Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Babylon

College of Information Technology

Software Department

Brain Tumor Detection and Classification Using Machine Learning

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Software

By

Teba Mustafa Kadhum Jassim

Supervised by

Asst. Prof.Dr. Nashwan Jassim Hussain

2023 A.C. **1445 A.H.**

بسم الله الرحمن الرحيم

﴿ لَئِن شَكَرْتُمْ لَأَزِيدَنَّكُمْ ﴾

صدق الله العظيم

سورة ابراهيم ألاية 7

Declaration

I as a result of this declare that this dissertation entitled "Brain Tumor Detection and classification using Machine learning", submitted to the University of Babylon in partial fulfilment of requirements for the degree of Master in Information Technology \ Software, has not been proposed as an exercise for a similar degree atany other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Signature:Name:

Date: / / 2023

Supervisor Certification

I certify that this thesis is prepared under my supervision at the Department of Software / Collage of Information Technology / Babylon University, by *Teba Mustafa Kadhum* as a partial fulfillment of the requirements for thedegree of Master in Information Technology

Signature:

Supervisor Name: Dr. Nashwan Jassim Hussain

Date: / / 2023

The Head of the Department Certification

In view of the available recommendations, I forward this thesis for debateby the examination committee.

Signature:

Asst. Prof. Ahmed Saleem Abbass

Head of Software DepartmentDate: / / 2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Brain Tumor Detection and classification using Machine learning**) presented by the student (Teba **Mustafa Kadhum**) and examined him/herin its content and what is related to it, and that, in our opinion, it is adequate with (Viva Result) standing as a thesis for the degree of Master inInformation Technology-Software.

Signature:
Name:
Title:
Date: / / 2023
(Member)

Signature:
Name: Title:
Date: / /2023
(Chairman)

Signature:
Name:
Title:
Date: / / 2023
(**Member and Supervisor**)

Signature:
Name:
Title:
Date: / / 2023
(Member)

Approved by the Dean of the College of Information Technology,University of Babylon.

Signature:
Name:
Title:
Professor
Date: / / 2023
(Dean of College of Information Technology)

Acknowledgements

In the name of ALLAH, the Most Gracious, the Most Merciful, all praise and gratitude are due to ALLAH, and blessings and peace be upon His Prophet Mohammed (may Allah's blessings be upon him) and his family. I am deeply grateful to ALLAH Almighty for granting me success in accomplishing this task.

I extend my sincere appreciation and respect to my supervisor, Dr. Nashwan Jassim Hussain, for his invaluable guidance and supervision throughout this endeavor.

My heartfelt thanks and utmost respect go to the staff of the College of Information Technology for their exceptional cooperation and assistance.

Lastly, I wish to extend my gratitude to all the kind-hearted, helpful, and wonderful individuals who directly or indirectly supported me in completing this work. Although I am unable to mention them by name here, their support will forever hold a special place in my heart.

Dedication

To whom I have the honor of bearing his name,

To the soul that taught me the meaning of loss,

Since the pain is not in the first days of loss, the happy days come

To someone who puts in a lot of effort

And he left before seeing the fruit of his planting...

My father, may God have mercy on him

To the light of my eyes, the light of my path, and the joy of my life

My mother, then my mother, then my mother...

To the support, humerus, and forearm, my sister and brothers, are the pillars of my strength

I give you the gift of love, honor and dignity

To the one who holds my heart and soul,

whose smiles bring warmth and light,

Reflection of the past and dreams of the future,

My dear daughters, my eternal love.

Teba

2023

Abstract

Brain tumors must be accurately identified and categorized for early detection and effective treatment. Magnetic resonance imaging (MRI) image interpretation by hand is laborious and error-prone. A novel feature extraction method combined by convolutional neural networks and long short-term memory (LSTM) networks are proposed in this thesis to address these issues.

This thesis uses machine learning methods to classify brain tumors in MRI scans. Tomura, which describes texture and structure based on coarseness, contrast, directionality, and roughness, is used to extract relevant attributes from MRI scans. The characteristics are combined with fast Fourier transform and K-means clustering to create a 54-dimensional feature set. These features capture characteristic brain tumor patterns for analysis and classification.

First, machine learning models such as Random Forest (RF) and K-Nearest Neighbors (KNN) for binary classification (tumor vs. no tumor) as well as abnormality classification (different types of tumors) are designed. The findings revealed that both Random Forest and KNN models exhibited high accuracy 97% and 93% in distinguishing between normal and abnormal brain tumor images. Furthermore, these models performed effectively in identifying specific categories of abnormalities with accuracy rates of 80% and 72%, respectively.

Next, the thesis investigates the efficacy of Convolutional Neural Networks (CNN) in classifying brain tumors. The CNN model achieved exceptional outcomes not only in binary classification but also within the domain of abnormality classification tasks. To address the sequential nature of the data, a model is enhanced by incorporating a LSTM network. This LSTM architecture utilizes memory cells and gates to capture long-term dependencies and temporal patterns within brain tumor data. Additional fully connected layers and dropout layers follow the LSTM layers to improve prediction accuracy while mitigating overfitting.

To summarize, this thesis offers a thorough methodology for categorizing

brain tumors in MRI images. The approach encompasses the utilization of feature extraction methods, CNNs, and LSTM networks. By implementing these advanced deep learning techniques, the proposed model achieves 100% accuracy in tumor identification and classification.

Table of Contents

List of Tables

List of Figure

List of Algorithms

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
ANN	Artificial Neural Networks
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CT	Computed Tomography
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
FC	Fully Connected
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbors
LBG	Linde-Buzo-Gray
LSTM	Long Short-Term Memory
ML	Machine Learning
MRI	Magnetic Resonance Imaging
NN	Neural Network
NSCLC	Non-Small Cell Lung Cancer
PCA	Principal Component Analysis
PET	Positron Emission Tomography
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SVM	Support Vector Machines

TCIA	The Cancer Imaging Archive
TN	True Negative
TP	True Positive
VQ	Vector Quantization

Chapter One

Introduction

Chapter One

Introduction

1.1. Introduction

Brain tumors are connected with a high mortality rate among cancer patients. The aforementioned phenomena have a considerable influence due to its close closeness to the human body's principal neural motor, where even tiny aberrations can have significant implications. As a result, developing tools for early identification or notification of the probable presence of a brain tumor is critical. The importance of early detection stems from its ability to significantly increase the likelihood of illness remission and patient survival [1]. There have been substantial advancements in cancer treatment in recent years, notably in the care of early-stage malignancies. The likelihood of survival is significantly greater for patients who receive prompt medical intervention in the initial stages of illness, as opposed to individuals who are not afforded this opportunity[2].

A brain tumor is a cellular growth or aggregation that occurs within the brain. The aforementioned cells are classified as anomalous cells since they differ from healthy brain cells. Within the cranial vault, which houses the encephalon, cells proliferate and enlarge. The increase in cellular mass within the stiff skull shell puts pressure on the cerebral cells, causing severe discomfort and complications [3]. In general, brain tumors or tumors in general can be categorized into two categories. The first form of tumor is categorized as benign or non-cancerous, whereas the latter is labeled as dangerous and cancerous, and is generally referred to as a malignant tumor. The spread of these two types of neoplastic growths within the cranial cavity puts pressure on the brain and poses a serious threat to the patient's health. Tumors are classified into two types based on their origin. The entities indicated above are primary and secondary neoplasms. The original tumor is usually benign and arises within the brain. A secondary tumor, also known as a metastatic tumor, develops from a primary malignancy in another organ of the body and spreads to the brain via the bloodstream or lymphatic system [4]. The earliest detection of any ailment is critical in the treatment of patients, increasing their chances of survival. This behavior also occurs in the context of brain tumors. Early diagnosis of a medical issue has been found to reduce mortality risk and enhance treatment possibilities by up to 90%. The timely detection of a tumor demands the participation of qualified specialists throughout the patient's evaluation procedure [5]. The undertaking under consideration is deemed financially onerous and poses substantial obstacles in terms of feasibility for a large population. The use of computer-aided detection for brain tumor diagnosis is becoming increasingly important. Computer-Aided Detection (CAD) is a technique that allows the initial phase of tumor identification to be completed using specialized software. The Magnetic Resonance Imaging (MRI) device is used to create images of the brain, and the software is responsible for detecting any abnormal regions or places within the brain, such as tumors. Following that, the computer-aided design (CAD) system will assist the human specialist in creating an initial report on probable malignancies. The use of computer-based detection technologies can have a significant impact on the detection of brain cancer [6].

A substantial amount of study has recently been focused into the investigation of automated tumor detection methods for various types of cancers. Scholars are encouraged to investigate creative ways for improving the efficacy of automated tumor recognition and partitioning in magnetic resonance (MR) images. Numerous researchers [7] have proposed CAD-based brain tumor detection strategies.

Brain tumors can be classified using both deep learning (DL) and machine learning (ML) methods. ML-based systems rely on labor-intensive, human-error-prone operations such as manually extracting features and segmenting data prior to classification. The disadvantage of traditional ML-based algorithms is that they use a human feature extraction method. Features are extracted from training photos prior to categorization. Accurate tumor detection frequently necessitates the assistance of a specialist with substantial knowledge in determining the best feature extraction and segmentation algorithms [8]. As a result, while working with larger datasets, the performance of these systems is unstable. In contrast, DL-based algorithms conduct these stages automatically and have proven to be incredibly beneficial in a variety of applications, including medical picture analysis. Because of its dependable performance and weight-sharing architecture, the convolutional neural network (CNN) is a popular DL model. It is feasible to extract both low-level and high-level characteristics from training data automatically. As a result, researchers and academics are considering employing these techniques [9].

Artificial neural networks have been widely used in image processing and medical imaging. Structures for medical imaging have evolved as a fundamental tool for illness identification and image processing. Artificial neural networks have shown great proficiency in doing activities that are considered complex and generally require the involvement of a biological brain. The effectiveness of artificial neural network architectures has allowed them to play a key role in a variety of medical disciplines. Advances in digital electronics and artificial neural network software have resulted in the introduction of neural networks as an important approach for early detection of malignant tumor masses [10]. The feature extraction method aims to improve the discrimination performance of a model by merging various low- and high-level features into a single feature vector [11]. This method does this by eliminating the need to employ a single model's feature map. To effectively categorize tumors, relevant and discriminative characteristics must be extracted from MRIs, which is where a feature fusion-based technique comes in. We evaluated our model using a widely used dataset of brain tumors and numerous distinct quantitative characteristics to demonstrate the usefulness of the proposed method [11],[13].

The current research is focused on the exploration of brain tumors and their timely identification using image processing approaches and artificial neural networks. This study will concentrate on the examination of various brain tumor images, with an emphasis on their processing, and the classification of malignant brain tumors into three classes: glioma, meningioma, and pituitary tumor.

1.2. Thesis Motivation

The motivation behind the thesis can be summarized as follows:

1. Brain tumors have a significant impact on patients and their families, making accurate detection and classification necessary for effective treatment planning. Classification helps identify the tumor type, guide decision-making processes, and enables healthcare specialists to develop personalized treatment plans based on individual tumor characteristics.
2. Manual interpretation of images is a challenging task that requires expertise and experience to identify small or complex tumor regions within the intricate brain anatomy.
3. Tumor heterogeneity is a characteristic of brain tumors that requires comprehensive analysis of different image features to accurately classify them based on type, grade, and subtype.

1.3. Problem Statement

Brain tumor detection and classification are essential tasks in medical imaging analysis and play a crucial role in early diagnosis, treatment planning, and patient prognosis. However, it is a complex problem due to various factors:

1. The difficulty in classifying tumors arises from the complexity of brain tumors, making it challenging to create a single model that accurately detects and classifies tumor types. Despite the use of biopsy and spinal tap procedures for tumor classification, they are inconvenient and time-consuming.
2. The scarcity of available data poses a challenge in developing deep learning

models for classifying brain cancers using magnetic resonance imaging (MRI) due to the cost of data collection and the significant effort required.

3. The imbalance in the distribution of classes within the dataset can make it difficult to train a reliable model that correctly identifies all tumor categories. Therefore, it is essential to address this issue and ensure that the model does not unfairly bias towards the most common category.

1.4 . Research Questions

1. How can MRI imaging data be classified accurately with the use of machine learning and deep learning algorithms when it comes to brain tumors?

2. In what ways can one assess and contrast the effectiveness of machine learning and deep learning models in categorizing brain tumors through various metrics, including but not limited to accuracy, sensitivity, specificity?

3. How can machine learning and deep learning be applied in the clinical setting to classify brain tumors, including but not limited to forecasting treatment outcomes, evaluating disease progression, and assisting with surgical planning?

1.5. Thesis Aim and Objectives

The primary objectives of this thesis are:

1. Building a classification model for subtypes of brain tumors using magnetic resonance imaging (MRI) images with high accuracy and sensitivity to achieve an accurate diagnosis of different tumor types using advanced deep learning techniques.

2. Developing a lightweight model (Hybrid CNN) using deep learning techniques that can accurately detect brain cancers and their types with minimal time and invested resources. The model aims to be efficient and

practical for use in clinical applications where time and resources are limited.

3. Evaluating the effectiveness and performance of the proposed model in detecting and classifying brain tumors using MRI. This evaluation requires the use of multiple metrics, including accuracy, sensitivity, specificity, precision, and F1 score.
4. Conduct a comparative analysis between the suggested model and some popular machine learning approaches, as well as a few relevant recent studies.

1.6. Related Works

The investigation of malignant neoplasms has garnered the interest of numerous scholars globally. Numerous academic publications are produced annually that examine topics pertaining to brain tumors and various techniques for their timely identification. Several studies utilize image processing techniques such as segmentation in their proposed research endeavors. Artificial intelligence architecture is utilized by some to execute such functions. Various detection methods are being employed in other research types to achieve detection.

The process of manually identifying and categorizing brain tumors within large databases of medical images in typical clinical settings incurs significant costs in terms of both time and labor. Therefore, cutting-edge approaches have been devised for brain tumor segmentation, detection, and classification using machine learning (ML) and deep learning (DL) techniques.

Gómez-Guzmán et al. (2023) [14] conducted an investigation to determine the effectiveness of CNN models for a specific task. They evaluated six pre-existing CNN models and developed a generic model using their own dataset, Msoud, to classify brain tumors accurately. InceptionV3, one of the evaluated models, achieved the highest accuracy, with an average accuracy of 97.12%.

