accuracy at the expense of making the training process more difficult[6].

8. The authors (Habib et al., 2021) suggested system consists of subsequent actions. Noise reduction and image enhancement are both part of the pre-processing stage. The image is then divided into segments using the k-means pooling, watershed algorithm, thresholding, multiple segmentation, and average score of all algorithms. The suggested research employs Support Vector Machine (SVM) classifier accuracy of 90% following segmentation of the texture- and shape-based features, The features for classification can be extracted using the feature extraction method [78].

9. The authors (Filatov and Yar, 2022) proposed the use of pre-trained convolutional neural networks (CNN) for the diagnosis and classification of brain tumors. . Networks that has been used are ResNet50, EfficientNetB1, EfficientNetB7, EfficientNetV2B1. EfficientNet has shown promising results due to its scalable nature.

EfficientNetB1 showed the best results with training and validation accuracy of 87.67% and 89.55% respectively[79].

10. The authors (Ullah et al., 2023) suggested a method initially, Gabor filter and ResNet50 were applied to accurately extract the important features of brain tumors from the MRI images dataset. Firstly, the extracted features of Gabor and ResNet50 were classified individually through SVM, and secondly, the features from both these techniques were combined and then classified through SVM. The best results were shown by the combined features of Gabor and ResNet50, an advanced hybrid approach with 95.73% accuracy, 95.90% precision, and 95.72% f1 score[71].

11. In this paper, (Gómez et al., 2022) It is recommended The CNN models evaluated are Generic CNN, ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2, and EfficientNetB0. In the comparison of all CNN models, including a generic CNN and six pre-trained models, the best CNN model for this dataset was InceptionV3, which obtained an average Accuracy of 97.12% [80]. Table 2.3 provides an illustration of the summary of related work.

*Table 2.3: Summary of related works*

Author	Year	Proposed model	Dataset	Different Metrics
Minz and Mahobiya	2017	Adaboost for classification boosting, uses thresholding segmentation and median filtering	Acquisition dataset	Accuracy rate reaches 89.90% Sensitivity 88.23% Specificity 62.5%
Megha and Sushma	2019	Segmentation is using a threshold value, classification using SVM.	Collected dataset	Accuracy % 83.3

---

Arunkumar et al.	2019	Segmentation using K-means clustering and employed ANN to classifier	data collections— training data sets and testing data group	Accuracy 94.07% Sensitivity 90.09% and Specificity 96.78%
Pravitasari et al.	2020	The UNet-VGG16 fully convolutional network with used to classify ROI and non-ROI.	Collected dataset	Accuracy of about 96.1% CCR value is about 95.69%.
Sameer et al.	2020	Segmentation using U-NET and 3D CNN used for classification	Brats-2015 dataset	An accurate rate using 5-fold and 10-fold is 96% and 98.5%, respectively.
Kumar et al.	2021	K-nearest neighbor for classification, and fuzz C-means clustering for segmentation	Publicly accessible dataset with the BRATS MICCAI brain tumor dataset	Accuracy 96.5%, Sensitivity 100% and Specificity 93%
Garg and Garg	2021	Classifier segmentation (KNN-RF-DT) based on majority voting technique and Otsu threshold.	dataset available online	Accuracy 97.305%
Habib et al.	2021	k-means clustering, watershed algorithm and thresholding used multiple segmentation and SVM Classifier	loading the dataset from a directory comprising healthy and unhealthy folders	Accuracy 90%
Filatov et al.	2022	ResNet50, EfficientNetB1, EfficientNetB7, and EfficientNetV2B1 are some of the CNN networks that have been employed in the training process .The results were best for EfficientNetB1 in this case	figshare, SARTAJ dataset, and Br35H	Accuracy of 87.67%
Ullah et al.	2023	extracted features combined of Gabor and ResNet50 and then classified by SVM	figshare, SARTAJ dataset, and Br35H	Accuracy 95.73%, precision 95.90%, and f1 score 95.72%
Gómez et al.	2023	Generic CNN, ResNet50,	figshare, SARTAJ	Accuracy



---

el.	InceptionV3, InceptionResNetV2, Xception, MobileNetV2, and EfficientNetB0. The best CNN model was InceptionV3	dataset, and Br35H	97.12%
-----	--	--------------------	--------

---

## 2.15 Summary

This chapter summarizes the theoretical basis for diagnosing a brain tumor. It also explains machine learning and some of its algorithms; these include Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM). Convolutional neural networks are among the strategies, along with deep learning. Additionally, a set of criteria is used to analyze the effectiveness of the algorithms and the suggested model in order to evaluate the method as a whole. These criteria include accuracy, precision, recall, and an f1 score. Finally, it is explained related work.

## **CHAPTER THREE**

# **PROPOSED METHODOLOGY**

### 3.1 Overview

The creation of an intelligent MRI-based model for brain tumor classification is the main objective of this work. In this chapter, the proposed method is presented and explained in detail.

### 3.2 Proposed Models

The proposed models presents a tumor detection model using segmentation techniques and MRI image classification for the four types of human brain tumors (pituitary, glioma meningioma, and no tumor in normal cases). Two models have been applied:

1. The first model is a combination (VGG16) with machine learning algorithms (SVM, DT, RF, NB).

This model is applied after the pre-processing stage and the segmentation technique are completed. The output of the segmentation process will enter VGG16, which extracts complex and overlapping features from the images. This improves the accuracy of the model by extracting features related to shape, texture, color, and contrast. The model can recognize certain patterns such as tumor boundaries or fine features within them. These images are then entered into one of the machine learning algorithms. The data is trained within the algorithms and finally it is classified into the four types of brain tumors.

2. The second model is (CNN).

In this model, feature extracted and trained using a CNN model for layers with different parameter values.

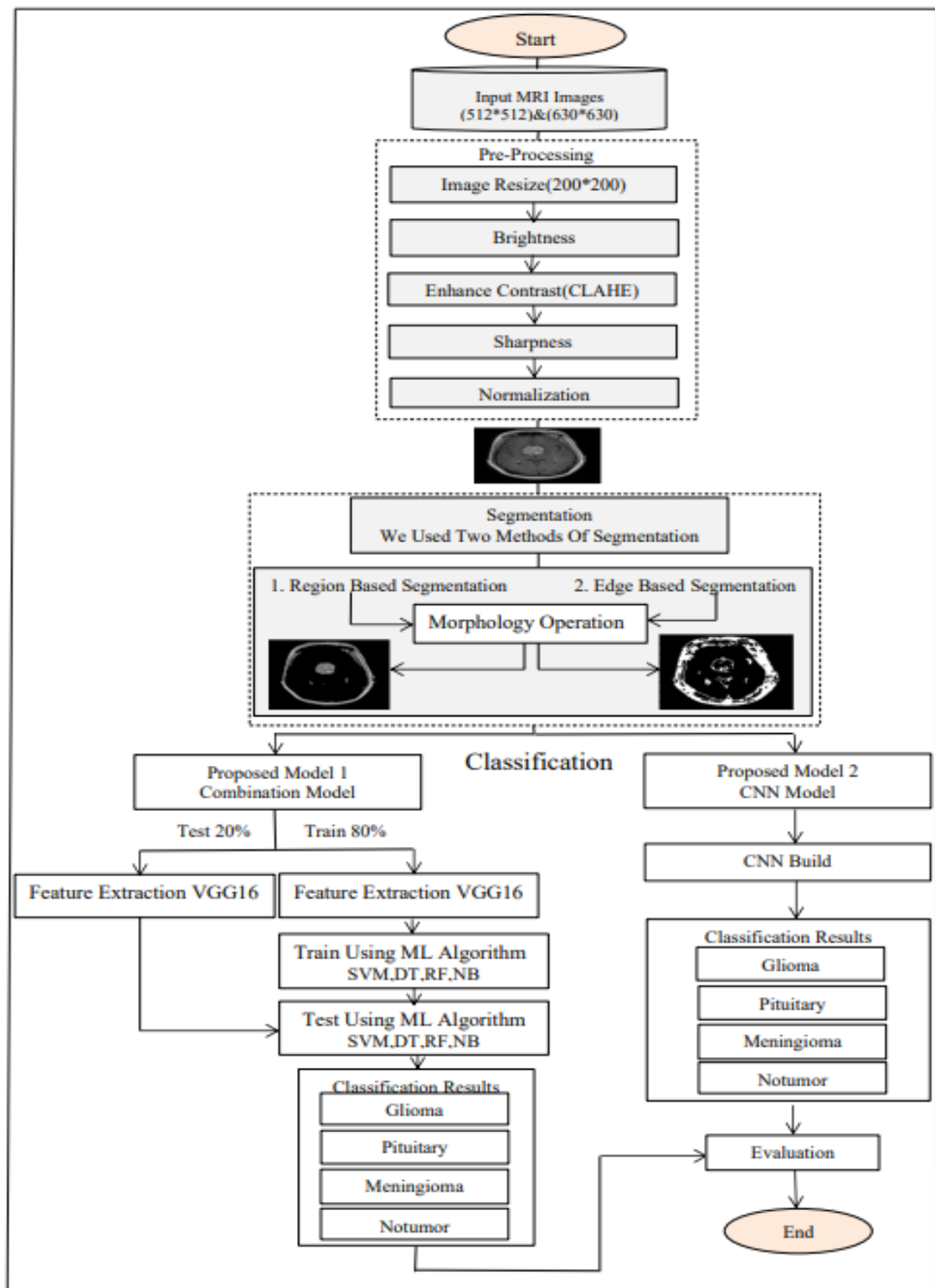


Figure 2.1: Block diagram of the two proposed Models

### 3.2.1 Combination Model

The combination model consists of several steps:

- Step1: The input dataset MRI with gray scale channels and splitting data into train and test.
- Step2: pre-processing.
- Step3: Segmentation is done using region- based segmentation, and edge- based segmentation. Two segmentation techniques were used to compare them.
- Step4: Feature extraction is done using pre-trained VGG16.
- Step5: using classifiers (machine learning algorithms).

Random Forest gives the best result in terms of the accuracy of tumor classification of combination model.

- Step6: Classification by traditional classifiers.

<b>Algorithm 3. 1 Combination Model</b>
<b>Input:</b> Image Segmentation
<b>Output:</b> Image Classifier “multiclass” (pituitary, meningioma, glioma, and no tumor)
<b>Begin</b>
<b>Step 1:</b> Call VGG16 model pre-trained
<b>Step 2:</b> Resized to the required input shape (200x200)
<b>Step 3;</b> Features Extraction from images using the VGG16 model
<b>Step 4:</b> Reshape features to one dimension(vector).
<b>Step 5:</b> Call Machine learning algorithms (SVM, DT, RF, and NB) respectively / Equations (2.1), (2.2), (2.3), and (2.4)
<b>Step 6:</b> Testing and evaluation phase

<b>End</b>
------------

### 3.2.2 CNN Model

The CNN model consists of several steps:

- Step1: The input dataset MRI with gray scale channels and splitting data into train and test.
- Step2: pre-processing.
- Step3: Segmentation is done using region- based segmentation, and edge- based segmentation. Two segmentation techniques were used to compare them.
- Step4: Build (CNN).
- Step5: Classification by CNN.