Patil and Kirange (2023) [15] introduced a hybrid deep learning model called (CNN-LSTM) for the classification and prediction of brain tumors using Magnetic Resonance Images (MRI). The research makes use of an MRI brain image dataset and makes use of effective preprocessing methods. Then, the CNN is used to pull useful information out of the pictures. The suggested model performs admirably, as evidenced by its 99.1% classification accuracy, 98.8% precision, 98.9% recall, and 99.0% F1-measure.

Nayak et al. (2022) [16] demonstrated classification of 3260 MRI images into one of four classes using a Convolutional Neural Network (CNN) trained with min-max normalization (Glioma, Meningioma, Pituitary, and No-tumor). The constructed network is based on a tweaked version of EfficientNet. Data showed that the model achieved 99.97% accuracy during training and 98.78% accuracy during testing.

Raza et al. (2022) [17] used a hybrid deep learning model called DeepTumorNet, which is based on a fundamental convolutional neural network (CNN) architecture, for the classification of glioma, meningioma, and pituitary tumors, the three most common types of BTs. The GoogLeNet framework is used as the basis for the CNN model. While developing the hybrid DeepTumorNet method, the last five levels of GoogleNet are removed and fifteen extra layers were put in their stead. To further enhance the model's expressiveness, a leaky ReLU activation function is incorporated into the feature map. The proposed model is tested on a publicly available research dataset, yielding results of 99.67% accuracy, 99.6% precision, 100% recall, and 99.66% F1-score.

Maqsood et al. (2022) [18] suggested a method with five distinct stages. In the first step, we use linear contrast stretching to locate the source image's edges. In the second step, a custom 17-layer deep neural network architecture is developed to perform the segmentation of brain tumors. In the third phase, we employ transfer learning to train a modified version of the MobileNetV2 architecture for use in feature extraction. In the last phase, a multiclass support vector machine (M-SVM) and an entropy-based controlled approach are used to select the best features. In the last stage, M-SVM is utilized to classify brain tumors and identify pictures of meningioma, glioma, and pituitary tumors. Figshare showed a 98.92% accuracy rate, but the BraTS 2018 dataset only achieved 97.47%.

Ullah et al. (2022) [19] introduced a new binary TumorResnet deep learning (DL) model for brain identification. The TumorResNet model computes the most distinctive deep features using 20 convolution layers with a leaky ReLU (LReLU) activation function. The categorization of brain tumors MRI into normal and tumorous is done using three completely integrated classification layers. For BTD, the suggested model had a respectable accuracy of 99.33%.

Ullah et al. (2022) [20] employed transfer learning to conduct a comparative analysis of nine classifiers using a common dataset (SARTAJ). The following are deep convolutional neural network models: InceptionResNetv2, InceptionV3, Xception, ResNet18, ResNet50, Resnet101, ShuffleNet, DenseNet201, and MobileNetV2. The InceptionResNetV2 model demonstrated the highest performance with an accuracy of 98.91%.

Abbood et al. (2022) [21] employing a multi-inputs 1D convolutional neural network (CNN) for COVID-19 texture feature extraction and classification is a unique deep learning technique. Using the Gray-Level Co-occurrence Matrix (GLCM), this method uses chest X-ray images to extract distinguishing characteristics. It might aid medical practitioners in making informed choices regarding patient management throughout the epidemic. The SARS-CoV-2 CTscan, COVID-CT, and DLAI3 Hackathon COVID-19 Chest X-Ray datasets are the three datasets utilized to assess our technique. For three datasets, the suggested system's accuracy is 98%, 89%, and 93%, respectively.

Deepak et al. (2021) [22] suggested a hybrid strategy combining support vector machine (SVM) and convolutional neural network (CNN) characteristics for the classification of medical images. Three different types of brain tumors are included in the MRI images that make up the Figshare open dataset. In comparison to the present leading method, the suggested system outperformed it with a classification accuracy of 95.82% across all categories.

Moitra and Mandal (2020) [23] constructed a one-dimensional Convolutional Neural Network (1D CNN) model for the purpose of automating the staging and grading process of Non-Small Cell Lung Cancer (NSCLC). The study utilized the recently updated NSCLC Radiogenomics Collection sourced from The Cancer Imaging Archive (TCIA). The tumor images that had been segmented were inputted into a hybrid model for feature detection and extraction, specifically the MSER-SURF model. The extracted features were combined with the clinical TNM stage and histopathological grade information and inputted into the 1D CNN model. The performance of the CNN model that was proposed exhibited satisfactory results. The accuracy and receiver operating characteristic area under the curve (ROC-AUC) score exhibited superior performance compared to alternative prominent machine learning techniques. The study exhibited favorable performance in comparison to contemporary studies. The presented model demonstrates that a one-dimensional convolutional neural network (1D CNN) is as effective as a traditional two-dimensional or three-dimensional CNN model in predicting non-small cell lung cancer (NSCLC). The model can be further enhanced by conducting experiments with diverse hyper-parameters. Additional research could be undertaken by exploring the application of semi-supervised or unsupervised learning methodologies. Summary ofthe related works, with more details are listed in Table (1.1).

Table (1.1): A Summary of Related Works.

References and work date	Dataset	Classification Method	Accuracy
Gómez-Guzmán et al. (2023) [14]	Brain Tumor MRI dataset Msoud	InceptionV3	97.12%
		ResNet50	96.97%
		InceptionResNetV2	96.78%
		Xception	95.67%
		MobileNetV2	95.45%
		EfficientNetB0	90.88%
		Generic CNN	81.08%
Patil and Kirange (2023) [15]	MRI brain image dataset	CNN-LSTM	99.1%
Nayak et al. (2022) [16]	Figshare	Dense Efficient-Net	99.97%
Raza et al. (2022) [17]	Figshare	DeepTumorNet	99.67%
Maqsood et al. (2022) [18]	Figshare	modified MobileNetV2 and Multiclass SVM	98.92%
	BraTS 2018		97.47%
Ullah et al. (2022)[19]	BTD-MRI dataset	TumorResNet	99.33%
Ullah et al. (2022) [20]	SARTAJ	InceptionResNetV2	98.91%
Abbood et al. (2022) [21]	SARS-CoV-2 COVID-CT	1D CNN	98.387%
	DLAI3		89.26%
	Hackathon		93.38%
Deepak et al. (2021) [22]	Figshare	CNN and SVM	95.82%
Moitra and Mandal (2020)[23]	NSCLC Radiogenomics Collection	1D CNN	96 %

Results on brain tumor classification using deep learning models presented in this literature review (Table 1.1) shows a high level of accuracy, usually above 90%. Notable achievements in this field include the 97.12% accuracy obtained by Gómez-Guzmán et al. with a model such as InceptionV3 or ResNet50, compared to the 99.1% accuracy obtained by Patil and Kirange Nayak et al. recorded a high AUC score of 99.97% with Dense Efficient-Net (Nayak et al. 2019b) while Raza et al. attained 99.67% using DeepTum Other researches also show high accuracy rates, in the range of 95-99% .

However, as with any thesis, there are caveats to this finding. First, high accuracies in certain papers may suggest overfits to their respective datasets. Deep learning models tend to "overfit" on the training data and do not generalize as well to new, unknown data.

In summary, these results demonstrate the promise of deep learning for brain tumor classification but require further investigation, including larger, more diverse datasets and clinical studies, to evaluate their utility in practice and account for possible issues such as overfitting and dataset biases.

1.7. Thesis Outline

The present thesis comprises five primary chapters, with the current section serving as Chapter 1. The remaining part of this work is organized as follows:

- **Chapter 2,** provides theoretical details pertaining to brain tumors in order to enhance readers' comprehension of the etiology, progression, manifestation, and therapeutic modalities associated with this medical condition..
- **Chapter 3,** provides software and running environment utilized in this project are presented.
- **Chapter 4,** presents the findings of the conducted experiments. Furthermore, these chapters presented the outcomes derived from various outputs generated by the project.
- **Chapter 5,** presents the conclusions of the study. This chapter is accountable for concluding the project and presenting the findings obtained from the experiment.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1. Introduction

Artificial intelligence (AI) and its potential applications in medicine are now hot topics. Because it allows robots to learn new skills without being explicitly taught, machine learning (ML) is the core focus of AI research. Deep learning (DL) is a new science that has emerged to implement ML primarily through the use of neural networks for learning and prediction [24]. This chapter discusses the many ML methods that can be used to classify brain tumors.

2.2. Brain Tumor

When brain cells grow abnormally or cancer cells spread to the brain. Because it controls so many biological systems, the brain is vital. Thus, a brain tumor may cause life-threatening symptoms. Primary brain cancers are the original tumors, while secondary ones have spread (referred to as brain metastases)[25]. Brain tumors account for 1–3% of all cancers, occurring 1.9–5.4 per 100,000 people annually. Due to public ignorance of brain tumors, many patients' diagnoses and treatments have been delayed, and some have died. Despite therapy advances that have extended lives, chronic disease cost the US $206 billion in 2006. One-third of that total, $78 billion, goes to real medical care. The rest comes from intangible costs like illness and death-related output loss.[26], [27].

2.2.1. Causes

The cause of brain tumors, like the cause of most tumors, is unknown. Except for vinyl chloride and ionizing radiation, there are no known environmental risk factors for brain cancer [28]. Brain cancers have been connected to viruses, toxins, radiation, genetics, and even embryonic remnants, although each proposed explanation can only account for a minority of tumors [29]. According to some researchers, smoking may cause cancer, albeit the underlying pathogenic pathways are not well known. There is no proven link between these factors and human brain cancers. Multidisciplinary research is still needed to properly explain the etiology of brain tumors [30].

2.2.2. Classification

Brain tumor treatment necessitates an accurate diagnosis of the condition. When a brain tumor is correctly detected, a specific treatment strategy can be provided for the best possible outcome. The WHO classifies brain tumors into numerous kinds based on their cellular origin and histologic appearance. Gliomas are primary brain tumors that grow from glial cells such as astrocytes, oligodendrocytes, or ependymal cells. Brain tumors are characterized based on several variables, including their location, cell type, and grade or aggressiveness [31], [32]. Here are some examples of frequent forms of brain tumors:

1- Gliomas: These tumors originate from the glial cells in the brain and can be further classified into astrocytomas, oligodendrogliomas, and ependymomas.
2- Meningiomas: These tumors develop from the meninges, the membranes that surround the brain and spinal cord.
3- Pituitary adenomas: These tumors develop in the pituitary gland, which is located at the base of the brain and controls various hormonal functions.

The categorization of a brain tumor is critical in selecting the best treatment strategy and predicting the prognosis. Brain tumors can be diagnosed and classified using a biopsy and imaging tests such as an MRI or CT scan. Meningiomas are primary tumors that develop from meningothelial cells and account for around 20% of all primary malignancies [33].

2.3. Medical Imaging

Brain tumors must be diagnosed quickly for effective treatment. Brain biopsies and imaging are used to diagnose these growths. A brain biopsy involves drilling a hole in the skull and removing tumor tissue for microscopic examination. This procedure can reveal the tumor's nature, composition, and etiology, but it's risky[34]

However, medical imaging can detect brain abnormalities without surgery. Clinicians can detect abnormalities using non-invasive imaging methods like MRI, CT, and PET. Physicians can better understand brain abnormalities and their locations by examining images from different modalities [35], [36]. By incorporating biopsy and medical images into patient care regimens, healthcare providers may assure the most accurate diagnosis and treatment options for people with brain tumors.

Magnetic resonance imaging is used to diagnose brain tumors because it is non-invasive and provides good soft tissue contrast. MRI is the standard imaging tool for brain tumor diagnosis because it produces crisp, detailed images of the brain's soft tissues. MRI also provides brain vascular supply and cellular structure information, which is essential for tumor monitoring, diagnosis, and treatment. CT is better for imaging bones, but MRI is better for brain imaging. Combining molecular and anatomical imaging improves diagnostic and surveillance scans for some brain cancers [10], [37].

2.4. Preprocessing

The process of brain tumor classification utilizing deep learning encompasses multiple stages, each of which plays a significant role in the overall procedure. Initially, the process of converting MR images from RGB to greyscale is utilized in order to simplify the input data for subsequent analysis. Subsequently, the process of feature extraction is executed by employing the Fast Fourier Transform (FFT) in conjunction with vector quantization, K-means clustering, and Tamura methods. This enables the identification and capture of significant tumor-related attributes from the provided data. The aforementioned stages collectively contribute to the enhancement of the deep learning model's capacity to accurately classify brain tumors based on MRI images.

2.4.1. Convert Images to Gray Scale Images

Converting RGB images to grayscale images is an important step in feature extraction since it simplifies further analysis by reducing each pixel from three channels (red, green, and blue) to only one channel representing brightness. This not only saves processing complexity but also maintains essential image features for further analysis[38]. A weighted total is generated for each pixel's red, green, and blue values using factors that account for human visual perception variances between these hues to achieve this conversion technically. This yields a single intensity number ranging from 0-255 that describes how bright or dark that specific pixel appears without any color information.

Overall, such modifications pave the door for extracting significant image elements such as edges or textures while also making it easier to store or transfer data over digital platforms with less bits required per pixel when compared to colored [39], [40]:

$$Grayscale\ Image = (0.2989 * R) + (0.5870 * G) + (0.1140 * B) \quad (2.1)$$

Where the red, green, and blue values are indicated by R, G, and B[41].

Algorithm (2-1): Convert RGB images to gray scale[42].
Input: RGB Image **Output: Gray scale image** **Variable: RGB_Image**
Begin
Read RGB_Image
For each pixel in the RGB_image: • ***Calculate the grayscale intensity value using the formula (2.1)*** • ***Set the corresponding pixel in the grayscale image to the calculated grayscale intensity value.***
End for
Return the grayscale image.
End

2.4.2. Gaussian Blurring

A common image processing technique known as Gaussian smoothing or

Gaussian filtering is used to reduce noise and smooth images in order to remove fine detail texture. In this method, a Gaussian kernel, a 2D matrix that describes the weights assigned to each pixel during the blurring phase, is convolved with the image.

The steps to apply Gaussian distribution[43], [44]:

1- Select the size of the Gaussian kernel and the Gaussian distribution's standard deviation (sigma). The level of blurring is determined by the sigma value; higher values provide stronger blurring effects. The neighborhood that is taken into account for blurring is determined by the kernel size, which is commonly an odd-sized square matrix (such as 3x3, 5x5).

2- Using the selected sigma and kernel size, create a 2D Gaussian kernel matrix. The kernel matrix values are computed using the Gaussian function formula, which gives pixels nearer the kernel center larger weights.

The formula is as follows[45]:

$$G(x, y) = (1 / (2 * \pi * \sigma^2)) * exp(-((x^2 + y^2) / (2 * \sigma^2))) \qquad (2.2)$$

Where:

G (x, y) represents the value at position (x, y) within the kernel matrix.

σ (sigma) is the standard deviation of the Gaussian distribution.

The weights in the Gaussian kernel are made to get smaller the further out one goes from the center. This characteristic makes sure that the blurring effect is apparent in the kernel's center and gradually lessens near the edges.

3- Apply convolution: Use a convolution technique to combine the image with the Gaussian kernel. at order to do this, the kernel must be moved across the image while the weighted sum of pixel intensities at each point of the kernel is calculated. The outcome is a new image with a fuzzy appearance since each pixel is now a weighted average of its nearby pixels.

4- After blurring, it could be necessary to normalize or modify the image's pixel values to make sure they fit within the appropriate intensity range.

2.4.3. Histogram Equalization

The contrast of the images can then be altered using a process known as histogram equalization. Histogram equalization reshuffles the image's intensity values to give a more consistent distribution across the image. As a result, visual details that were before buried by low contrast become more visible. Histogram equalization is performed by successfully dispersing out the majority of the frequently occurring intensity values[46] the processes for Histogram equalization are shown here:

1- Compute the Cumulative Distribution Function (CDF):

$$cdf(v) = \sum[H(v) / N] \quad (2.3)$$

Where, H(v): is the histogram value for the pixel value v. The histogram is a representation of how many times each pixel value occurs in the image.

N: is the total number of pixels in the image.

2- Apply Histogram Equalization [47]:

$$T[pixel] = round(((cdf(v) - cdf(v)\_min) / (I * J - cdf(v)\_min)) * (L - 1)) \quad (2.4)$$

In this equation:

- T[pixel]: represents the output intensity value of a pixel in the equalized image.
- cdf(v): is the cumulative distribution function value corresponding to the input intensity value v.
- cdf(v)_min: is the minimum value of the cumulative distribution function among all intensity values.
- I and J: represent the dimensions (width and height) of the image.
- L: is the maximum intensity level of the image.

2.4.4. Image Re-size

Brain tumor enlargement medical imaging and analysis rely heavily on MRI images. It entails scaling all of the images to comply to standard dimensions while maintaining the aspect ratio. This method is used to reduce computational complexity, alter images for rendering or viewing, and reduce storage requirements [48], [49]. However, any information or detail loss that may occur during the scaling process must be considered, especially when large scaling factors are used. The method used for resizing photos, is

Area-based interpolation[50]:

$$new\_pixel\_value = (1/A) * \Sigma(s(i,j)) \tag{2.5}$$

Where:

s(i,j): represent source_pixel_value(i, j)

A: represents the area of the destination pixel.

Σ(source_pixel_value): represents the sum of the pixel values in the source image that fall within the corresponding pixel area in the destination image.

2.5. Feature Extraction

Raw data must be modified for analysis and modeling purposes using a process known as feature extraction. This is especially important when working with high-dimensional data or unstructured data types such as text, images, or audio. The purpose of feature extraction is to extract the most useful information from a dataset and display it in a more digestible and actionable fashion[51], [52].

Here are a few commonly used feature extraction techniques [53], [54]:

a. Statistical Features: Mean, median, standard deviation, skewness, and kurtosis are some of the basic statistical metrics that can be calculated from raw data to summarize its distributional qualities.
b. Domain-Specific Features: These features are designed based on expert knowledge or specific domain understanding. For example, in natural language processing, features like word frequency, part-of-speech tags, or sentiment scores can be extracted from text data.
c. Transformations: Techniques like principal component analysis (PCA), discrete wavelet transform (DWT), and the Fourier transform can be applied to data in order to better visualize patterns within the data or to reduce the data's dimensionality.
d. Feature Encoding: Categorical variables can be used in machine learning models after being converted into numerical representations through encoding techniques such as one-hot encoding or label encoding.

2.5.1. Fast Fourier Transform (FFT)

The FFT algorithm is used to convert an MRI image from the spatial domain (or time domain)to the frequency domain. FFT is a technique used in machine learning to extract valuable frequency-domain information from an MRI image, such as power spectra, texture descriptors, and feature extraction. Power spectra represent the distribution of energy in an image across different frequencies, while texture descriptors capture information about the textural properties of tissues, and feature extraction for machine learning can be used to learn discriminative patterns for tasks such as image classification, segmentation, or disease diagnosis[55]. It is possible to apply the FFT approach on a two-dimensional image while taking into account both the rows and columns. The Fast Fourier Transform (FFT) is a clever algorithm that takes advantage of the structure of the Discrete Fourier Transform (DFT) to significantly reduce the computational effort required to compute the transform. [56]

$$F(u, v) = \sum\sum [f(x, y) * \exp(-2\pi i((ux/M) + (vy/N)))] \quad (2.6)$$

Where:

- F (u, v): is the complex-valued result of the DFT at frequency indices u and v.
- M and N: are the dimensions of the image, representing the number of rows and columns, respectively.
- x and y: have values ranging from 0 to M-1 and 0 to N-1, respectively.

Because of its efficiency and adaptability in signal analysis and processing, the Fast Fourier Transform (FFT) is a vital instrument in many fields of research and engineering. It allows us to learn about a signal's frequency content, extract features, eliminate noise, and execute other operations that rely on frequency-domain representations.

2.5.2. K-mean Clustering Vector Quantization

Vector quantization (VQ) can be used to efficiently capture the distribution of a large dataset of feature vectors while working with a large dataset of feature vectors. This is done to save time and resources during the training phase by not having to maintain and process every single feature vector. The majority of VQ implementations use the k-means clustering technique, which separates a vector space into k clusters iteratively, each of which represents a codeword.

The Linde-Buzo-Gray (LBG) algorithm is a variation of the k-means approach that improves the vector quantization process even further. The LBG approach improves on the initial codeword by iteratively splitting the codewords and adding additional vectors to the relevant clusters. This technique is repeated until a preset target is met, such as codebook size or convergence [57].

An initial set of representative vectors, known as centroids, is chosen. The centroids can be initialized at random. Each data point is then assigned to the centroid with the shortest distance to it, using Euclidean distance as a metric [58].

$$d(x,y) = \sqrt{\sum_{i=1}^{k}(x_j - y_i)^2} \qquad (2.7)$$

Where x_j and y_i represents jth component related to the input vector. In vector quantization, the goal is to reduce the dimensionality of the input data while minimizing the information loss. By replacing each input vector with its nearest centroid, thus can achieve significant data reduction[59].

1- Create a training vector space, T, consisting of M training vectors. Each training vector, Xi, is represented as a set of k-dimensional values: Xi = $\{x_{i1}, x_{i2}, x_{i3}, ..., x_{ik}\}$.
2- Calculate the centroid, C, of the training vector space by taking the average of each column. The centroid, C, serves as the initial codevector: C = $\{C_1, C_2, C_3, ..., C_M\}$.
3- Add a constant error, E, to the centroid to create two new vectors: C1 = C + E and C2 = C - E.
4- Calculate the Euclidean distance between each training vector, Xi, and the codevectors C1 and C2. Euclidean distance can be calculated as in Eq. (2.10).
5- Assign each training vector to the cluster with the minimum Euclidean distance. If the distance to C1 is smaller than the distance to C2, assign the training vector to the first cluster. Otherwise, assign it to the second cluster.
6- Repeat steps 2 to 5 for each cluster until the desired codebook size is obtained.
7- Stop the algorithm when the desired codebook size is reached, and the

clusters are well-defined.

The codebook vectors that result is optimum representation of the original dataset. Each data point can now be represented by its own codebook vector, minimizing the need for storage or transmission.

2.5.3. Tamura Texture Features

Tamura features, often known as Tamura descriptors, are a set of texture features used in image analysis and computer vision. These features give a quantitative description of an image's texture, enabling texture-based analysis and classification[60].

Here are the equations commonly used to calculate the Tamura texture features [61], [62]:

1- Coarseness:

Coarseness measures the average size of the textural elements in an image.

The equation to calculate coarseness is:

$$C = (1 / N)\, \Sigma(G(i,j) / Gmax) \tag{2.8}$$

Where:

- N: is the number of pixels in the image
- G (i, j): represents the local gradient magnitude at pixel (i, j)
- Gmax: is the maximum gradient magnitude across the image.

Coarseness is typically calculated by analyzing the gradient magnitude of the image using techniques such as Sobel or Gabor filters[61].

2- Contrast:

Contrast characterizes the intensity variations within a texture.

The equation to calculate contrast is [62]:

$$Cn = (1/N)\sum[G(i,j) - \mu G]^2 \tag{2.9}$$

where:

- N: is the number of pixels in the image
- G (i, j): represents the local gradient magnitude at pixel (i, j)

- μG: is the mean gradient magnitude across the image.

Contrast is computed by measuring the squared differences between the local gradient magnitudes and the mean gradient magnitude.

3- Directionality:

Directionality captures the preferred orientation or direction of the texture elements in an image.

The equation to calculate directionality is [61]:

$$D = (1 / M) \Sigma[(H(i,j) - \mu H) / \sigma H]2 \quad (2.10)$$

Where:

- M: is the number of pixels in the image
- H (i, j): represents the local gradient orientation at pixel (i, j),
- μH: is the mean gradient orientation across the image and
- σH: is the standard deviation of the gradient orientations.

Directionality is computed by measuring the squared differences between the local gradient orientations and the mean gradient orientation, normalized by the standard deviation.

4- Roughness

Describes the perceived roughness or granularity of an image's texture. It quantifies the variation in pixel intensities within the image's local neighborhoods.

The formula for calculating the roughness feature [61], [62]:

$$\text{Roughness} = \sigma / \mu \quad (2.11)$$

Where:

σ: represents the standard deviation of the pixel intensities within local neighborhoods.

μ: denotes the mean of the pixel intensities within those neighborhoods.

2.6. Machine Learning Techniques

Machine learning (ML) is critical to artificial intelligence because it enables computers to learn on their own. Without being explicitly told, ML approaches can draw inferences and detect patterns in data. This is accomplished by exposing it to a huge number of training sets, each of which aids it in better understanding the system's underlying concept and structure. Essentially, the algorithms have been appropriately trained [32].

One of machine learning's numerous benefits is its capacity to filter through enormous amounts of data in search of useful patterns. Image-based data may be easily processed, supporting specialists in making critical decisions. Furthermore, it can process massive volumes of data in real time, something the human brain cannot [63]. The application of machine learning techniques is not confined to the healthcare industry. Because clinical data analysis has significant risks and costs, ML approaches have been developed for application in the healthcare sector. Machine learning can be more effective than traditional approaches in a variety of situations, including when time and money are limited during development[64].

Supervised learning, unsupervised learning, and semi-supervised learning are the three most important subclasses of ML, as shown in Figure (2.1).

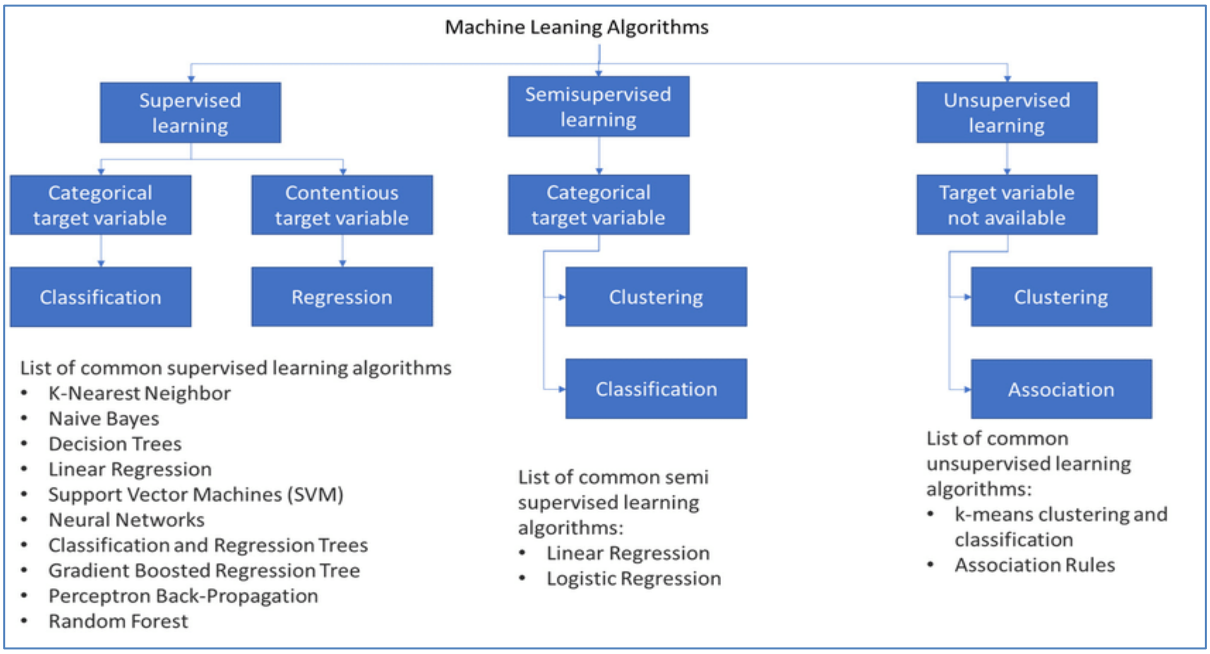


Figure 2.1: The Three Main Categories of Machine Learning Methods[65].

Supervised learning is the foundation of machine learning, in which algorithms are taught to make predictions by being given predetermined inputs and outputs (features and objectives, respectively). In the second form, known as unsupervised learning, algorithms are taught to predict outcomes (targets) based on data presented to them. During the training process, algorithms learn to create predictions by creating connections and patterns in previously unknown data. There are no similarities between the first two types of learning and the third, semi-supervised learning. an algorithm is trained using both labeled and unlabeled data, which can be useful when acquiring labeled data is expensive or time-consuming but unlabeled data are abundant [66], [67]. In this study, supervised learning is used to provide predictions about brain tumors using categorization.