<b>Algorithm 3. 2 CNN Model</b>
<b>Input:</b> Image Segmentation
<b>Output:</b> Image Classifier “multiclass” (pituitary, meningioma, glioma, and no tumor)
<b>Begin</b>
<b>Step 1:</b> Define the build_ model() as sequential model
<b>Step 2:</b> Build convolution layer2D of 32 channel of 3x3 kernels , input_ shape=(200,200,3) ,activation= “ relu” and same padding
<b>Step 3:</b> Build max pool layer with pool size (2x2), and stride 2x2
<b>Step 4:</b> Build convolution layer of 64 channel of 3x3 kernels , activation= “ relu”, and same padding
<b>Step 5:</b> Build max pool layer with pool size (2x2), and stride (2x2)
<b>Step 6:</b> Build convolution layer of 32 channel of 3x3 kernels ,

---

activation= “ relu” ,and same padding

**Step 7:** Build max pool layer with pool size (2x2), and stride (2x2)

**Step 8:** Build convolution layer of 64 channel of 3x3 kernels ,  
activation= “ relu”, and same padding

**Step 9:** Build max pool layer with pool size (2x2), and stride (2x2)

**Step 10:** Build convolution layer of 32 channel of 3x3 kernels ,  
activation= “ relu”, and same padding

**Step 11:** Build max pool layer with pool size (2x2), and stride (2x2)

**Step 12:** Build convolution layer of 64 channel of 3x3 kernels ,  
activation= “ relu”, and same padding

**Step 13:** Build max pool layer with pool size (2x2), and stride (2x2)

**Step 14:** Build flatten layer, extracted features from the previous layer  
into a 1-dimentsional

**Step 15:** Build batch normalization layer

**Step 16:** Build fully connected (dense) layer of 16 filters with  
activation='relu'

**Step 17:** Build fully connected (dense) layer of 4 filters with  
activation=' Softmax' it used for multiclass classification.

**Step 18:** Model. compile (optimizer=Adam(learning\_ rate=0.001),  
loss='sparse\_ categorical\_ cross entropy' , metrics=['accuracy'])

**Step 19:** Return model

**Step 20:** model = build\_ model()

**Step 21:** Print the summary of the model's architecture

**End**

### 3.3 Splitting Dataset

In the data splitting stage, the percentages of train data and test data are determined, 80% of the training data and 20% of the test data are split. This is the ratio that works best..

### 3.4 Preprocessing

Preprocessing of images plays an important role in providing enhanced image features. Smoothing and contrast enhancement techniques are employed to enhance details and make tumors more noticeable. Algorithm (3.3) illustrates Preprocessing image. The

<b>Algorithm 3. 3 pre-processing</b>
<p><b>Input:</b> Load images with the extension jpg dataset MRI</p> <p><b>Output:</b> Preprocessing images</p>
<p><b>Begin</b></p> <p><b>Step 1:</b> Resize the image from 512*512 to 200*200</p> <p><b>Step 2:</b> Adjust image brightness at random rate (0.8,1.2)</p> <p><b>Step 3:</b> Enhance Contrast (CLAHE) adjustment methods / Equation (2.1)</p> <p><b>Step 4:</b> Enhance image with a sharpness filter using sobel filter</p> <p><b>Step 5:</b> Normalizes the image dividing by 255 / Equation (2.2)</p> <p><b>End</b></p>



### 3.5 Segmentation for Brain Tumor

This procedure is carried out to lessen the quantity of work that needs to be done in the next steps. Two techniques for segmentation were used (region-based segmentation and edge-based segmentation).

#### 3.5.1 Segmentation for brain tumor using region based segmentation

Region split and merge is one of the image processing techniques that were used in this work for image segmentation. Algorithm (3.4) illustrate Region split and merge segmentation.

<b>Algorithm 3. 4 Split and Merge Segmentation</b>
<b>Input:</b> Preprocessing images
<b>Output:</b> Mask with enhanced contrast
<p><b>Begin</b></p> <p><b>Step 1:</b> :Convert the input image from RGB to HSV color space Hue, Saturation, Value.</p> <p><b>Step 2:</b> Split HSV image into separate planes</p> <p><b>Step 3:</b> Apply the CLAHE algorithm on the first plane (Hue) of the HSV image / Equations (2.12) ,(2.13)</p> <p><b>Step 4:</b> Merge the modified Hue channel with the original Saturation (S) and Value (V) channels</p> <p><b>Step 5:</b> Define lower and upper limits of the HSV value to create a range that represents the desired color to mask. In this case, The lower limit is [0, 0, 100] and the upper limit is [255, 30, 255]</p> <p><b>Step 6:</b> Create mask by comparing the HSV values of the image with the lower and upper bounds</p>

**Step 7:** Convert the resulting HSV image back to the RGB color space

**Step 8:** Returns the mask with enhanced contrast

**End**

### 3.5.2 Segmentation for brain tumor using edge based segmentation

In edge detection, use canny edge detection to find pixels that represent edge pixels of an object. First, the image is converted to grayscale and then two thresholds are applied. More than one trial value is performed, as shown in Tables (3.1).

*Table 3.1: Accuracy comparison between traditional machine learning classifiers using different Threshold*

Classifiers	Accuracy when Threshold	Accuracy when Threshold	Accuracy when Threshold
	110-200	150-300	50-200
SVM	93.67%	94.53%	91.50%
RF	98.78%	98.24%	98.33%
DT	91.53%	90.17%	92.86%
NB	81.54%	81.54%	80.54%
CNN	98.55%	98.25%	98.21%

The best results were achieved when using (110) as the low threshold and (200) as the high threshold. Experimentation and fine-tuning are necessary to achieve satisfactory results. Algorithm (3.5) illustrate Canny edge detection for segmentation.

---

**Algorithm 3. 5 Canny edge detection for segmentation**

**Input:** Preprocessing images

**Output:** 3-channel Gray scale image segmentation

**Begin**

**Step 1:** Convert input image from RGB to grayscale

**Step 2:** Apply canny edge detection, with a high threshold(200) and a low threshold(110), to determine potential edges

**Step 3:** Perform morphological closing operation

**Step 4:** Merged resulting edges to create 3-channel gray scale

**Step 5:** Return 3-channel gray scale image segmentation

**End**

### 3.6 Morphology Operation

Morphological processes help improve image quality closing operation have been implemented. Algorithm (3.6) illustrate morphology operation.

- First, a dilation process was performed, where the pixels were partially matched between the structural element and the image resulting from the segmentation process to fill in small gaps, enlarge areas, and add pixels.
- Second, an erosion process was performed, where the pixels were total matched between the structural element and the image resulting from the dilation process to remove pixels.

The morphological equations (2.11) and (2.12) mentioned in Chapter

Two were used.

<b>Algorithm 3. 6 Morphological Operations</b>
<p><b>Input:</b> Mask with enhanced contrast</p> <p><b>Output:</b> image segmentation</p>
<p><b>Begin</b></p> <p><b>Step 1:</b> Convert input image from BGR color space to HSV color space</p> <p><b>Step 2:</b> Create a kernel element type an elliptical kernel of size (7, 7)</p> <p><b>Step 3:</b> Perform morphological closing operation / Equations (2.12), (2.13)</p> <p><b>Step 4:</b> Apply dilation process by partially matched between kernel and mask</p> <p style="padding-left: 40px;">If <math>1 \oplus 1 \mid \mid 1 \oplus 0</math> return 1</p> <p style="padding-left: 40px;">Else return 0</p> <p><b>Step 5:</b> Apply erosion process after dilation by total matched between kernel and the result of the dilation process</p> <p style="padding-left: 40px;">If <math>1 \oplus 1</math> return 1</p> <p style="padding-left: 40px;">Else return 0</p> <p><b>Step 6:</b> Convert the resulting HSV image back to the RGB color space</p> <p><b>Step 7:</b> Return image segmentation</p> <p><b>End</b></p>

### 3.7 Feature Extraction

It is to extract the main details of an image, such as its shape, texture, color, and contrast. Using CNN and VGG16, these features are usually represented as matrices containing the extracted values. In other words, the set of numbers reflecting the recovered features serves as a representation of each image. These representations may be in the form of two-dimensional or flat vectors describing features. Converting these features to vectors is important because it allows the data to be presented in a format suitable for models and workbooks. This makes the training and classification process smoother, allows data to be stored more efficiently.

### 3.8 VGG16

- A pre-trained (CNN) architecture (VGG16) available in the Keras library was used to extract features from the images by passing them sequentially through multiple layers.
- VGG16 layers has 13 convolutional layers with ReLU activation function, that analyze features in images, 5 pooling layers, that reduce dimensionality and remove unimportant details, and then reshape the extracted features into a one-dimensional vector, 3 fully connected layers and a linear softmax layer at the output.
- The pre-trained VGG16 model is loaded onto the ImageNet dataset. The weights='ImageNet' parameter ensures that the model is initialized with the weights learned while training on the ImageNet dataset. They

contain the parameter `top = False`, which means that the fully connected layers (top layers) of the VGG16 model are not included.

- VGG16 architecture has a total of around 144 million parameters such as (weight, optimizer, learning rate, number of neurons, number of epochs, and number of convolutional layers).

### **3.9 Classification for Brain Tumor**

The classification determines the classification of different types of tumors (pituitary, glioma, meningioma, or no tumor in normal cases). Two classification methods were used:

#### **3.9.1 Classification Using Combination Model**

In the brain tumor classification process, the combination model (VGG16) with four traditional machine learning classifiers (SVM, DT, RF, and NB) to classify model.

Features are taken from the images and used in the VGG16 model. The images are converted into vectors through the model and the next step is to feed the extracted features into one of Machine learning classification algorithms (SVM, NB, DT, and RF) to predict categories such as (glioma, pituitary, meningioma, and no tumor in case normal). These models learn to classify tumors based on extracted features. It's seen in Figure (3.2).

The main idea behind this model is to take advantage the unique capabilities of each model (deep learning and traditional machine learning) to improve brain tumor classification and detection performance.

After applying these algorithms, showed RF gives the best result in terms of the accuracy of tumor classification of hybrid model.

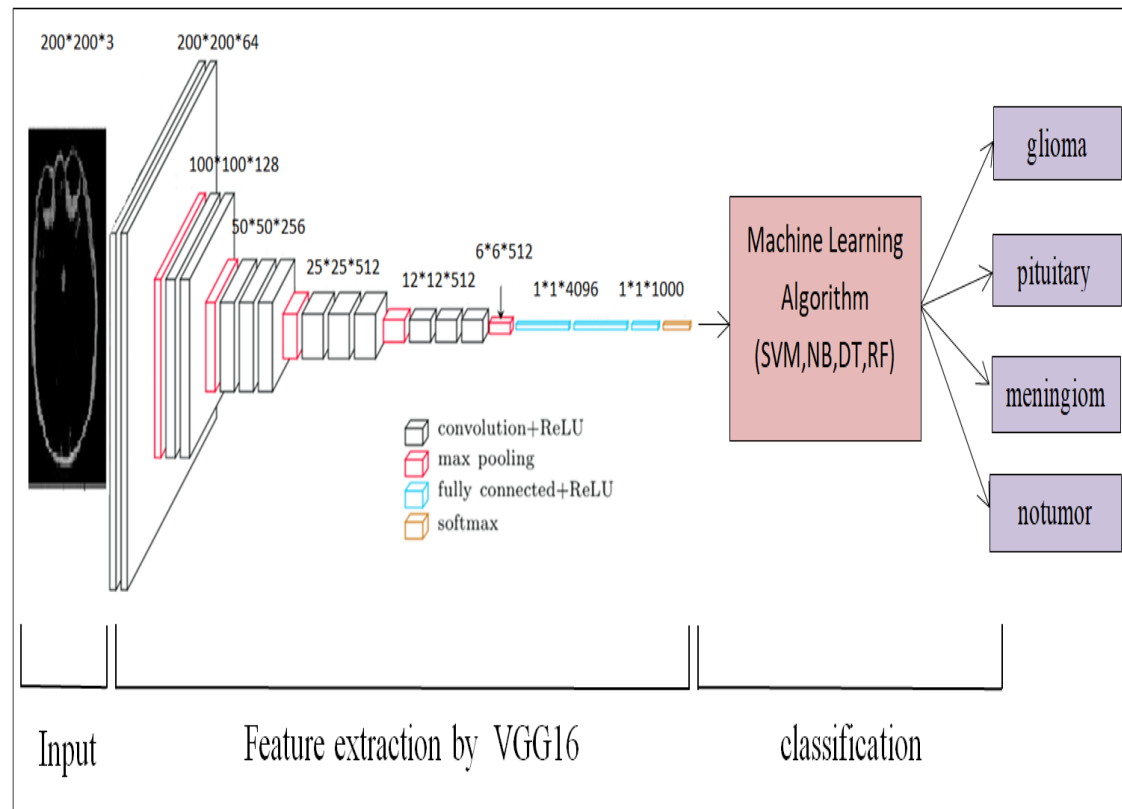


Figure 3.2: Combination Model (VGG16+Machine Learning Algorithms)

### 3.9.2 Classification Using CNN Model

- The CNN model was trained, and different number of parameters such as the number of epochs were run, and the results were obtained using the CNN model for layers with different parameter values. The CNN model has several steps to classify MRI images of brain tumors.
- The size and number of filters, activation function, and size of the input image are the four parameters that need to be provided to the first

---

convolutional layer. The main goal of convolution processing is to find features in the image using filters and then store them in feature maps.