2.6.1. Random Forest

The Random Forest Classifier is a popular machine learning technique for classification tasks. It is a sort of ensemble learning in which multiple models are used to increase prediction accuracy rather than just one. The Random Forest Classifier is built on decision trees. A Random Forest's decision trees are trained using data generated at random throughout the training process. This technique is referred to as "bagging" (bootstrap aggregation). Each decision tree makes predictions on its own, and the final forecast is derived by integrating the results of all the trees[68]. To make use of the training data, the Random Forest approach builds a large number of decision trees. During the training process, each tree in the forest learns to predict on a subset of the original dataset's features. This is accomplished by selecting a subset of features at each split point at random[69]. The Random Forest algorithm outperforms other machine learning approaches in a variety of ways. For example, it can handle big datasets with high dimensionality and non-linear feature interactions. Furthermore, it is resistant to overfitting and can estimate feature relevance, making it suitable for feature selection problems.

Furthermore, the Random Forest approach does not require data normalization or scaling, and it works well with noisy or missing data. Furthermore, the Random Forest algorithm is highly interpretable and can aid in explaining how judgments are formed via its decision trees. In conclusion, the Random Forest algorithm is a powerful ensemble learning technique that mixes several decision trees to produce high-quality predictions [70]. A majority vote of all individual decision tree outcomes is used to reach the forest's consensus verdict (see Fig. (2.2)).

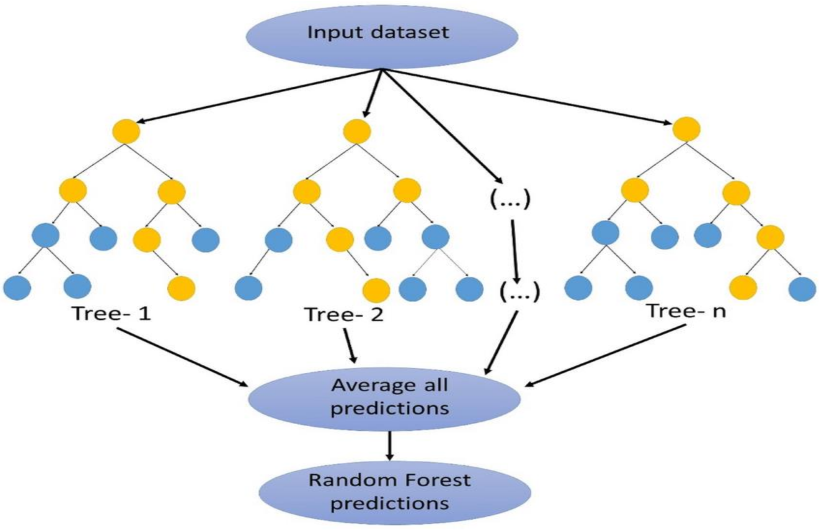


Figure 2.2: Architecture of the Random Forest Classifier [71] .

Algorithm (2-2): Random Forest [72].
Input: Training data, number of trees, number of features to consider at each split
Output: Random Forest model
Varaible: int number_of_trees, votes.count **List decision_trees**
Begin
Step 1: Split dataset randomly into subsets (70% training, 30% testing).
Step 2: decision_trees = []
Step 3: for _ in range(number_of_trees): ▪ ***Randomly select a subset of features.*** ▪ ***Grow a decision tree from the given dataset.*** ▪ ***Add the tree to the list of decision trees.*** ***return decision_trees***
Step 4: Predicts the class label for the given data. ***votes = []*** ***for tree in decision_trees:*** ***vote = tree.predict(data)*** ***votes.append(vote)***
Step 5: predicted_class = max (votes, key=votes.count)

return predicted_class.
End.

2.6.2. K-Nearest Neighbors

K-Nearest Neighbors (KNN) classification is used in supervised machine learning to predict a data point's class by examining the classes of its neighbors in the training data [73].

When just basic data distribution information is given, the KNN is the simplest and most fundamental classification approach. As illustrated in Figure (2.3), the K (number of neighbors) and distance metric options have a significant impact on the performance of a KNN classifier [74], [75]. The distance measure used to compare two data points strongly influences whether they are considered similar or dissimilar. The following is distance metrics in KNN used[76] :

1- Minkowski distance: is a generalization of both Euclidean and Manhattan distances. It allows flexibility in the distance metric by introducing a parameter p that controls the order of the distance calculation.

$$d(A, B) = (|x2 - x1|\^{}p + |y2 - y1|\^{}p)\^{}(1/p) \qquad (2.12)$$

When p = 2, the Minkowski distance becomes the Euclidean distance, and when p = 1, it becomes the Manhattan distance.

The steps for KNN [73] are shown in Algorithm (2.3):

Algorithm (2-3): Algorithm of KNN [76]**.**
Input: Training set S containing feature vectors x1, x2, ..., xn
Output: Class label for the test vector ***Xtest***
Variables: Int k, i, distance, class_count **String label, class_label** **List neighbor**
Begin
Step 1: Set the value of k (e.g., k = 5)
Step 2: Create an empty dictionary class_votes

For each data point (Xi, yi) in S:

- ***Calculate the Minkowski distance dist between Xtest and Xi using equation (2.20)***
- ***Add the distance dist and the corresponding label yi to the dictionary class_votes***

Sort the distances in ascending order

Step 3: Create an empty list neighbor

For each i in range(k):

If i < number of sorted_distances:

Add all labels corresponding to the k smallest distances to the neighbors list

Step 4: Create an empty dictionary class_counts

For each label in neighbors:

If the label is already in class_counts:

Increment its count

Else, add it to class_counts with count 1

Find the maximum count among the class_counts values

Step 5: Create a variable class_label and set it to None

For each label, count in class_counts:

If the count is equal to the maximum count:

If class_label is None:

Set class_label to the label

Else:

If the distance of nearest neighbor < than the kth smallest distance:

Set class_label to the label

Step 6: Return class_label as the predicted class label for Xtest

End.

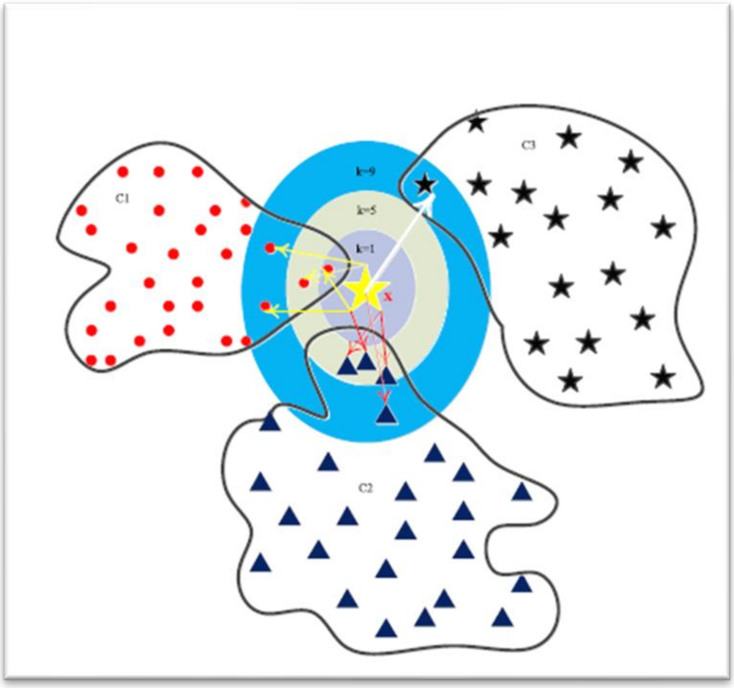


Figure 2.3: An Example of the *K*NN Classification Process [74].

2.7. Deep Learning (DL)

Deep learning is a new field of study that focuses on building theories and algorithms that resemble human brain networks in order to enable machines to learn new information on their own. Deep learning is a subfield of machine learning that was created as an AI technique to replicate human learning processes in a specific area[77]. Deep learning algorithms are layered in a complicated and abstract hierarchy of layers, with each higher layer building on the previous lower layer, thus the name hierarchical learning. Deep learning made its debut in 2007. Because of its ability to process enormous datasets and solve tough problems, Microsoft, Facebook, Amazon, Google, and Baidu all employ it on a regular basis[78].

Deep learning uses computer models made up of numerous processing layers to represent data at increasing degrees of granularity in order to describe complex structures in huge and massive datasets. Deep learning, as a unique form of machine learning, bridges the gap between classical ML and AI. Deep learning has various uses, including object detection, speech recognition, and even medicine[79]. Feature extraction is a critical technique in machine learning. Natural information, on the other hand, may be beyond the capability of ordinary machine learning to process and deal with in its raw form [80]. Deep neural networks beat shallow ML methods in most applications involving the processing of text, image, video, voice, and audio data, making DL particularly advantageous in many businesses with enormous, high-dimensional data.The rapid advancement of ML algorithms and processing is one reason for deep learning's popularity[81].

2.7.1. Convolutional Neural Network (CNN)

The diagnosis and treatment of brain tumors are strongly reliant on precise prognoses. Several attempts have been made to construct deep learning models for this challenge. Convolutional neural networks (CNNs) are one of the most often used types of deep neural networks in machine vision. The Convolutional Neural Network (CNN) is a deep learning technique that tries to replicate how the brain processes information [82]. CNN networks are comparable to multi-layer networks (Perceptron), but they can combine numerous locally connected networks into a single one. Aside from the layers used for feature extraction, there are other layers used for categorization. Increased accuracy in automated diagnostic systems and disease prediction are two areas where CNN shows great promise. Due to their high data-processing capability, CNNs have become increasingly popular in the field of artificial intelligence[82].

A convolutional neural network (CNN)'s output becomes the input for the next layer. The input layer is the first in the network, followed by the output layer. The secret layers are the components of the network that connect the input and output nodes[83].

2.7.1.1. Components of CNN Architecture

It is possible to create a convolutional neural network that acts and thinks like a human brain. Prediction is a major advantage of this network in general. There are two primary elements that make up a CNN[84]:

A) Feature Extractor

CNN's first stage in data processing is feature extraction and feature map construction. CNN is composed of various filters, each of which serves a distinct purpose. This resulted in the creation of a multiplicity of feature maps, each of which represented a unique collection of filters. The low-dimensionality features vector is the output of the features extraction approach and is given into a classifier. The feature extractor is composed of numerous layers (multiple convolution layers with optional pooling layers). In a convolution layer, the input and filter are convolved to yield feature maps, which are then pooled down in a reduction layer. The output feature maps are then fed back into the system as input feature maps, where the process is repeated layer by layer in order to extract more sophisticated features. Finally, the feature maps with reduced dimensions are flattened to produce a low-dimensional feature vector[85].

B) Classifier

Following the extraction of feature maps, the best features from each are chosen to produce a low-dimensional feature vector, which is then input into a classifier. The classifier reports the likelihood of an input belonging to a specific class. To do this, the classifier is composed of one or more completely linked layers [85].

2.7.1.2. CNN Architecture

The input layer of a CNN represents the model's input (the intentionally specified characteristics), but its size is independent of the network's overall number of layers. For processing gene expression data, the input layer of a convolutional neural network (CNN) is commonly a two-dimensional matrix of size (n m), where n is the number of samples and m is the number of features [86]. A neural network is made up of layers, as shown in Figure (2.6). These are the many levels:

1- Convolutional layer.

2- Pooling layer (or Sub Sampling layer).

3- Fully Connected Layer (Classification layer).
4- Dropout Layer.
5- Activation Function.
6- Optimization.

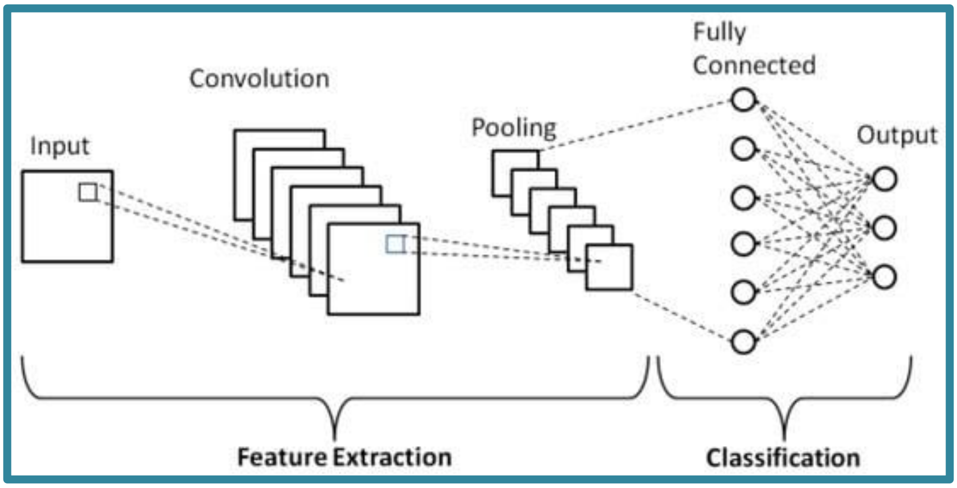


Figure (2.4): A Simplified Representation of a Convolutional Neural Network (CNN)[87].

1- Convolutional layer

Convolutional neural networks (CNNs) are a deep learning architecture that is used for image and video processing. The convolutional layer extracts local patterns and features from the input data using a collection of learnable filters. Each filter includes a distinct set of learnable weights that are assigned at random and adjusted throughout the training phase via gradient descent and backpropagation. The convolution operation is applied between the filter and the receptive field of the input data during the forward pass of the convolutional layer. In the output feature map, the dot product of the filter weights and the associated input values is computed and added to produce a single value[88].

The network adjusts the weights during the training phase to improve its performance on the given job, such as picture categorization or object detection. Backpropagation gradients are used to update the weights, gradually refining the filters to better capture the desired features. The convolutional layer captures a wide range of features and creates rich representations of the input data by combining many filters with distinct weights[89].

The architecture of a Convolutional Neural Network (CNN) contains various parameters that are used to control things like the model's behavior, the amount of its output, and how long it takes to run (hyperparameters). Choosing proper hyperparameter values is critical to ensuring optimal network performance. Here are some critical CNN hyperparameters:

a) **Number of filters**: The number of filters in each convolutional layer and their spatial size (width and height) determine the complexity and expressiveness of the network.

b) **Stride**: The stride determines how big a step the filters take when moving over the input. Larger strides can speed up processing, but at the risk of losing fine-grained data.

c) **Learning rate:** The learning rate controls the gradient descent optimization step size, which affects how quickly the network converges. For suffered and effective training, the right learning rate is essential.

d) **Batch size:** The batch size controls how many training examples are handled all at once before the model's weights are updated. It has an impact on how much memory is used, how effectively computation is done, and how the network behaves when it converges.

e) **Padding:** Padding is a method of surrounding the input with additional pixels in order to retain spatial dimensions and minimize information loss at the boundaries. It is crucial to select a suitable padding scheme (such as "valid" or "same") based on the intended output size and the filters' receptive field.

2- **Pooling layer or Sub Sampling layer**

CNN accomplishes its results by combining convolution and pooling layers. The main purpose of this layer is to provide reduced-dimensional output by lowering the input dimensions while retaining the most important information. This layer reduces dimensionality by using maximum and average pooling [90]. Figure (2.7) depicts the maximum amount of pooling.

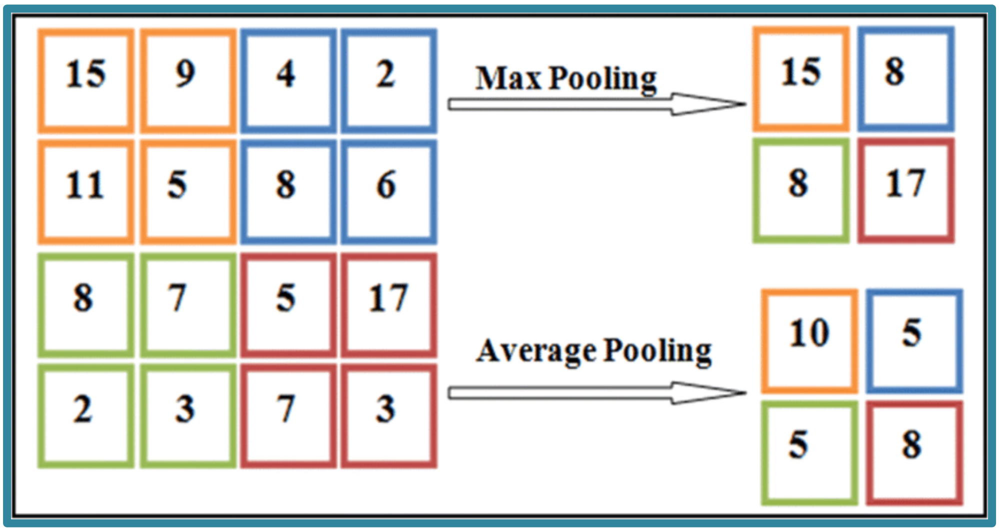


Figure (2.5): Max-Pooling Operation with a 4X4 Block Size [90].

There are two common types of pooling operations used in CNNs:

1- Max pooling: In max pooling, the maximum value within each receptive field is selected as the output value.

2- Average pooling: Average pooling calculates the average value within each receptive field as the output value [91], [92].

3- **Fully Connected Layer (Classification layer) (FC).**

A fully connected layer (FC), also known as a dense layer, is included as the network's last layer. Every neuron in the layer below it is related to every neuron in the layer above it in a completely linked layer. Before the previous layer's resultant feature map can be properly coupled with the output layer, it must first be flattened into the form of a feature vector. The output layer is made up of neurons with the same number of classes as the final CNN layer, which uses softmax or sigmoid activation functions to categorize the learned data. These activation functions are optimized for multi-class and binary class classification, respectively [93]. Figure (2.8) presents a visual representation of the relationship that exists between completed feature maps and a layer that is fully connected.

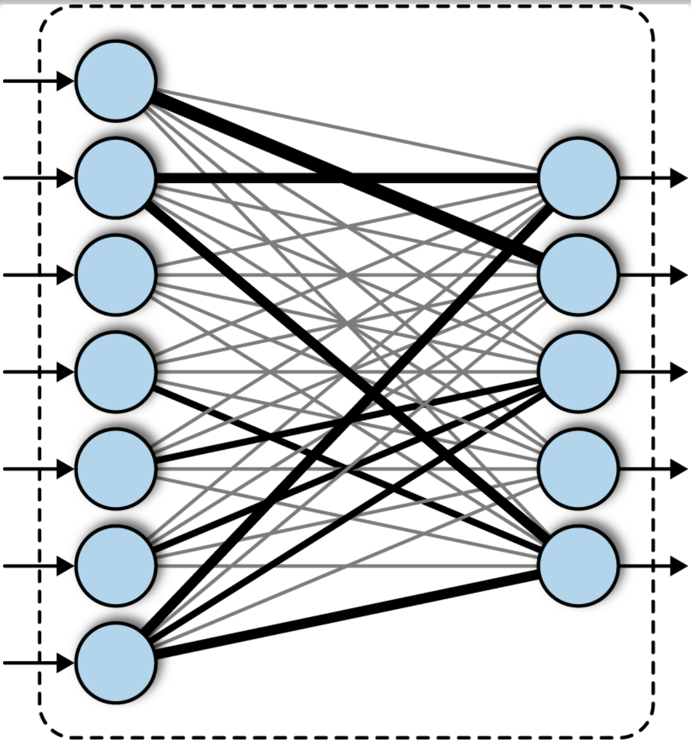


Figure (2.6): Connection between Convolution Layer and Fully Connected Layer [93] .

1. Input Transformation: The input to the FC layer is a flattened representation of the high-level features extracted by the preceding convolutional or pooling layers.
2. Weighted Connections: Neurons in the FC layer are fully connected, with each connection having an associated weight parameter. Output is obtained by applying a weighted sum of inputs followed by an activation function.
3. Non-linear Activation: FC layer introduces non-linear activation functions to enable the network to learn complex decision boundaries.
4. Output Units: The number of neurons in the FC layer corresponds to the number of output classes in the classification task, indicating the probability or confidence of the input belonging to that class.

5- Activation functions

Activation functions are often applied element-by-element to a neuron's weighted sum of inputs (also known as activation). The activation function introduces nonlinear transformations into the data, allowing neural networks to mimic nonlinear functions and capture more complicated patterns.

Here are some examples of neural network activation functions[94].

1- **Leaky ReLU Activation**: The leaky ReLU is a variant of the ReLU activation function that addresses one of its limitations by allowing small negative values for negative inputs. It is given by the equation [96]:

$$f(x) = \max(ax, x) \tag{2.12}$$

Where:

f(x): represents the output value after applying the activation function to the input x.

2- **The softmax function,** also known as the soft argmax or normalized exponential function. It takes as input a vector of arbitrary real values and transforms them into values between 0 and 1, while ensuring that the sum of the transformed values is equal to 1.

Mathematically, the softmax function is defined as follows[97]:

$$f(x\_i) = e\hat{}(x\_i) / (\Sigma\, e\hat{}(x\_j)) \tag{2.13}$$

Where:

f(x): represents the output value after applying the softmax function.

x_i: is the i-th element of the input vector x.

exp(x_i): represents the exponential of x_i.

Because the softmax function assigns greater probability to larger input values, it is appropriate for identifying the most likely class in a multi-class classification problem. It is frequently combined with the cross-entropy loss function when training neural networks for classification problems.

3- Adam Optimization Algorithm

Adam is able to iteratively adjust the network weights without any additional data beyond the training set, which is a significant improvement over stochastic gradient descent. Adam relies on AdaGrad and RMS Prop to solve sparse gradients in very noisy settings. In order to hasten convergence and enhance performance, Adam automatically modifies the learning rate for each parameter. Each parameter's adaptive learning rate is determined by taking the mean of its preceding gradient and squared values. Through these equations, Adam is able to estimate the gradient's momentum and variance over time, allowing him to fine-tune the learning rate for each parameter. The very latest version [97], [98]:

$$\theta = \theta - \alpha * m / (\sqrt{v} + \varepsilon) \tag{2.14}$$

If the first and second moments, m and v, respectively, are, the learning rate is, and the little constant, keeps us from dividing by zero. The Adam optimizer combines momentum-based optimization with adaptive learning rates to improve the training of deep learning models by dynamically adjusting the learning rate of each parameter in response to its past gradients [99].

2.8. Long Short-Term Memory (LSTM)

This section will delve into the nuances of Long Short-Term Memory, a subtype of recurrent neural network that has sparked a lot of interest and produced impressive results in a variety of application areas, including NLP, time series analysis, and sequential data modeling. It is especially effective at capturing long-term dependencies within sequences with varying time intervals [100]. The fundamental goal of building LSTM networks is to avoid the vanishing gradient problem that is typical in standard RNNs. This problem occurs when the network struggles to learn and retain knowledge from distant past time steps due to exponential decay during backpropagation induced by repeated gradient multiplication.

To address this issue, LSTM provides memory cells and gates that control the flow of information during network operations [101].The memory cell, input gate, forget gate, and output gate are the key components of an LSTM unit. As shown in Figure (2.10), the memory cell is a storage component that allows the network to retain information across longer sequences. The input gate controls how much fresh data is stored in the memory cell, while the forget gate controls how much old data is deleted.

Which information is sent on to successive layers or used for final predictions is determined by the output gate. In terms of architecture, LSTM is made up of a series of interconnected units.

Each unit's output becomes the input for the next unit in the sequence. This design enables long-term information retention and updating across numerous time steps, making it suited for jobs involving sequential data [102].The parameters of an LSTM network are tuned using a process known as backpropagation through time to train it. Depending on the job, the network adjusts its parameters to minimize a given loss function, such as mean squared error or category cross-entropy.

In the context of classifying brain tumors, LSTMs can effectively capture temporal dependencies and patterns within input data. This results in accurate predictions and improved performance[103].

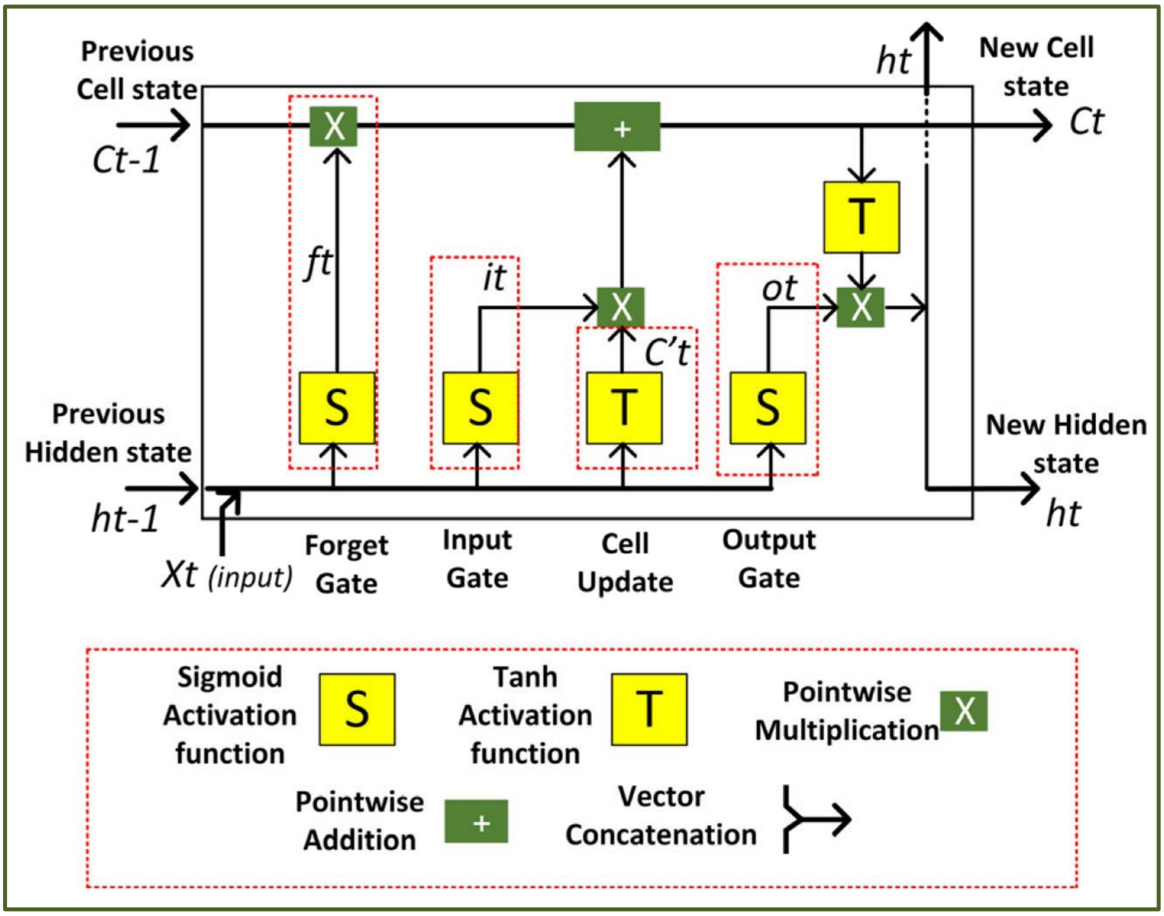


Figure (2.7): The Architecture of LSTM Network[102].

2.9. Performance Evaluation

In order to train a machine learning model to its full potential, evaluation metrics are crucial. As such, choosing appropriate assessment criteria is a crucial step in differentiating and achieving the best model.

2.9.1. Confusion Matrix

There exist several metrics that can be employed to evaluate the efficacy of certain classification algorithms from an academic perspective. These consist of accuracy, f1-score, precision, and recall. To determine these measures, a confusion matrix is computed, this table represents the number of instances correctly or erroneously predicted by a particular model for classification as shown in Figure (2.11). Detailed explanations pertaining to each value presented within this tabular format are provided below:

1. True positive (TP) refers to the correctly classified positive instances.
2. A false negative (FN) refers to instances where positive examples are inaccurately identified as negatives.
3. A false positive (FP) refers to negative instances that are mistakenly forecasted and categorized.
4. A true negative (TN) refers to instances that are correctly classified as negative by the model of classification.