- Using the basic design, the modified network consists of six alternating Conv2D layers (with ReLU enabled) and MaxPooling2D layers with a filter size of 2x2.
- 32 filters of 3x3 pixel size are used in the first layer. In the second layer, 64 filters of the same size are used. The first and second layers use a ReLU activation function and a  $2 \times 2$  filter size for maxpooling2D.
- The third layer uses the same size of 32 filters, the fourth layer uses the same size of 64 filters, and five layer use the same size of 32 filters, then 64 in a layer with 6 Conv2D filters.
- Next, the flattening layer flattens the matrix after six convolutional layers.
- The classification layer is determined by the dense layer in the final fully connected layer. To locate test images in the output layer, Softmax is used. Figure (3.3) shows the Architecture of CNN model.



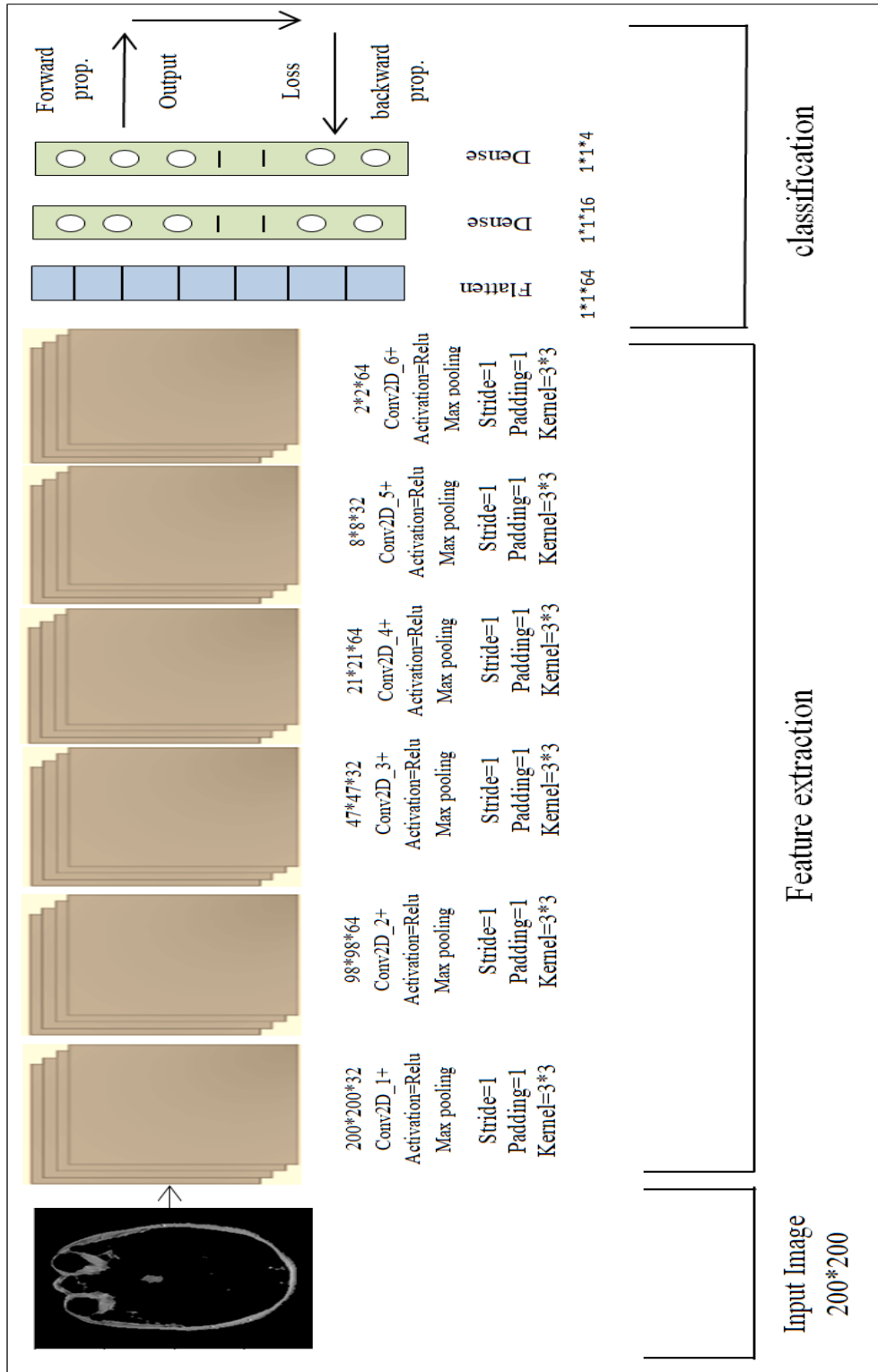


Figure 3.3: Architecture of the CNN model

**3.10 Summary**

This chapter explores the architectural features and learning settings used to design the proposed model. It also describes how to split the data into training and testing.

**CHAPTER FOUR**

**RESULTS AND DISCUSSION**

## 4.1 Overview

In this chapter includes a summary of the results of the suggested system representation (Automated detection and classification of human brain tumor using machine learning), where all the results were discussed and analyzed, and tests were conducted on them, in addition to comparisons with previous research and studies.

## 4.2 Experimental Setting

The proposed system was implemented on a calculator MSI CPU 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz RAM 16.0 GB system type 64-bit Windows 11 Pro N and the programming language used (Python / ANACONDA / Jupyter environment version 3.9.12 64-bit).

## 4.3 Dataset Description

The dataset used to implement the proposed system is Brain Tumor MRI Dataset. This dataset has been chosen due to it is intended to detect different types of brain tumors. This dataset is originally a combination of three datasets: Figshare, the SARTAJ data set, and the Br35H dataset. Glioma, meningioma - without tumor, and pituitary are the four categories that were classified from the 7023 human brain MRI images in this dataset. No tumor class images were taken from the Br35H dataset. The SARTAJ dataset had a problem: the glioma category images were not labeled correctly, so the images on Figshare were used

instead. This dataset is located on the following website:

<https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>. Figure 4.1 represents samples from the data set.

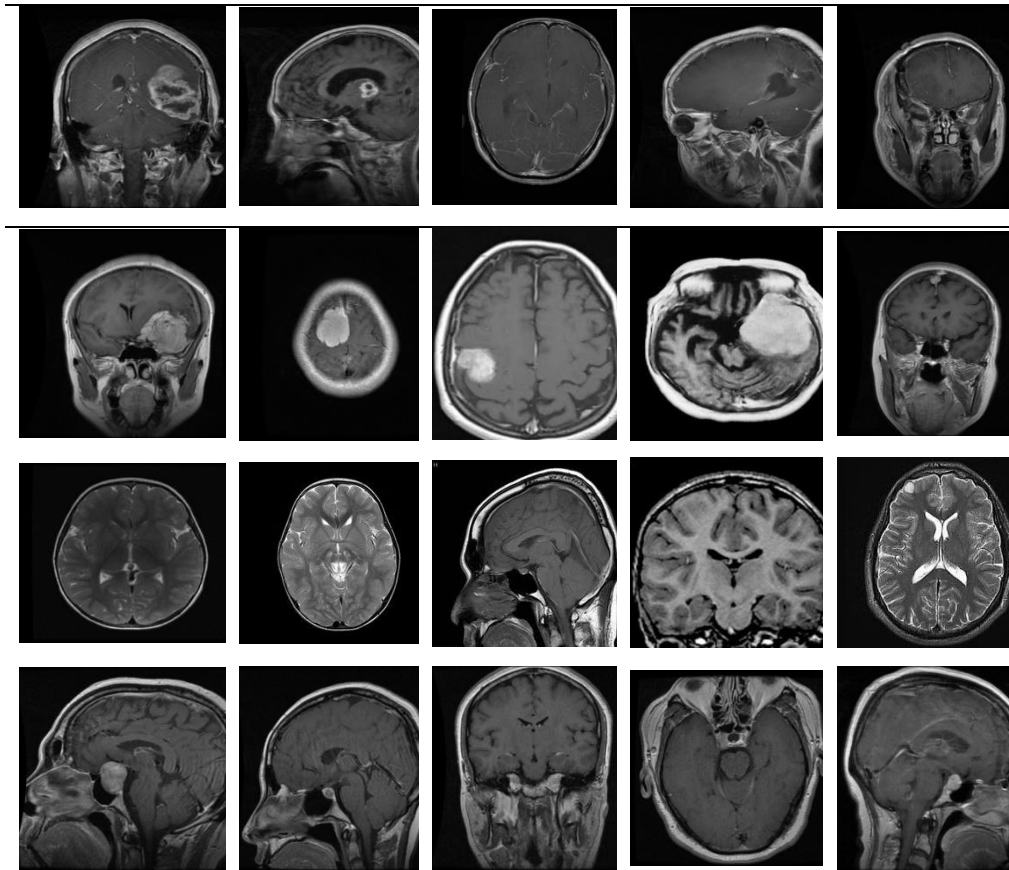


Figure 4.1: Sample for dataset[79]

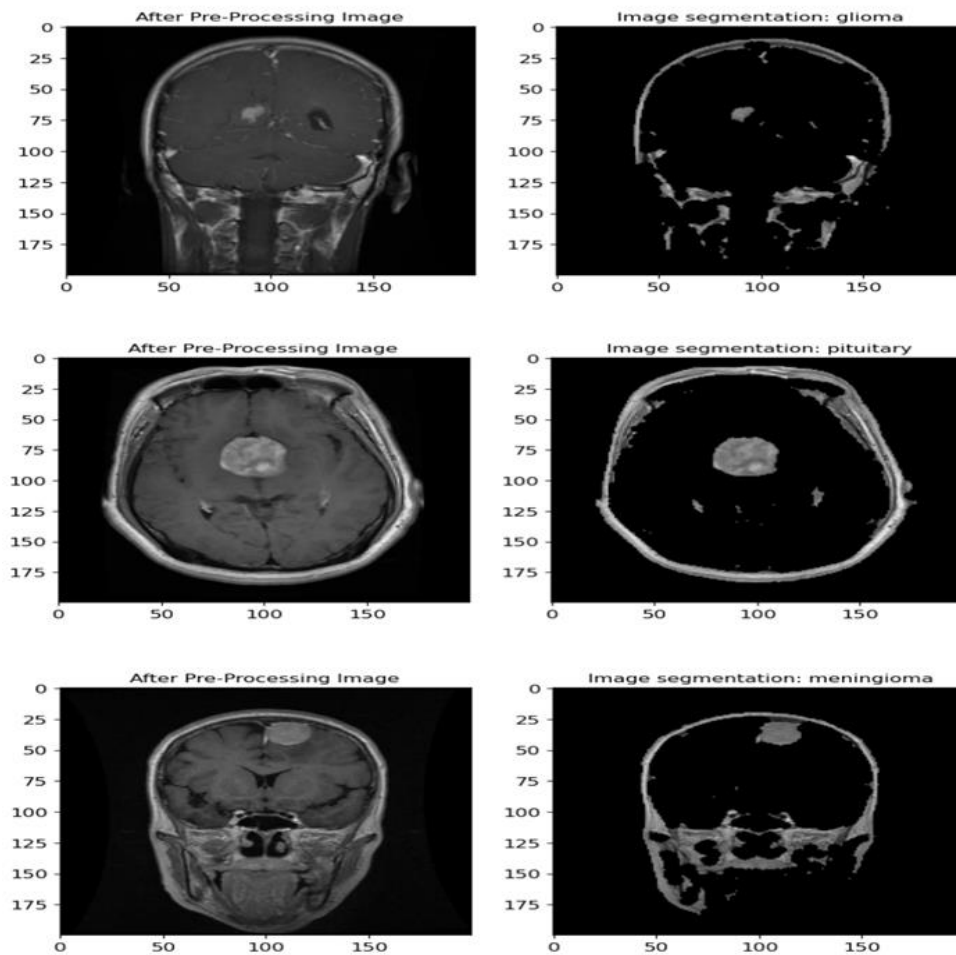
## 4.4 Experimental Results

Experimental results for the two proposed models were applied using segmentation techniques for tumor detection.