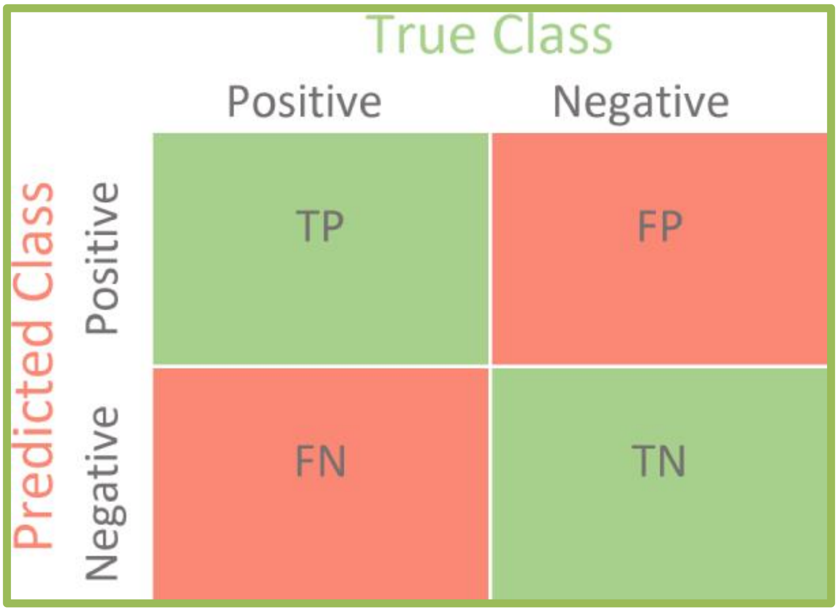


Figure (2.8): Confusion Matrix [106].

2.9.2. Performance Metrics

Evaluating the performance of the models is vital in determining whether or not they are helpful in properly predicting tumor existence and type in the context of brain tumor classification utilizing machine learning and deep learning algorithms. The model's performance and classification skills are measured using a number of different performance criteria. The following are the primary indicators of performance used in this analysis:

- Accuracy, or the degree to which a model is likely to accurately predict outcomes, is defined by the proportion of correct predictions relative to the total number of predictions, as shown in Eq. (2.15) [104]:

$$\text{Accurcy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.15}$$

- Precision refers to how accurately a group of documents describes its subject, and thus how precisely they are classified. Class ci, symbolized by the symbol (Pi), has an accuracy that can be quantified as follows, as shown in Eq. (2.16) [105]:

$$\text{P}_\text{i} = \frac{\text{TP}_\text{i}}{\text{TP}_\text{i} + \text{FP}_\text{i}} \tag{2.16}$$

- Recall measures how well a classifier can identify documents as belonging to a given class (as demonstrated by Eq (2.17). Class ci recall (Ri) can be calculated using the formula [105]:

$$\text{R}_\text{i} = \frac{\text{TP}_\text{i}}{\text{TP}_\text{i} + \text{FN}_\text{i}} \tag{2.17}$$

In this case, $\boldsymbol{TP_i}$ points to a true-positive value. $\boldsymbol{FP_i}$ stands for false positives and $\boldsymbol{FN_i}$ represents false negatives.

- F1 is the precision-recall synchronization rate. Overall system performance is good if F1 is high. Given Eqs. (2.18), the following is a description of F1 [105]:

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{2.18}$$

Chapter Three

The Proposed System

Chapter Three

The Proposed System

3.1. Introduction

Using machine learning and deep learning approaches, this chapter offers a system for detecting and classifying brain cancers from MRI data. The system's goal is to increase the accuracy and efficiency of brain tumor diagnosis, allowing for more rapid and appropriate treatment options. The suggested method consists of several stages, including image preprocessing, feature extraction, and classification utilizing advanced algorithms. Specifically, Fast Fourier Transform (FFT), K-means, and Tamura will be used to extract features, as well as convert MRI images from RGB to grayscale.

3.2. Proposed System

The present part aims to outline the entire workflow of the proposed system, which includes multiple processes as illustrated in Figure (3.1). The full process for categorizing brain tumors may be summed up as follows: First, the MRI image must be preprocessed using standard methods to assure consistency and improve picture quality. These methods include grayscale conversion, Gaussian blurring, histogram equalization, and scaling. Second, extract useful characteristics from the preprocessed images using three separate algorithms: the Fast Fourier Transform (FFT), the K-means clustering, and the Tamura technique. The dimensionality of the data is reduced and significant tumor features are captured by these techniques. Third, generate feature vectors by combining feature extraction results from the three techniques. This procedure yields a condensed representation of the MRI images, which faithfully records the essential details required for categorization. In this way, the proposed method may efficiently and accurately classify MRI images of brain tumors by extracting and combining important information.

Following that is classification, in which machine learning or deep learning algorithms are used to classify data into tumor or non-tumor classes based on the expected outcomes.

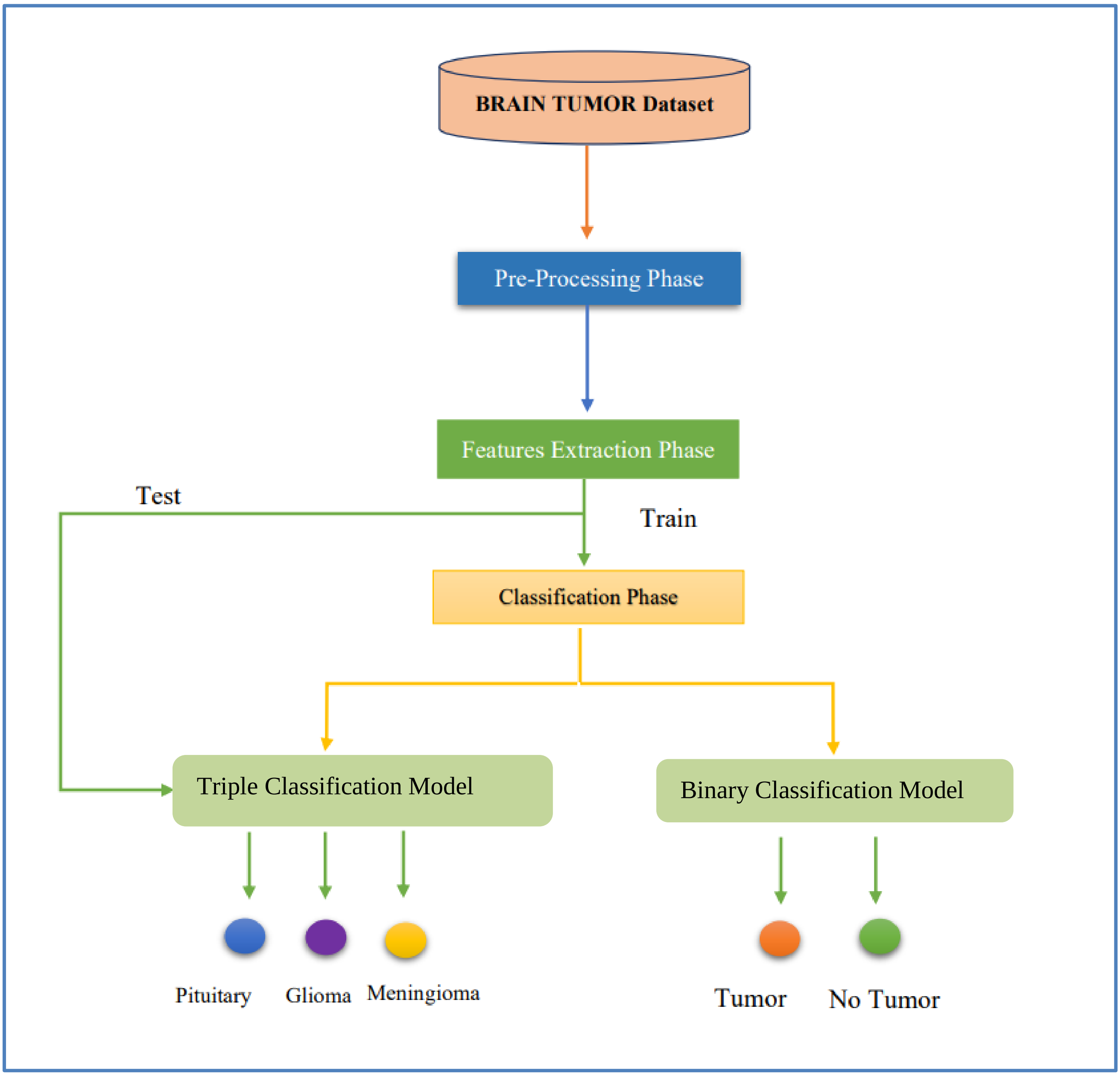


Figure (3.1): The Proposed System.

Algorithm (3-1): The Proposed system.

Input: Brain Tumor MRI Image
Output: Classified Images

Stage 1: Preprocessing

Step 1: Convert the MRI image to grayscale using Algorithm (2-1).
Step 2: Apply Gaussian Blurring using Eq. (2.2) to the grayscale image for the abnormal class detection.
Step 3: Perform Histogram Equalization using Eq. (2.4) on the preprocessed image to enhance contrast.
Step 4: Resize the preprocessed image to a fixed dimension for uniformity using Eq. (2.8).

Stage 2: Feature Extraction

Step 1: Apply Fast Fourier Transform (FFT) using Eq. (2.9) with K-means clustering using steps in section (2.5.2) to the preprocessed image to extract features.
Step 2: Apply Tamura method for feature extraction, including coarseness, contrast, directionality, and roughness using Eq. (2.11),(2.12),(2.13),(2.16).
Step 3: The features extracted from FFT with K-means and Tamura are combined into one file.

Stage 3: Machine Learning

Step 1: Utilize the Random Forest algorithm using algorithm (2-2) to train a classifier using the extracted features from Stage 2. This model will learn to classify MRI images into different classes based on the extracted features.

Step 2: Apply k-nearest neighbors (KNN) algorithm using algorithm (2-3) as another classifier using the same features for image classification.

Stage 4: Hybrid CNN-LSTM

Step 1: Set up a hybrid model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers.

Step 2: Train the hybrid CNN-LSTM model using the original MRI images as input data.

Step 3: Fine-tune the model using the extracted features from Stage 2 to enhance classification performance.

Step 4: The hybrid CNN-LSTM model will predict the class labels of new MRI images based on the learned patterns from the training data.

3.2.1. Dataset Characterization

This dataset contains 7023 pictures of the human brain. The dataset exhibits four types of brain cancers in unequal numbers across the training and testing phases: gliomas (1621), meningiomas (1645), no tumors (2000), and pituitary tumors (1757).

3.2.2. Preprocessing Stage

To preprocess the MRI images before applying machine learning or deep learning algorithms for tumor detection and classification, the system employs a variety of image processing approaches. The suggested system includes three techniques: image conversion to grayscale, histogram equalization, and image scaling for tumor and no tumor datasets as shown in Figure (3.2) and Figure (3.3). Where the tumor type data is subjected to all of the above procedures as well as the gaussian blur preprocessing.

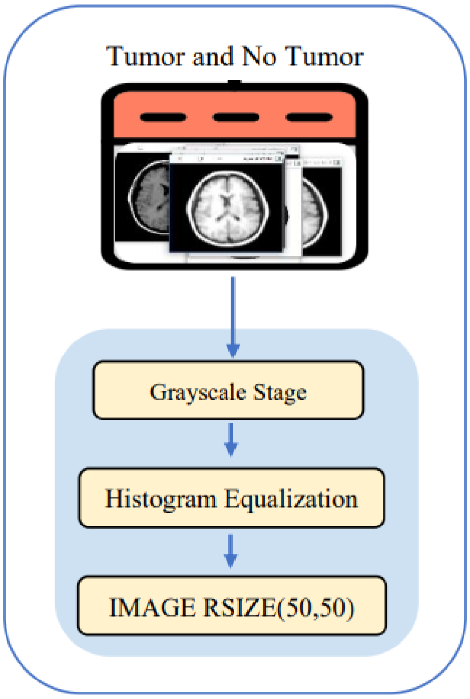


Figure (3.2): Preprocessing Stage for Tumor and No Tumor Dataset.

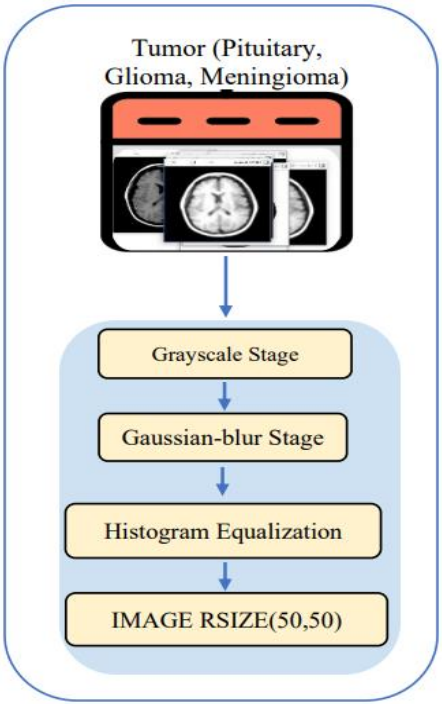


Figure (3.3): Preprocessing Stage for Tumor (Pituitary, Glioma, Meningioma) Dataset.

3.2.2.1. Grayscale MRI image conversion

The first stage of image pre-processing is grayscale conversion of MRI images. Grayscale images include only one channel that displays intensity values, making computations and algorithms easy to apply to the image. Working with grayscale images can also help with image analysis techniques. At this point, the method described in Chapter 2 Algorithm (2-1) can be used to convert each pixel of the input MRI image with 24 bits into grayscale pixels with only 8 bits.

3.2.2.2. Gaussian blur

In this stage, the MRI images including the tumor are processed using a Gaussian blur. Tumor borders in MRI scans may appear uneven or entail anomalies due to radiation exposure or patient movement; gaussian blurring serves to smooth out these pixel-level irregularities, giving a less noisy and more visually consistent image. Gaussian blurring can be done using the steps described in Section (2.4.2) of Chapter 2 with kernel size of 5x5.

3.2.2.3. Histogram Equalization

The histogram equalization approach is used to improve the contrast of images, notably MRI (Magnetic Resonance Imaging) images. Low contrast in MRI scans occurs when there is minimal differentiation in the image between different tissue types or structural features. The approach's purpose is to disperse the intensity values across the entire dynamic range in order to maximize the accessible levels and increase the apparent number of details. This method can be used to highlight tumors in grayscale images. Histogram equalization works by altering the cumulative distribution function (CDF) of the image's histogram, as described in section (2.4.3).