### 4.4.1 Experiment of Region-Based Segmentation

Results were obtained after applying split and merge techniques on the MRI images resulting from the preprocessing stage. Figure (4.2)

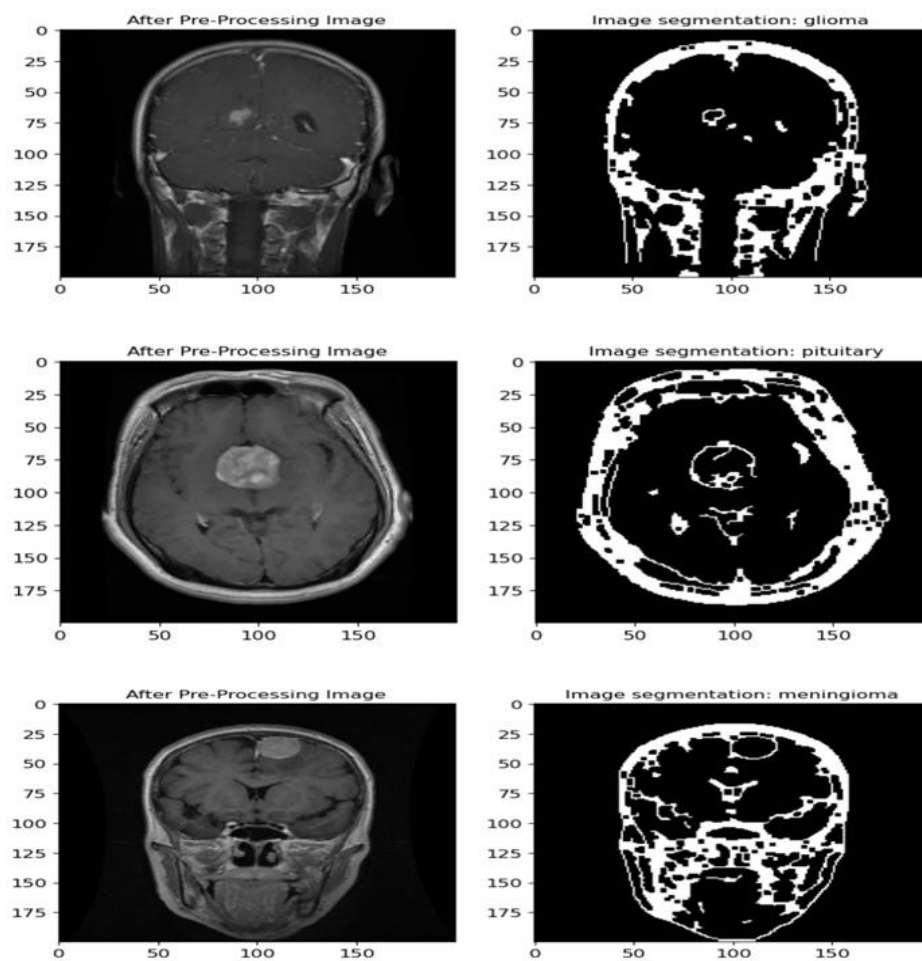
illustrates the segmentation image using region-based segmentation.



*Figure 4.2: Displays an example-segmented image using region-based segmentation*

#### **4.4.2 Experiment of Edge-Based Segmentation**

Results obtained after applying canny edge detection techniques on the images MRI resulting from the preprocessing stage. Figure (4.3) illustrates the segmentation image using edge-based segmentation.



*Figure 4.3: Displays an example segmented image using edge-based segmentation*

The results using region-based segmentation are better than the results using edge-based segmentation based on accuracy.

## 4.5 Results and Discussion

Results acquired from the implementation of the two proposed systems (the combination model and the CNN model) were presented on the datasets available on the Internet.

### 4.5.1 Results of Combination Model

Four traditional machine learning algorithms were used (SVM, NB, DT, RF), and performance of each classifier is measured by calculating accuracy from the confusion matrix using an MRI image.

#### 4.5.1.1 Results of Combination Model Using Region-Based Segmentation and Edge-Based Segmentation

The combination model was applied (VGG16 with machine learning classifiers (SVM, NB, DT, RF)) using region-based segmentation and edge-based segmentation .

The result classification report of Combination model (VGG16 with SVM) obtained as shown in Tables (4.1), and (4.2).

*Table 4.1: The performance evaluation of Combination model (VGG16 with SVM) using region-based segmentation*

Classifiers	Tumor Type	Precision	Recall	F1-Score	Support
<b>VGG16 + SVM</b>	glioma	0.95	0.95	0.95	300
	meningioma	0.94	0.92	0.93	306
	no tumor	0.99	0.99	0.99	405
	pituitary	0.95	0.98	0.96	300

*Table 4.2: The performance evaluation of Combination model(VGG16 with SVM) using edge-based segmentation*

Classifiers	Tumor Type	Precision	Recall	F1-Score	Support
<b>VGG16 + SVM</b>	glioma	0.93	0.94	0.94	300
	meningioma	0.92	0.84	0.88	306
	no tumor	0.96	0.99	0.97	405
	pituitary	0.93	0.97	0.95	300



In confusion matrix columns represent actual values and rows represent predicted values. Figures (4.4), and (4.5) show the results of the confusion matrices for combination model (VGG16 with SVM)

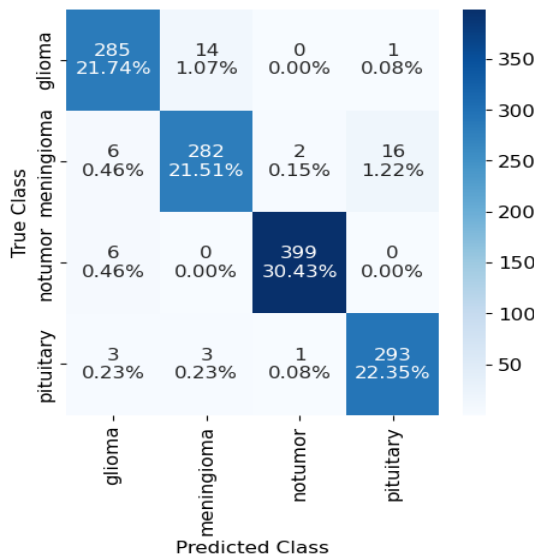


Figure 4.4: confusion\_matrix of Combination model (VGG16 with SVM) using region-based segmentation

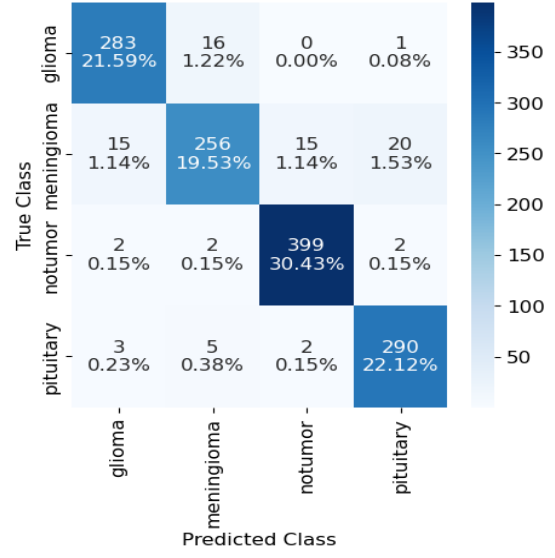


Figure 4.5: confusion\_matrix of Combination model (VGG16 with SVM) using edge-based segmentation

- In Figure (4.4), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 399 images. The highest FP is at the intersection of the pituitary row and the pituitary column, where the model predicted incorrectly with 16 images. The highest FN is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 16 images.
- In Figure (4.5), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 399 images. The highest FP is at the intersection of the pituitary row and

the pituitary column, where the model predicted incorrectly with 20 images. The highest FN is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 20 images.

The result classification report of combination model (VGG16 with DT) obtained as shown in Tables (4.3), and (4.4).

*Table 4.3: The performance evaluation of Combination model (VGG16 with DT) using region-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>VGG16 + DT</b>	glioma	0.90	1.00	0.95	300
	meningioma	0.91	1.00	0.95	306
	no tumor	0.98	0.94	0.96	405
	pituitary	0.97	0.81	0.88	300

*Table 4.4: The performance evaluation of Combination model(VGG16 with DT) using edge-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>VGG16 + DT</b>	glioma	0.90	1.00	0.94	300
	meningioma	0.85	1.00	0.92	306
	no tumor	0.95	0.93	0.94	405
	pituitary	0.99	0.72	0.83	300

Figures (4.6), and (4.7) show the results of the confusion matrices for Combination model (VGG16 with DT)

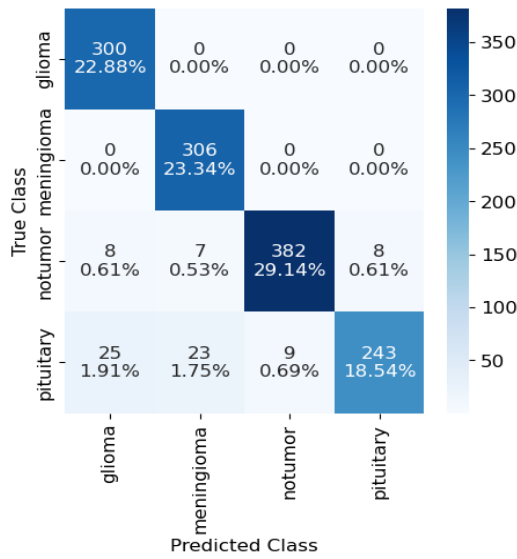


Figure 4.6: confusion\_matrix of Combination model (VGG16 with DT) using region-based segmentation

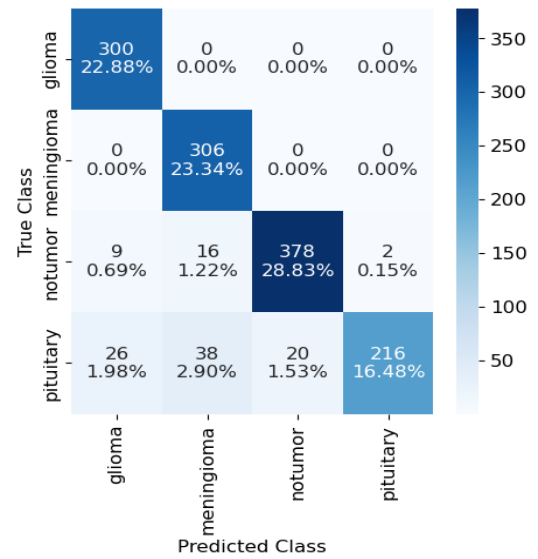


Figure 4.7: confusion\_matrix of Combination model (VGG16 with DT) using edge-based segmentation

- In Figure (4.6), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 382 images. The highest FP is at the intersection of the glioma row and the glioma column, where the model predicted incorrectly with 25 images. The highest FN is at the intersection of the pituitary row and the pituitary column, where the model predicted incorrectly with 25 images.
- In Figure (4.7), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 378 images. The highest FP is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 38 images. The highest FN is at the intersection of the pituitary row and the pituitary column, where the model predicted incorrectly with 38 images.