3.2.2.4. Image Resize

Area interpolation according to equation (2.8) in Chapter 2 is used to attain the desired size of 50x50 pixels. These strategies aid in the preservation of crucial information throughout the resizing process, as well as the reduction of distortions or loss of critical details.

3.2.3. Feature Extraction

The following step is to extract distinctive features from the preprocessed MRI scans. The goal of this procedure is to record important features that can be used to distinguish tumors from healthy tissue. The two feature extraction techniques that will be employed are the Fast Fourier Transform with k-mean cluster and Tamura.

3.2.3.1. Fast Fourier Transform (FFT) and k-mean cluster

Following that, FFT is a technique used to convert preprocessed spatial domain MRI image representations into frequency-domain representations, exposing important information about the image's characteristic frequencies. This approach aids in the accurate identification of brain tumors during medical diagnosis procedures.

To simplify the representation of a big dataset, vector quantization (VQ) uses a subset of typical vectors, or centroids, as codewords. The codebooks used in VQ are collections of vertices, and K-means clustering is a common method for designing them. The initial centroid is selected at random. The output features from the Fast Fourier

Transform (FFT) are then mapped to the centroid with the shortest Euclidean distance using equation (2.10) from Chapter 2. Each iteration of the method refines the centroids that have been placed in each cluster. Thus, the final centroids reflect a condensed version of the characteristics found in the FFT output. This method may be used anywhere an efficient representation of data is required since it decreases data size without losing valuable frequency information. Overall, integrating FFT and K-means allows for more efficient feature extraction in MRI images with frequency characteristics essential to detecting anomalies. Combining FFT and K-means for feature extraction has significant advantages in brain tumor detection and classification. FFT provides important information on tumor texture and spatial distribution, while K-means clustering allows for additional analysis and categorization.

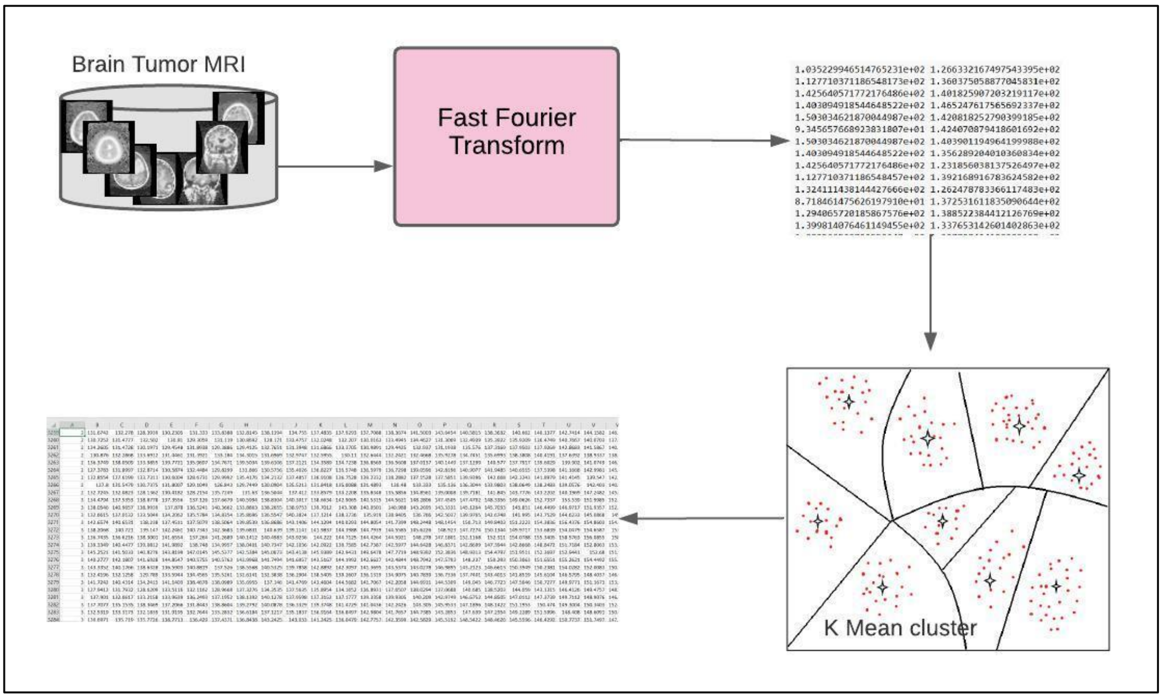


Figure (3.4): FFT and K-Mean Cluster Feature Extraction.

3.2.3.2. Tamura

Tamura is an MRI feature extraction technique that focuses on four primary attributes: coarseness, contrast, directionality, and roughness according to the equations (2.11), (2.12), (2.13), (2.16) in Chapter 2. These qualities are utilized to explain the patterns of brain tumors visible in MRI scans, and Tamura is an excellent way for detecting brain tumor patterns by carefully picking these features from six alternative options.

3.2.3.3. Integration of Feature Extraction

By combining the aforementioned feature extraction approaches, a full feature representation for use in identifying and categorizing brain tumors can be created. As illustrated in Figure (3.3), Three different methods are used in this feature extraction procedure to successfully glean useful characteristics from the provided data. The first and second methods both work together to extract useful frequency and compact representations through Fast Fourier Transform (FFT) and K-means clustering, respectively. The third approach is to use the Tamura technique to record vital textural and structural features of the data. Both of these processes produce their own unique collection of features in two different feature files. Finally, the rich texture, frequency, and compactness information collected from the three separate extraction algorithms is combined with the two individual feature files to provide a unified, informative feature set. The combined final feature file provides a thorough representation of the original data, improving the precision and efficiency of subsequent analysis and classification.

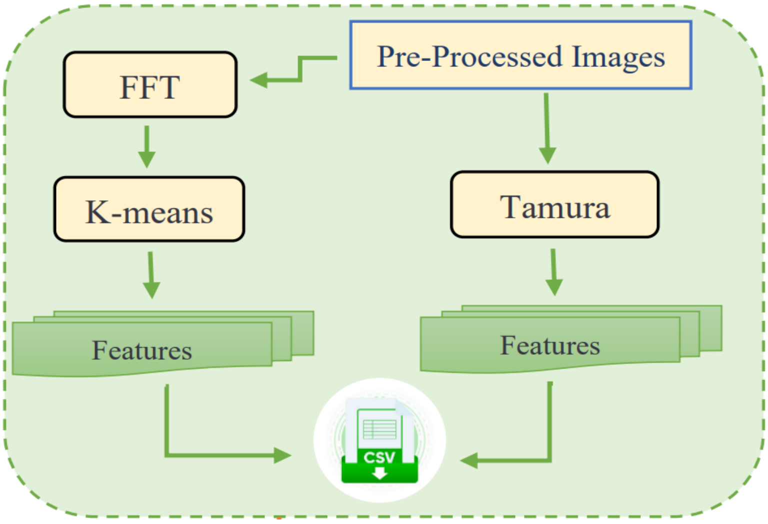


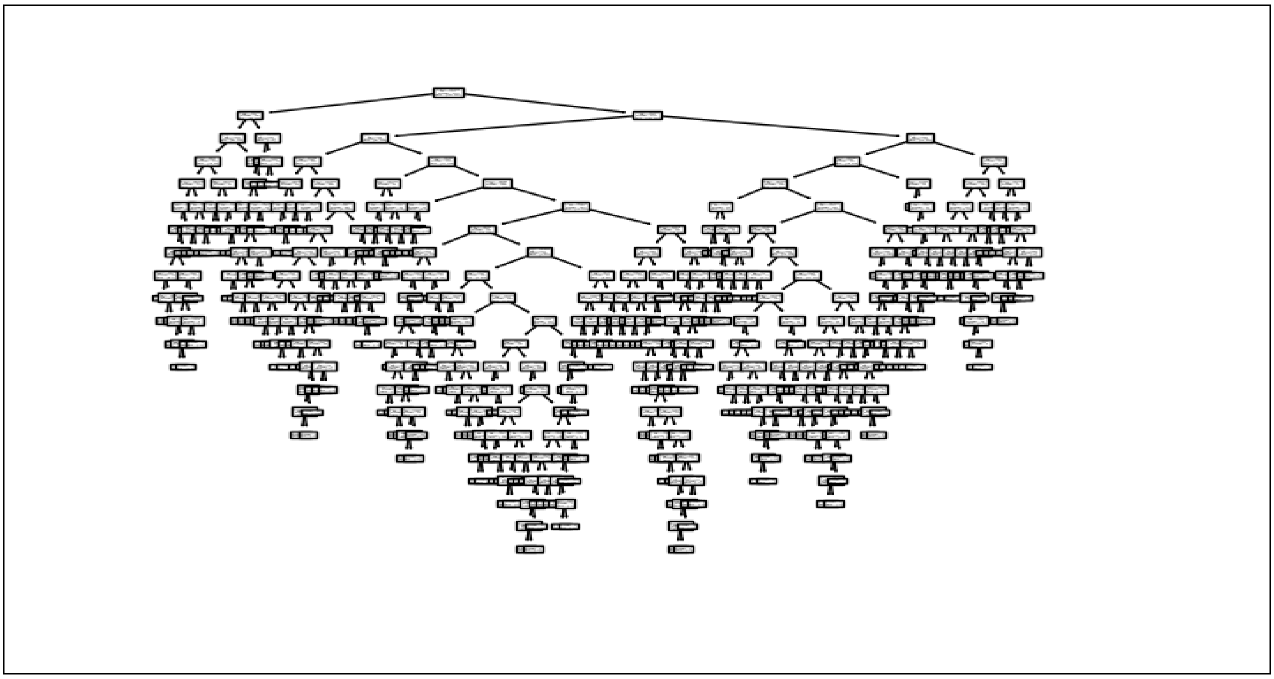
Figure (3.5): Integration of Feature Extraction.

3.2.4. Machine Learning Classification

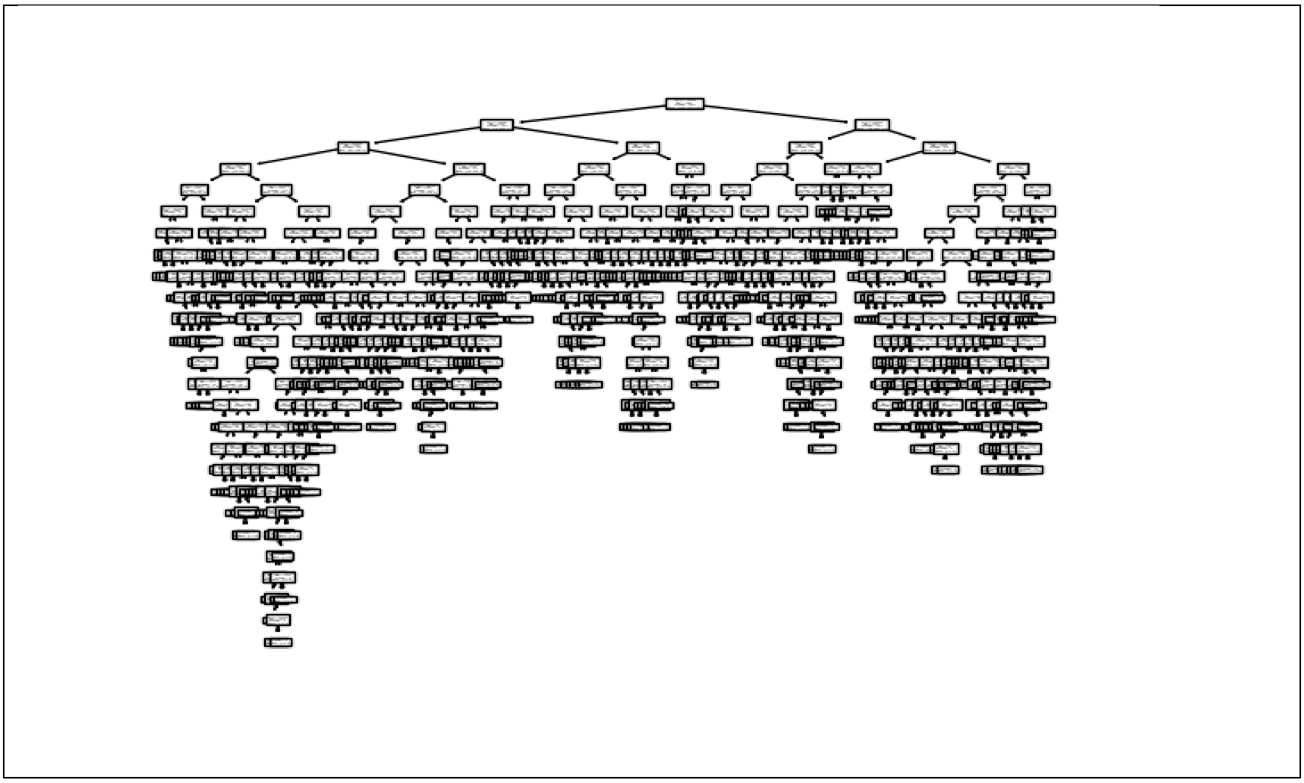
The tumor in the MRI scans will be recognized and classified using the retrieved features. The system's goal is to effectively classify brain tumor images utilizing machine learning classification algorithms such as Random Forest, and K-Nearest Neighbors (KNN).

3.2.4.1. Random Forest

The Random Forest approach provides a feasible option for detecting and classifying tumor types on MRI data. The procedure requires feeding in various attributes extracted from images as shown in algorithm (2-2) in Chapter 2, which provide a plethora of information on tumor properties. During the training phase, the Random Forest algorithm creates a large number of decision trees that rank different tumor kinds based on the attributes provided. The final classification is determined by merging all of these trees' predictions, which successfully determines a specific MRI image's estimated tumor type.