The result classification report of Combination model (VGG16 with RF) obtained as shown in Tables (4.5), and (4.6).

*Table 4.5: The performance evaluation of Combination model (VGG16 with RF) using region-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>VGG16 + RF</b>	glioma	0.99	1.00	0.99	300
	meningioma	0.98	1.00	0.99	306
	no tumor	1.00	1.00	1.00	405
	pituitary	1.00	0.97	0.98	300

*Table 4.6: The performance evaluation of Combination model(VGG16 with RF) using edge-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>VGG16 + RF</b>	glioma	0.98	1.00	0.99	300
	meningioma	0.98	1.00	0.99	306
	no tumor	0.99	1.00	1.00	405
	pituitary	1.00	0.95	0.97	300

Figures (4.8), and (4.9) show the results of the confusion matrices for combination model (VGG16 with RF).

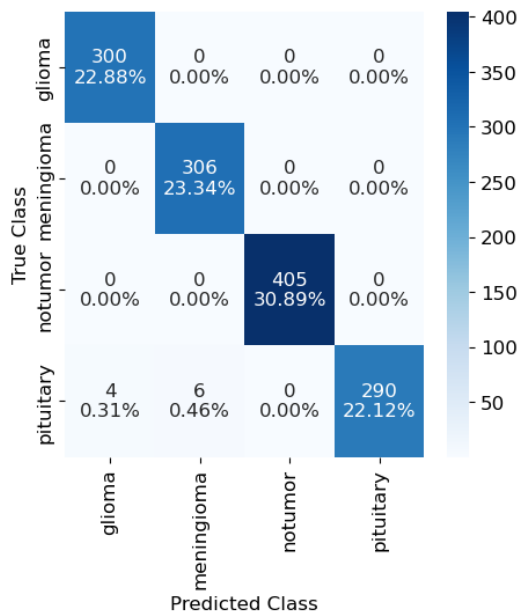


Figure 4.8: confusion\_matrix of Combination model (VGG16 with RF) using region-based segmentation

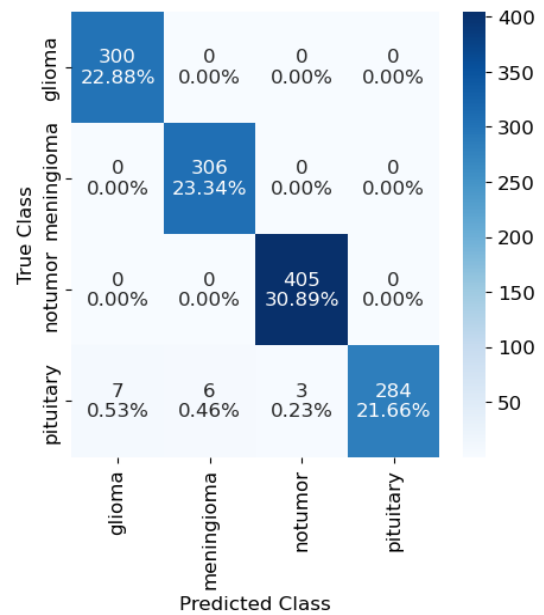


Figure 4.9: confusion\_matrix of Combination model (VGG16 with RF) using edge-based segmentation

- In Figure (4.8), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 405 images. The highest FP is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 6 images. The highest FN is at the intersection of the pituitary row and the pituitary column, where the model predicted incorrectly with 6 images.
- In Figure (4.9), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 405 images. The highest FP is at the intersection of the glioma row and the glioma column, where the model predicted incorrectly with 7

images. The highest FN is at the intersection of the pituitary row and the pituitary column, where the model predicted incorrectly with 7 images.

The result classification report of Combination model (VGG16 with NB) obtained as shown in Tables (4.7), and (4.8).

*Table 4.7: The performance evaluation of Combination model(VGG16 with NB) using region-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>VGG16 + NB</b>	glioma	0.76	0.99	0.86	300
	meningioma	0.74	0.68	0.71	306
	no tumor	0.95	0.90	0.92	405
	pituitary	0.89	0.77	0.82	300

*Table 4.8: The performance evaluation of Combination model(VGG16 with NB) using edge-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
<b>VGG16 + NB</b>	glioma	0.73	0.99	0.84	300
	meningioma	0.74	0.65	0.69	306
	no tumor	0.96	0.85	0.90	405
	pituitary	0.83	0.76	0.79	300

Figures (4.10), and (4.11) show the results of the confusion matrices for Combination model (VGG16 with NB)

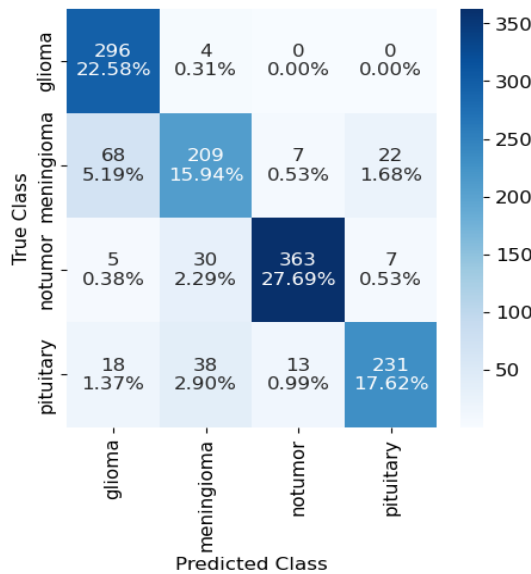


Figure 4.10: confusion\_matrix of Combination model (VGG16 with NB) using region-based segmentation

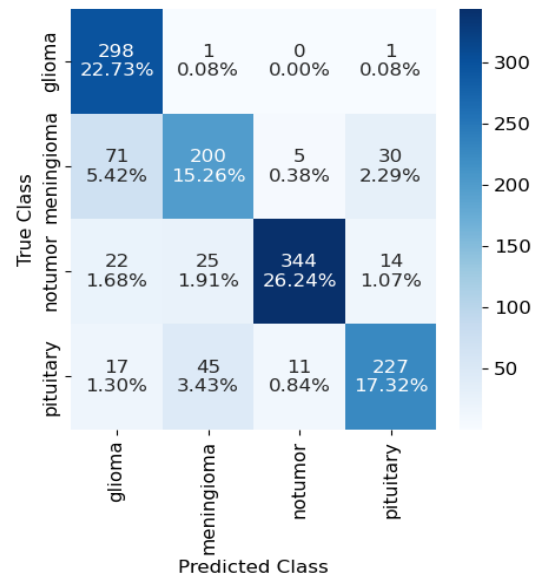


Figure 4.11: confusion\_matrix of Combination model (VGG16 with NB) using edge-based segmentation

- In Figure (4.10), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 363 images. The highest FP is at the intersection of the glioma row and the glioma column, where the model predicted incorrectly with 68 images. The highest FN is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 68 images.
- In Figure (4.11), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 344 images. The highest FP is at the intersection of the glioma row and the glioma column, where the model predicted incorrectly with 71 images. The highest FN is at the intersection of the meningioma row and

the meningioma column, where the model predicted incorrectly with 71 images.

#### 4.5.2 Results of CNN Model

The metrics were applied to a variety of training and testing samples while taking into consideration the architecture of the neural network as well as number of convolution layers that were included in the deep model. As a direct consequence of the application of these criteria, the model is able to classify MRI images of brain tumors.

A variety of experiments were carried out in an effort to achieve the best possible outcome with the CNN model by change different parameters for exemplar the number of epochs.

##### 4.5.2.1 Results of CNN Model Using Region-Based Segmentation and Edge-Based Segmentation

Some experiments represent the relation between the learning rate, the number of epochs, and its level of accuracy are laid out in table (4.9).

*Table 4.9: Accuracy of the CNN model with region-based segmentation and Edge-Based Segmentation*

Segmentation Method	Learning Rate	Number of Epoch	Accuracy
Region-based segmentation	0.001	10	95.04%
		30	96.95%
		90	98.93%
Edge-Based Segmentation	0.001	10	95.27%
		30	95.65%
		90	98.55%



When the segmentation method is region-based segmentation, 0.001 is the learning rate, and 90 is the epoch, the CNN model obtained the best accuracy, which was 98.93%. This percentage is higher compared to the CNN model when using edge-based segmentation.

The result classification report of CNN model obtained as shown in Tables (4.10), and (4.11).

*Table 4.10: The performance evaluation of CNN model using region-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
CNN	glioma	1.00	0.99	1.00	300
	meningioma	1.00	0.97	0.99	306
	no tumor	0.97	1.00	0.98	405
	pituitary	1.00	0.99	0.99	300

*Table 4.11: performance evaluation of CNN model using edge-based segmentation*

<b>Classifiers</b>	<b>Tumor Type</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Support</b>
CNN	glioma	1.00	1.00	1.00	300
	meningioma	0.94	1.00	0.97	306
	no tumor	1.00	0.99	1.00	405
	pituitary	1.00	0.95	0.97	300

Figures (4.12), (4.13), (4.14), and (4.15) show the accuracy and loss plot of CNN model.

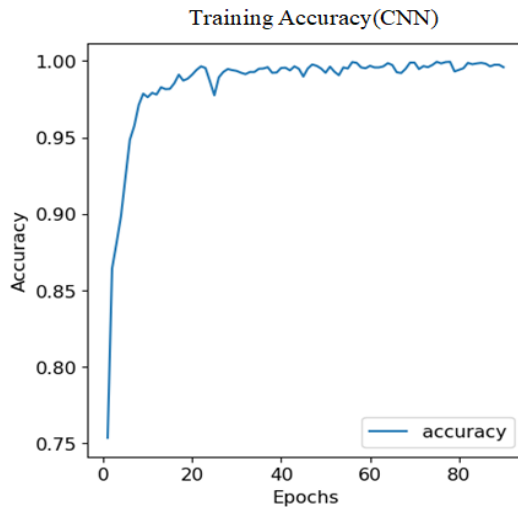


Figure 4.12: Accuracy function of CNN model using region-based segmentation

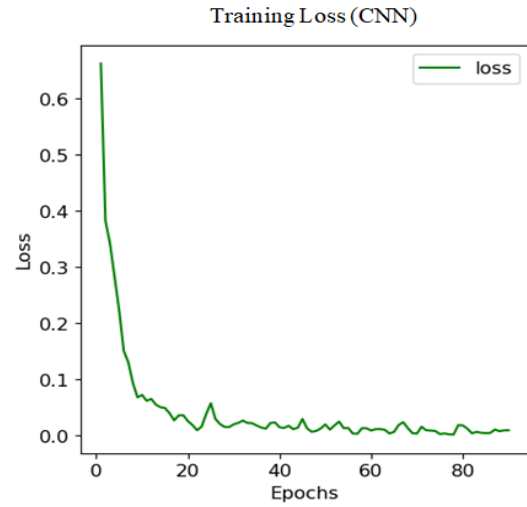


Figure 4.13: Loss function of CNN model using region-based segmentation

- In Figure (4.12), the accuracy increases with the increase in the number of epoch using region-based segmentation.
- In Figure (4.13), the loss decreases with the increase in the number of epoch using region-based segmentation.

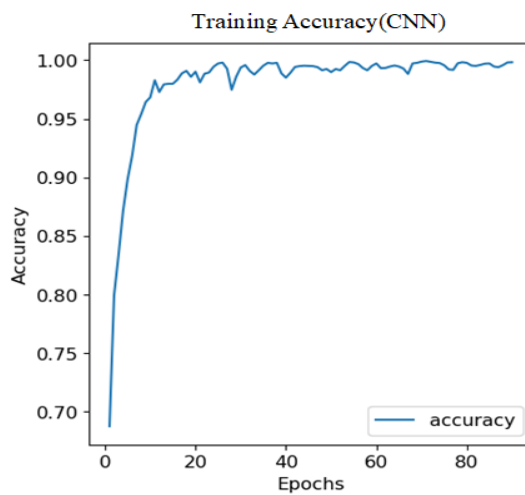


Figure 4.14: Accuracy function of CNN model using edge-based segmentation

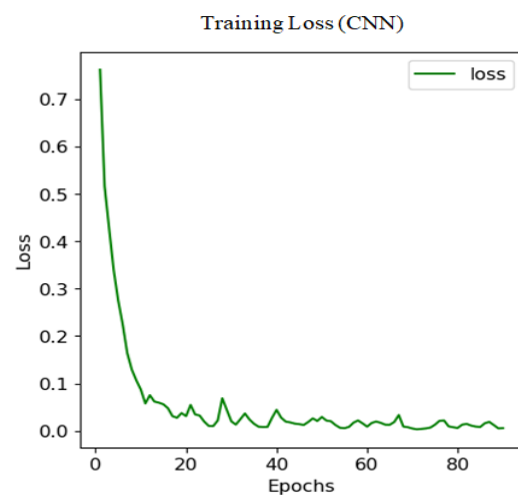


Figure 4.15: Loss function of CNN model using edge-based segmentation

- In Figure (4.14), the accuracy increases with the increase in the number of epoch using edge-based segmentation.
- In Figure (4.15), the loss decreases with the increase in the number of epoch using edge-based segmentation..

The result of the confusion matrix was also obtained for this model as shown in the figure below (4.14).

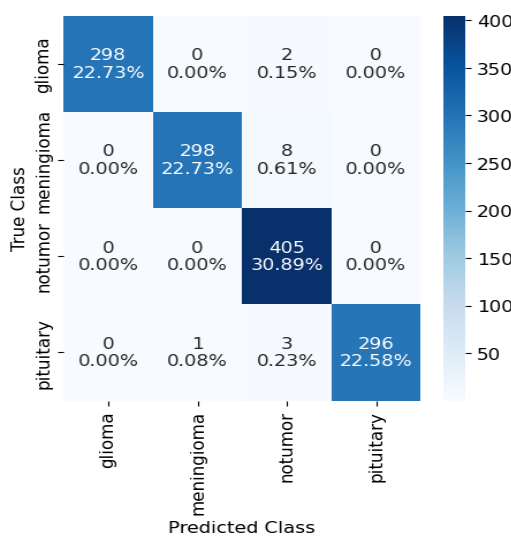


Figure 4.16: confusion\_matrix of CNN model using region-based segmentation

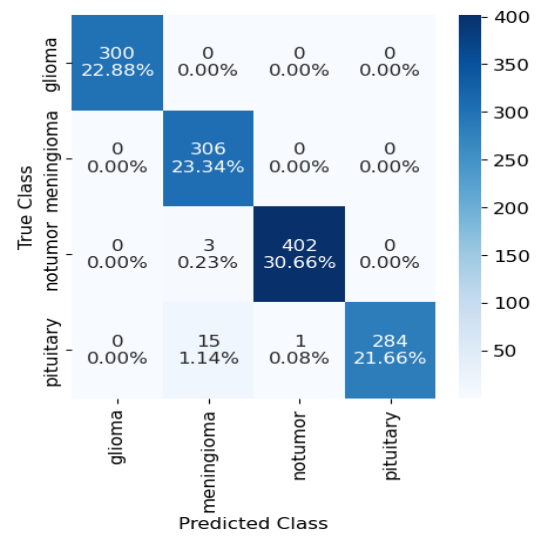


Figure 4.17: confusion\_matrix of CNN model using edge-based segmentation

- In Figure (4.16), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 405 images. The highest FP is at the intersection of the notumor row and the glioma column, where the model predicted incorrectly with 0 images. The highest FN is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 8 images.

- In Figure (4.17), the highest TP is at the intersection of the notumor row and the notumor column, where the model predicted correctly with 402 images. The highest FP is at the intersection of the meningioma row and the meningioma column, where the model predicted incorrectly with 15 images. The highest FN is at the intersection of the pituitary row and the pituitary column, where the model predicted incorrectly with 15 images.

#### 4.6 Performance Comparison between Traditional Machine Learning Classifiers

Accuracy results are compared between traditional machine learning classifiers (SVM, NB, DT, and RF) using two segmentation methods region-based segmentation and edge-based segmentation as shown in Tables (4.12) and (4.13).

*Table 4.12: Accuracy comparison among traditional machine learning classifiers in combination model (VGG16 with ML Classifiers) by using region-based segmentation*

Segmentation Method	Combination Model	Accuracy
Region-Based Segmentation	VGG16 + SVM	96.03%
	VGG16 + DT	93.90%
	VGG16 + RF	99.24%
	VGG16 + NB	83.83%

*Table 4.13: Accuracy comparison among traditional machine learning classifiers in combination model (VGG16 with ML Classifiers) by using edge-based segmentation*

Segmentation Method	Combination Model	Accuracy
Edge-Based Segmentation	VGG16 + SVM	93.67%
	VGG16 + DT	91.53%
	VGG16 + RF	98.78%
	VGG16 + NB	81.54%

When the segmentation method is region-based segmentation, we obtained the best accuracy in the random forest: 99.24%.

#### 4.7 Performance Comparison between Combination Model and CNN Model

The two proposed classification models (Combination model and CNN model) based on two different segmentation methods were compared. Tables (4.14) and (4.15) show the comparison between of these two models.

*Table 4.14: Performance comparison between the Combination model and CNN model using region-based segmentation*

Segmentation Method	Model	Machine Learning	Accuracy
Region-Based Segmentation	Combination model (VGG16+ML Algorithm)	SVM	96.03%
		DT	93.90%
		RF	99.24%
		NB	83.83%
Region-Based Segmentation	CNN model Epoch=90	CNN	98.93%

*Table 4.15: Performance comparison between the Combination model and CNN model using edge-based segmentation*

Segmentation Method	Model	Machine Learning	Accuracy
Edge-Based Segmentation	Combination model (VGG16+ML Algorithm)	SVM	93.67%
		DT	91.53%
		RF	98.78%
		NB	81.54%
Edge-Based Segmentation	CNN model Epoch=90	CNN	98.55%

#### 4.8 Comparison with the most modern models

Results of the model were compared with those of comparable works performed on the same dataset, and the classification was used to find a solution to the issue of the classification of brain tumor .[81], [79], and [80]. Table (4.16) displays these results.

*Table 4.16: Comparison of proposed model and related research*

Author	Dataset	Methodology	Accuracy
Ullah at el. [81]	figshare, SARTAJ dataset, and Br35H	extracted features combined of Gabor and ResNet50 and then classified by SVM	95.73%
Filatov at el. [79]	figshare, SARTAJ dataset, and Br35H	ResNet50, EfficientNetB1, EfficientNetB7, and EfficientNetV2B1 are some of the CNN networks that have been employed in the training process. The results were best for EfficientNetB1 .	87.67%

Gómez at el. [80]	figshare, SARTAJ dataset, and Br35H	Generic CNN, ResNet50, InceptionV3, InceptionResNetV2, Xception, MobileNetV2, and EfficientNetB0. The best CNN model was InceptionV3	97.12%
Our proposed model	figshare, SARTAJ dataset, and Br35H	Tumor detection by segmentation and classification by two proposed models. The first is the combination model (Vgg16 with traditional machine learning algorithm (NB, SVM, DT, and RF)) using region-based segmentation; the model achieved accuracy (83.83%, 93.90%, 96.03%, 99.24%) respectively. As for using edge-based segmentation, the model achieved accuracy (81.54%, 91.53%, 93.67%, 98.78%) respectively. The second model is the convolutional neural network (CNN) using region-based segmentation, where the model achieved an accuracy of (98.93%), while using edge-based segmentation, the model achieved an accuracy of (98.55%). The best results obtained were the combination model (Vgg16 with RF) using region-based segmentation by ratio 99.24%.	99.24%

## 4.9 Summary

In this chapter, results and analysis of brain tumor classification performance are presented. Four performance metrics were calculated. The conclusion and further works are going to be discussed in the following chapter.

## **CHAPTER FIVE**

# **CONCLUSION AND FUTURE WORK**



## 5.1 Overview

This chapter briefly summarizes the results and conclusions of the work, as well as some suggestions for further research.

## 5.2 Conclusion

Brain tumors have become one of the most severe problems today. Early detection of brain tumors reduces the number of deaths.

1. This study showed that the pre-processing stage is important to improve the brightness, sharpness of edges, and contrast of pixels in the image, and this leads to improving the accuracy of the model.
2. The image segmentation stage necessarily facilitates work, detects the tumor, and improves the accuracy of the model. Two segmentation methods were used: (edge-based segmentation and region-based segmentation). The results of region-based segmentation were better than the results of edge-based segmentation, as in Tables (4.15) and (4.14)
3. In this work, two proposed models were applied to classify types of brain tumors in MRI images, the first is a combination model (VGG16) with four traditional classifiers, namely Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Nave Byes (NB). The second model is the Convolutional Neural Network (CNN).
4. The results were compared between traditional classifiers as in Tables (4.12) and (4.13). The results of the combination model were then compared with the results of the CNN model. The result of the

comparative combination model is more accurate than the CNN model, as in Tables (4.15)(4.14).

5. The combination model (VGG16 with Random Forest classifier) using region-based segmentation obtained the highest accuracy of 99.24%, as shown in the Confusion Matrix in Figure (4.8).

6. Finally, our results were compared with existing research in the field of segmentation and classification, as in Table (4.16), where our results proved to be the best.

### **5.3 Future Work**

Because there is still area for development to acquire improved accuracy, there are more prospects for advancement or study in our future work.

1. Can improve the system by finding out the size of and growth rate tumor which will help the radiologist in making decisions.
2. Image background can be removed, it will reduce time and cost.
3. As a further development of the model, combination CNN can be used and to enhance effectiveness of the CNN model.
4. The proposed method can also be used for other medical imaging classification, such as lung, breast, and colon cancer.

## **REFERENCES**

## REFERENCES

- [1] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, "Classification using deep learning neural networks for brain tumors," *Futur. Comput. Informatics J.*, vol. 3, no. 1, pp. 68–71, 2018, doi: 10.1016/j.fcij.2017.12.001.
- [2] K. Sharma, A. Kaur, and S. Gujral, "Brain Tumor Detection based on Machine Learning Algorithms," *Int. J. Comput. Appl.*, vol. 103, no. 1, pp. 7–11, 2014, doi: 10.5120/18036-6883.
- [3] G. Hemanth, M. Janardhan, and L. Sujihelen, "Design and implementing brain tumor detection using machine learning approach," *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, vol. 2019-April, no. Icoei, pp. 1289–1294, 2019, doi: 10.1109/icoei.2019.8862553.
- [4] S. S. R. J. Seetha, "Brain Tumor Classification Using Convolutional Neural Networks," *Biomed. Pharmacol. Journal, Sept. 2018.*, vol. 11, no. September, pp. 1457–1461, 2018, doi: 10.1109/EMES52337.2021.9484102.
- [5] N. Niharika, S. Patel, K. P. Bharath, B. Subramanian, and M. Rajesh Kumar, "Brain tumor detection and classification based on histogram equalization using machine learning," *Handb. Res. Deep Learn. Image Anal. Under Constrained Unconstrained Environ.*, vol. 9, no. 7, pp. 23–43, 2020, doi: 10.4018/978-1-7998-6690-9.ch002.
- [6] G. Garg and R. Garg, "Brain Tumor Detection and Classification based on Hybrid Ensemble Classifier," no. 3, pp. 1–18, 2021. doi.org/10.48550/arXiv.2101.00216
- [7] V.A.R.Barao, R.C.Coata, J.A.Shibli, M.Bertolini, and J.G.S.Souza, "Brain Tumor Detection Using CNN," *Braz Dent J.*, vol. 33, no. 1, pp. 1–12, 2022.doi.org/10.22214/ijraset
- [8] J. Amin, M. Sharif, A. Haldorai, M. Yasmin, and R. S. Nayak, "Brain tumor detection and classification using machine learning: a comprehensive survey," *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3161–3183, 2022, doi: 10.1007/s40747-021-00563-y.
- [9] A. Sravanthi Peddinti, S. Maloji, and K. Manepalli, "Evolution in diagnosis and detection of brain tumor – review," *J. Phys. Conf. Ser.*, vol. 2115, no. 1, 2021, doi: 10.1088/1742-6596/2115/1/012039.
- [10] A. Raza *et al.*, "A Hybrid Deep Learning-Based Approach for Brain Tumor Classification," *Electron.*, vol. 11, no. 7, pp. 1–17, 2022, doi: 10.3390/electronics11071146.

- [11] M. resonance Imaging and it also helps to detect abnormalities in the brain at earlier stages than other imaging modalities [22]. The MRI scan is used to completely analyze different body parts, *Brain Tumor Segmentation Using K-means – FCM Hybrid Technique*. Springer Singapore. doi: 10.1007/978-981-10-7386-1.
- [12] V. Jalali and D. Kaur, “A study of classification and feature extraction techniques for brain tumor detection,” *Int. J. Multimed. Inf. Retr.*, vol. 9, no. 4, pp. 271–290, 2020, doi: 10.1007/s13735-020-00199-7.
- [13] C. Srinivas *et al.*, “Deep Transfer Learning Approaches in Performance Analysis of Brain Tumor Classification Using MRI Images,” *J. Healthc. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/3264367.
- [14] S. Molaei, N. Ghorbani, F. Dashtiahangar, M. Peivandi, Y. Pourasad, and M. Esmaeili, “FDCNet: Presentation of the Fuzzy CNN and Fractal Feature Extraction for Detection and Classification of Tumors,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/7543429.
- [15] A. Sinha, A. R P, M. Suresh, N. Mohan R, D. Abinaya, and A. G. Singerji, “Brain Tumour Detection Using Deep Learning,” *Proc. 2021 IEEE 7th Int. Conf. Bio Signals, Images Instrumentation, ICBSII 2021*, 2021, doi: 10.1109/ICBSII51839.2021.9445185.
- [16] G. Siracusano, A. La Corte, M. Gaeta, G. Cicero, M. Chiappini, and G. Finocchio, “Pipeline for advanced contrast enhancement (Pace) of chest x-ray in evaluating covid-19 patients by combining bidimensional empirical mode decomposition and contrast limited adaptive histogram equalization (clahe),” *Sustain.*, vol. 12, no. 20, pp. 1–18, 2020, doi: 10.3390/su12208573.
- [17] A. Gumaei, M. M. Hassanf(x-y), M. R. Hassan, A. Alelaiwi, and G. Fortino, “A Hybrid Feature Extraction Method with Regularized Extreme Learning Machine for Brain Tumor Classification,” *IEEE Access*, vol. 7, pp. 36266–36273, 2019, doi: 10.1109/ACCESS.2019.2904145.
- [18] T. D. Pham, “Kriging-Weighted Laplacian Kernels for Grayscale Image Sharpening,” *IEEE Access*, vol. 10, pp. 57094–57106, 2022, doi: 10.1109/ACCESS.2022.3178607.
- [19] L. Li, D. Wu, J. Wu, H. Li, W. Lin, and A. C. Kot, “Image Sharpness Assessment by Sparse Representation,” *IEEE Trans. Multimed.*, vol. 18, no. 6, pp. 1085–1097, 2016, doi: 10.1109/TMM.2016.2545398.
- [20] E. Hassan and A. Aboshgifa, “Detecting Brain Tumour from Mri Image Using Matlab GUI Programme,” *Int. J. Comput. Sci. Eng. Surv.*, vol. 6, no. 6, pp. 47–60, 2015, doi: 10.5121/ijcses.2015.6604.

- [21] R. Kaur and A. Doegar, "Localization and classification of brain tumor using machine learning & deep learning techniques," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 9, pp. 59–66, 2019, doi: 10.35940/ijitee.I1010.0789S19.
- [22] A. Z. H. Ooi *et al.*, "Interactive blood vessel segmentation from retinal fundus image based on canny edge detector," *Sensors*, vol. 21, no. 19, pp. 1–22, 2021, doi: 10.3390/s21196380.
- [23] E. A. Sekehravani, E. Babulak, and M. Masoodi, "Implementing canny edge detection algorithm for noisy image," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1404–1410, 2020, doi: 10.11591/eei.v9i4.1837.
- [24] . M. Z., . A. A. S., . S. H. S., . A. Z. J., and . A. K., "Brain Tumor Segmentation through Region-based, Supervised and Unsupervised Learning Methods: A Literature Survey Brain Tumor Segmentation through Image Processing Methods: A Literature Survey," *J. Biomed. Eng. Med. Imaging*, vol. 6, no. 2, 2019, doi: 10.14738/jbemi.62.6725.
- [25] M. S. H. Al-Tamimi and G. Sulong, "Tumor brain detection through MR images: A review of literature," *J. Theor. Appl. Inf. Technol.*, vol. 62, no. 2, pp. 387–403, 2014, doi: 1817/3195
- [26] Y. Kimori, "Morphological image processing for quantitative shape analysis of biomedical structures: Effective contrast enhancement," *J. Synchrotron Radiat.*, vol. 20, no. 6, pp. 848–853, 2013, doi: 10.1107/S0909049513020761.
- [27] A. Gurunathan and B. Krishnan, "Detection and diagnosis of brain tumors using deep learning convolutional neural networks," *Int. J. Imaging Syst. Technol.*, vol. 31, no. 3, pp. 1174–1184, 2021, doi: 10.1002/ima.22532.
- [28] M. I. Processing, "2 Morphological Image Processing," pp. 30–62.
- [29] T. O. Ayodele, "Atherosclerotic Cardiovascular Disease," *Atheroscler. Cardiovasc. Dis.*, 2012, doi: 10.5772/711.
- [30] M. Batta, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res.*, vol. 18, no. 8, pp. 381–386, 2018, doi: 10.21275/ART20203995.
- [31] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A Survey on Distributed Machine Learning," *ACM Comput. Surv.*, vol. 53, no. 2, 2020, doi: 10.1145/3377454.
- [32] O. F.Y, A. J.E.T, A. O, H. J. O, O. O, and A. J, "Supervised Machine Learning Algorithms: Classification and Comparison," *Int. J. Comput. Trends Technol.*, vol. 48, no. 3, pp. 128–138, 2017, doi: 10.14445/22312803/ijctt-v48p126.

- [33] M. F. Hassan and M. E. Manaa, "Big Data Processing with Hadoop and Data Mining," *HORA 2022 - 4th Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc.*, no. June 2022, pp. 1–8, 2022, doi: 10.1109/HORA55278.2022.9800085.
- [34] S. Anjum, L. Hussain, M. Ali, A. A. Abbasi, and T. Q. Duong, "Automated multi-class brain tumor types detection by extracting RICA based features and employing machine learning techniques," *Math. Biosci. Eng.*, vol. 18, no. 3, pp. 2882–2908, 2021, doi: 10.3934/MBE.2021146.
- [35] J. Sen, *Machine Learning Algorithms, Models and Applications Edited by Jaydip Sen*, vol. 7, 2021. [Online]. Available: <http://dx.doi.org/10.5772/intechopen>
- [36] H. M. Gomes, J. P. Barddal, A. F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Comput. Surv.*, vol. 50, no. 2, 2017, doi: 10.1145/3054925.
- [37] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, no. August, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
- [38] H. T. Zaw, N. Maneerat, and K. Y. Win, "Brain tumor detection based on Naïve Bayes classification," *Proceeding - 5th Int. Conf. Eng. Appl. Sci. Technol. ICEAST 2019*, pp. 1–4, 2019, doi: 10.1109/ICEAST.2019.8802562.
- [39] N. Salmi and Z. Rustam, "Naïve Bayes Classifier Models for Predicting the Colon Cancer," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 546, no. 5, pp. 0–8, 2019, doi: 10.1088/1757-899X/546/5/052068.
- [40] Z. Wang, R. M. Hope, Z. Wang, Q. Ji, and W. D. Gray, "An EEG workload classifier for multiple subjects," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, no. August, pp. 6534–6537, 2011, doi: 10.1109/IEMBS.2011.6091612.
- [41] F. M. Shah and T. Hossain, "Brain Tumor Detection using Convolutional Neural Network Supervised by Brain Tumor Detection using Convolutional Neural Network Bachelor of Science in Computer Science and Engineering Mohsena Ashraf Supervised by Mr . Faisal Muhammad Shah Department of Co," no. August, 2019, doi: 10.13140/RG.2.2.15562.52163.
- [42] R. V. Woldseth, N. Aage, J. A. Bærentzen, and O. Sigmund, "On the use of artificial neural networks in topology optimisation," *Struct. Multidiscip. Optim.*, vol. 65, no. 10, 2022, doi: 10.1007/s00158-022-03347-1.
- [43] S. Grampurohit, V. Shalavadi, V. R. Dhotargavi, M. Kudari, and S. Jolad, "Brain Tumor Detection Using Deep Learning Models," *Proc. - 2020 IEEE*

*India Counc. Int. Subsections Conf. INDISCON 2020*, pp. 129–134, 2020, doi: 10.1109/INDISCON50162.2020.00037.

- [44] S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, “Steel surface defect diagnostics using deep convolutional neural network and class activation map,” *Appl. Sci.*, vol. 9, no. 24, 2019, doi: 10.3390/app9245449.
- [45] A. Pashaei, H. Sajedi, and N. Jazayeri, “Brain tumor classification via convolutional neural network and extreme learning machines,” *2018 8th Int. Conf. Comput. Knowl. Eng. ICCKE 2018*, no. December, pp. 314–319, 2018, doi: 10.1109/ICCKE.2018.8566571.
- [46] D. Podareanu, V. Codreanu, S. Aigner, and C. V. L. Editor, “Best Practice Guide - Deep Learning,” *ResearchGate*, no. February, pp. 1–51, 2019, doi: 10.13140/RG.2.2.31564.05769.
- [47] H. Yingge, I. Ali, and K. Y. Lee, “Deep neural networks on chip - A survey,” *Proc. - 2020 IEEE Int. Conf. Big Data Smart Comput. BigComp 2020*, no. February, pp. 589–592, 2020, doi: 10.1109/BigComp48618.2020.00016.
- [48] N. Ahmad and K. Dimililer, “Brain Tumor Detection Using Convolutional Neural Network,” *ISMSIT 2022 - 6th Int. Symp. Multidiscip. Stud. Innov. Technol. Proc.*, no. June 2019, pp. 1032–1037, 2022, doi: 10.1109/ISMSIT56059.2022.9932741.
- [49] E. Jeczminek and P. A. Kowalski, “Flattening layer pruning in convolutional neural networks,” *Symmetry (Basel)*, vol. 13, no. 7, pp. 1–13, 2021, doi: 10.3390/sym13071147.
- [50] P. Ahamed, S. Kundu, T. Khan, V. Bhateja, R. Sarkar, and A. F. Mollah, “Handwritten Arabic numerals recognition using convolutional neural network,” *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5445–5457, 2020, doi: 10.1007/s12652-020-01901-7.
- [51] C. Pelletier, G. I. Webb, and S. Fellow, “Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series,” no. March, 2019, doi: 10.3390/rs11050523.
- [52] H. Chen *et al.*, “Pre-trained image processing transformer,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 12294–12305, 2021, doi: 10.1109/CVPR46437.2021.01212.
- [53] O. Sevli, “Performance Comparison of Different Pre-Trained Deep Learning Models in Classifying Brain MRI Images,” *Acta Infologica*, vol. 5, no. 1, pp. 141–154, 2021, doi: 10.26650/acin.880918.
- [54] A. M. Ibrahim, M. Elbasheir, S. Badawi, A. Mohammed, and A. F. M.



- Alalmin, "Skin Cancer Classification Using Transfer Learning by VGG16 Architecture (Case Study on Kaggle Dataset)," *J. Intell. Learn. Syst. Appl.*, vol. 15, no. 03, pp. 67–75, 2023, doi: 10.4236/jilsa.2023.153005.
- [55] Y. Kang, N. Cho, J. Yoon, S. Park, and J. Kim, "Transfer learning of a deep learning model for exploring tourists' urban image using geotagged photos," *ISPRS Int. J. Geo-Information*, vol. 10, no. 3, 2021, doi: 10.3390/ijgi10030137.
- [56] N. Ullah *et al.*, "An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning," *Appl. Sci.*, vol. 12, no. 11, 2022, doi: 10.3390/app12115645.
- [57] A. Rajanand and P. Singh, "ErfReLU : Adaptive Activation Function for Deep Neural Network," 2023. doi.org/10.48550/arXiv.2306.01822
- [58] M. K. Abd-Ellah, A. I. Awad, A. A. M. Khalaf, and H. F. A. Hamed, "Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks," *Eurasip J. Image Video Process.*, vol. 2018, no. 1, 2018, doi: 10.1186/s13640-018-0332-4.
- [59] D. R. S. Saputro, I. M. Putri, Sutanto, N. H. Noor, and P. Widyaningsih, "Generalized space time autoregressive (gstar)-artificial neural network (ann) model with multilayer feedforward networks architecture," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 243, no. 1, 2019, doi: 10.1088/1755-1315/243/1/012039.
- [60] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.
- [61] V. Papageorgiou, "Brain tumor detection based on features extracted and classified using a low-complexity neural network," *Trait. du Signal*, vol. 38, no. 3, pp. 547–554, 2021, doi: 10.18280/ts.380302.
- [62] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, and J. Gonzalez-Rodriguez, "Deconstructing cross-entropy for probabilistic binary classifiers," *Entropy*, vol. 20, no. 3, 2018, doi: 10.3390/e20030208.
- [63] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *Schedae Informaticae*, vol. 25, pp. 49–59, 2016, doi: 10.4467/20838476SI.16.004.6185.
- [64] M. Yaqub *et al.*, "State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images," *Brain Sci.*, vol. 10, no. 7, pp. 1–19, 2020, doi: 10.3390/brainsci10070427.

- [65] T. Shelatkar, D. Urvashi, M. Shorfuzzaman, A. Alsufyani, and K. Lakshmana, “Diagnosis of Brain Tumor Using Light Weight Deep Learning Model with Fine-Tuning Approach,” *Comput. Math. Methods Med.*, vol. 2022, 2022, doi: 10.1155/2022/2858845.
- [66] Ž. Vujović, “Classification Model Evaluation Metrics,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.
- [67] T. R. Shultz and S. E. Fahlman, *Encyclopedia of Machine Learning and Data Mining*. 2017. doi: 10.1007/978-1-4899-7687-1.
- [68] M. A. Mohammed, Z. A. Oraibi, and M. A. Hussain, “Content-Based Image Retrieval using Hard Voting Ensemble Method of Inception , Xception , and Mobilenet Architectures,” no. July, pp. 145–157, 2023, doi: 10.37917/ijeee.19.2.17.
- [69] Krish Maniar, Shafin Haque, and Kabir Ramzan, “Improving Clinical Efficiency and Reducing Medical Errors through NLP-enabled Diagnosis of Health Conditions from Transcription Reports,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, pp. 435–442, 2022, doi: 10.32628/cseit2283113.
- [70] N. M. Aboelenein, P. Songhao, A. Koubaa, A. Noor, and A. Afifi, “HTTU-Net: Hybrid Two Track U-Net for Automatic Brain Tumor Segmentation,” *IEEE Access*, vol. 8, pp. 101406–101415, 2020, doi: 10.1109/ACCESS.2020.2998601.
- [71] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, “Brain Tumor Classification Using Dense Efficient-Net,” *Axioms*, vol. 11, no. 1, 2022, doi: 10.3390/axioms11010034.
- [72] A. Minz and C. Mahobiya, “MR image classification using adaboost for brain tumor type,” *Proc. - 7th IEEE Int. Adv. Comput. Conf. IACC 2017*, pp. 701–705, 2017, doi: 10.1109/IACC.2017.0146.
- [73] C. Megha and J. Sushma, *Detection of brain tumor using machine learning approach*, vol. 1045. Springer Singapore, 2019. doi: 10.1007/978-981-13-9939-8\_17.
- [74] Arunkumar, “K-Means clustering and neural network for object detecting and identifying abnormality of brain tumor,” *Soft Comput.*, vol. 23, no. 19, pp. 9083–9096, 2019, doi: 10.1007/s00500-018-3618-7.
- [75] A. A. Pravitasari *et al.*, “UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1310–1318, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14753.

- [76] H. J. M. Sameer, Mustafa A., Oguz Bayat, “Brain Tumor Segmentation and Classification approach for MR Images Based on Convolutional Neural Networks,” *Proc. 2020 1st Inf. Technol. to Enhanc. E-Learning other Appl. Conf. IT-ELA 2020*, pp. 138–143, 2020, doi: 10.1109/IT-ELA50150.2020.9253111.
- [77] D. M. Kumar, D. Satyanarayana, and M. N. G. Prasad, “MRI brain tumor detection using optimal possibilistic fuzzy C-means clustering algorithm and adaptive k-nearest neighbor classifier,” *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 2, pp. 2867–2880, 2021, doi: 10.1007/s12652-020-02444-7.
- [78] H. Habib, A. Mehmood, T. Nazir, M. Nawaz, M. Masood, and R. Mahum, “Brain tumor segmentation and classification using machine learning,” *ICAEM 2021 - 2021 Int. Conf. Appl. Eng. Math. Proc.*, pp. 13–18, 2021, doi: 10.1109/ICAEM53552.2021.9547084.
- [79] D. Filatov and G. N. A. H. Yar, “Brain Tumor Diagnosis and Classification via Pre-Trained Convolutional Neural Networks,” pp. 0–5, 2022. doi.org/10.48550/arXiv.2208.00768.
- [80] M. A. Gómez-Guzmán *et al.*, “Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks,” *Electron.*, vol. 12, no. 4, pp. 1–22, 2023, doi: 10.3390/electronics12040955.
- [81] S. Ullah, M. Ahmad, S. Anwar, and M. I. Khattak, “An Intelligent Hybrid Approach for Brain Tumor Detection,” *Pakistan J. Eng. Technol.*, vol. 6, no. 1, pp. 42–50, 2023, doi: 10.51846/vol6iss1pp34-42.

## الخلاصة

يعد النظام الآلي مهمًا لمساعدة الأطباء وأخصائيي الأشعة على اكتشاف أورام المخ وتصنيفها. تنتج الأورام الصلبة داخل الجمجمة عن انقسام الخلايا غير الطبيعي وغير المنضبط. التحدي الرئيسي في اكتشاف أورام المخ هو الاختلاف في موقع الورم وشكله وحجمه وتنوع أورام المخ وتعقيدها. يعد التعلم الآلي والتعلم العميق الحل الأمثل لهذه المشكلة.

يتضمن العمل المقترح المعالجة المسبقة للبيانات وتجزئة الصور، وتم تطبيق تقنيتين للتجزئة (التجزئة على أساس الحافة والتجزئة على أساس المنطقة) لمقارنتهما، حيث كانت النتائج باستخدام المنطقة أفضل من النتائج باستخدام الحافة. تم تطبيق العمليات المورفولوجية بعد عملية التجزئة وتشمل عمليات الإغلاق (التمدد والتآكل).

بعد ذلك، تم تطبيق التعلم الآلي وخوارزميات التعلم العميق لتصنيف أورام الدماغ بالتصوير بالرنين المغناطيسي إلى أربعة أنواع: الغدة النخامية، والورم الدبقي، والورم السحائي، وعدم وجود ورم في الحالات العادية، بناءً على مجموعة محددة من الميزات التي تعمل على تحسين دقة التصنيف وتوفير الوقت والتكلفة.

في هذا العمل، تم تنفيذ نموذجين مقترحين. الأول هو الجمع بين (مجموعة الهندسة المرئية 16 (VGG16) وأربعة مصنفات تقليدية): آلة ناقل الدعم (SVM)، وشجرة القرار (DT)، والغابة العشوائية (RF)، و (NB) Naive Bayes. يتم تنفيذ هذا المزيج نظرًا لقدراته على التعلم العميق، حيث يمكنه استخلاص ميزات معقدة مثل تفاصيل أورام المخ. النموذج الثاني المقترح هو الشبكة العصبية التلافيفية (CNN).

في هذا العمل، أظهرت النتائج التجريبية أن الجمع ((VGG16) مع الغابة العشوائية) باستخدام التجزئة على أساس المنطقة حصل على دقة قدرها 99.24%. وكانت النسبة أعلى مقارنة مع الجمع ((VGG16) مع الغابة العشوائية) عند استخدام التجزئة على أساس الحافة حيث كانت النتيجة 98.78%.

مجموعة البيانات هذه هي في الأصل مزيج من ثلاث مجموعات بيانات: Figshare، ومجموعة بيانات SARTAJ، ومجموعة بيانات Br35H، التي تحتوي على صور التصوير بالرنين المغناطيسي للأنواع الأربعة من أورام الدماغ التي تم استخدامها.

وأخيراً، تمت مقارنة نتائجنا مع الأبحاث الموجودة في مجال التجزئة والتصنيف على نفس مجموعة البيانات، حيث أثبتت نتائجنا أنها الأفضل. حقق نموذجنا المقترح دقة بلغت 99.24%، في حين تراوحت نتائج الأبحاث السابقة من 95.73%، 97.12%، إلى 87.67%.



جامعة كربلاء  
كلية علوم الحاسوب وتكنولوجيا المعلومات  
قسم علوم الحاسوب

## الكشف والتصنيف الآلي لورم الدماغ البشري بأستخدام التعلم الآلي

رسالة ماجستير  
مقدمة الى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة كربلاء وهي جزء من متطلبات  
نيل درجة الماجستير في علوم الحاسوب

كتبت بواسطة  
سارة علي عبد الحسين قمر

بإشراف  
أ.م. د . الهام محمد ثابت عبد الامير