(A) Binary Classification

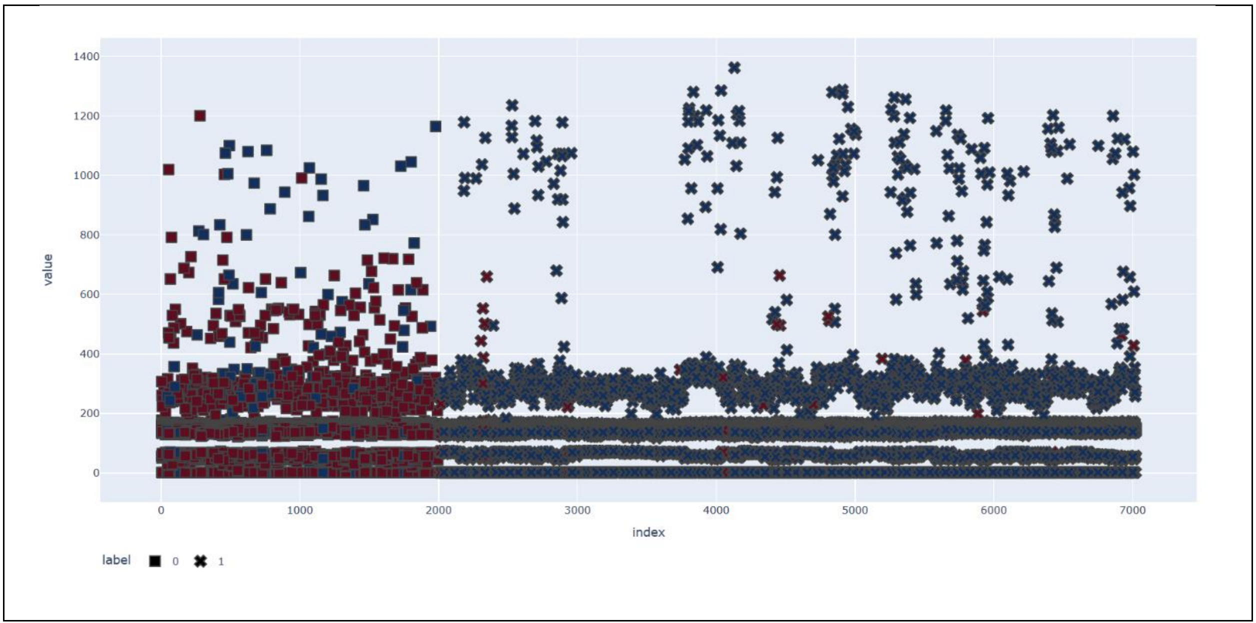


(B) Triple Classification

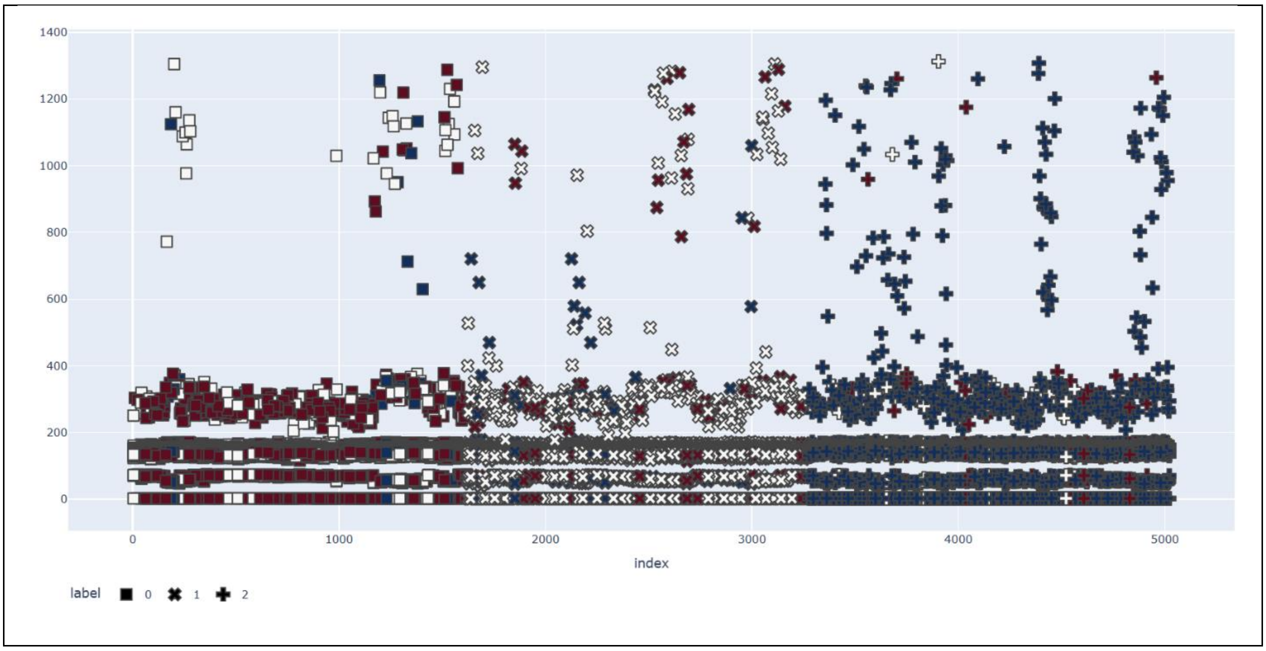
Figure (3.6): Random Forest Classifier

3.2.4.2. K-Nearest Neighbors (KNN)

The K-Nearest Neighbor technique could be used to determine tumor kind in MRI scans. This method is trained using labeled MRI images of known tumor types, which then calculates distances between neighboring targets and labeled images in a feature space Imaging experts can use the KNN algorithm (2-3) explained in Chapter 2 to classify the spatial distribution of brain tumors in MRI image processing, which adds greatly to progress in treating such neurological illness.



(A)Binary Classification



(B)Triple Classification

Figure (3.7): KNN Classifier.

3.2.5. 1D Convolutional Neural Network (1D CNN) Classification

One-dimensional Convolutional Neural Networks are extremely efficient in processing images, making them perfect for tasks like brain tumor classification. Several steps must be taken in order to develop a 1D CNN model for this purpose. These retrieved attributes enable complex pattern detection within the dataset in terms of the tumor kinds present. Once these distinct features have been computed using preprocessing methods, they can be fed into different layers inside the neural network such as convolutional layers, pooling layers, or fully connected layers. The steps mentioned above contribute towards creating an effective 1D CNN architecture capable enough of handling such complex problems thereby enhancing its performance beyond doubt. Once trained, this model can be applied to classify tumor types using new input data. The extracted features are transmitted through various filters and transformations within the layers of the 1D CNNs which help detect specific patterns indicative of assorted tumor types. Consequently, classification is yielded as output based on collected information by detecting geometric structure variations from patients suffering from different tumors achieved via accurate brain tumor classification methodology empowered with 1D CNN's capacity that allows capturing intricate dependencies about complex temporal or sequential activities within datasets resulting in precise diagnoses without human intervention while reducing time constraints significantly.

3.2.5.1. CNN Model for Binary and Triple Classification

The proposed system includes the 1D CNN layer depicted in tables (3.1) and (3.2). Here is a detailed description of the 1D CNN layer's classification process.

- The first convolutional layer (conv1d_1) has 16 filters with a kernel size of 3.
- The LeakyReLU activation function (leaky_re_lu_1) introduces non-linearity to the output of the first convolutional layer.

- The first max pooling layer (max_pooling1d_1) reduces the spatial dimensions of the output by taking the maximum value within each pool size of 2.
- The LeakyReLU activation function (leaky_re_lu_2) introduces non-linearity to the output of the third convolutional layer.
- The second convolutional layer (conv1d_2) has 32 filters with a kernel size of 3.
- The second max pooling layer (max_pooling1d_2) reduces the spatial dimensions further.
- The third convolutional layer (conv1d_3) has 64 filters with a kernel size of 3.
- The LeakyReLU activation function (leaky_re_lu_3) introduces non-linearity to the output of the third convolutional layer.
- The third max pooling layer (max_pooling1d_3).
- The fourth convolutional layer (conv1d_4) has 128 filters with a kernel size of 3.
- The LeakyReLU activation function (leaky_re_lu_4) introduces non-linearity to the output of the fourth convolutional layer.
- The fourth max pooling layer (max_pooling1d_4) reduces the spatial dimensions.
- The LSTM layer (lstm_1) is a recurrent layer that operates on the temporal sequence of the previous output.
- The LeakyReLU activation function (leaky_re_lu_5) introduces non-linearity to the output of the LSTM layer.
- The fifth max pooling layer (max_pooling1d_5).
- The fifth convolutional layer (conv1d_5) has 512 filters with a kernel size of 3.
- The LeakyReLU activation function (leaky_re_lu_6) introduces non-linearity to the output of the fourth convolutional layer.
- The sixth max pooling layer (max_pooling1d_6).

- The sixth convolutional layer (conv1d_6) has 1024 filters with a kernel size of 3.
- The LeakyReLU activation function (leaky_re_lu_7) introduces non-linearity to the output of the sixth convolutional layer.
- The second LSTM layer (lstm_2) is another recurrent layer that operates on the temporal sequence of the previous output.
- The seventh convolutional layer (conv1d_7) has 35 filters with a kernel size of 3.
- The Flatten layer (flatten_1) reshapes the output to a 1D tensor by flattening the spatial dimensions.
- The Dense layer (dense_1) and SoftMax activation function performs the final classification.

In general, the 1D CNN layer extracts hierarchical features from input sequences via a succession of convolutional and pooling processes, as shown in Figures (3.6) and (3.7). The LSTM layers capture temporal dependencies in a sequence, whereas the final convolutional layer and dense layer perform categorization. The network is better able to learn intricate patterns and improve classification accuracy by introducing non-linearity, as with the LeakyReLU activation function.

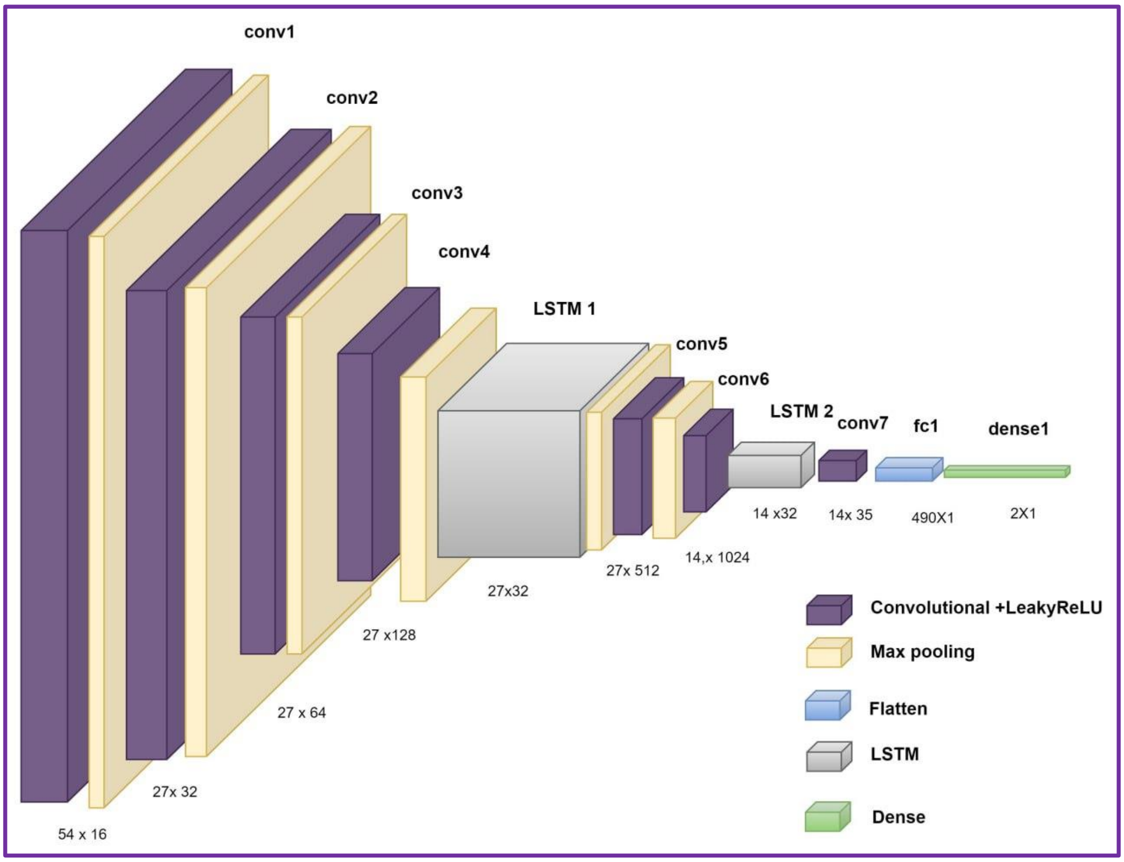


Figure (3.8): 1D CNN Binary Classification Model Layers.

Table (3.1): 1D CNN Binary Classification Model Layers.

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 54, 16)	64
leaky_re_lu_1 (LeakyReLU)	(None, 54, 16)	0
max_pooling1d_1 (MaxPooling1)	(None, 27, 16)	0
leaky_re_lu_2 (LeakyReLU)	(None, 27, 16)	0
conv1d_2 (Conv1D)	(None, 27, 32)	1568
max_pooling1d_2 (MaxPooling1)	(None, 27, 32)	0
conv1d_3 (Conv1D)	(None, 27, 64)	6208
leaky_re_lu_3 (LeakyReLU)	(None, 27, 64)	0
max_pooling1d_3 (MaxPooling1)	(None, 27, 64)	0
conv1d_4 (Conv1D)	(None, 27, 128)	24704
leaky_re_lu_4 (LeakyReLU)	(None, 27, 128)	0
max_pooling1d_4 (MaxPooling1)	(None, 27, 128)	0
lstm_1 (LSTM)	(None, 27, 32)	20608
leaky_re_lu_5 (LeakyReLU)	(None, 27, 32)	0
max_pooling1d_5 (MaxPooling1)	(None, 27, 32)	0
conv1d_5 (Conv1D)	(None, 27, 512)	49664
leaky_re_lu_6 (LeakyReLU)	(None, 27, 512)	0
max_pooling1d_6 (MaxPooling1)	(None, 14, 512)	0
conv1d_6 (Conv1D)	(None, 14, 1024)	1573888
leaky_re_lu_7 (LeakyReLU)	(None, 14, 1024)	0
lstm_2 (LSTM)	(None, 14, 32)	135296
conv1d_7 (Conv1D)	(None, 14, 35)	3395
flatten_1 (Flatten)	(None, 490)	0
dense_1 (Dense)	(None, 2)	982

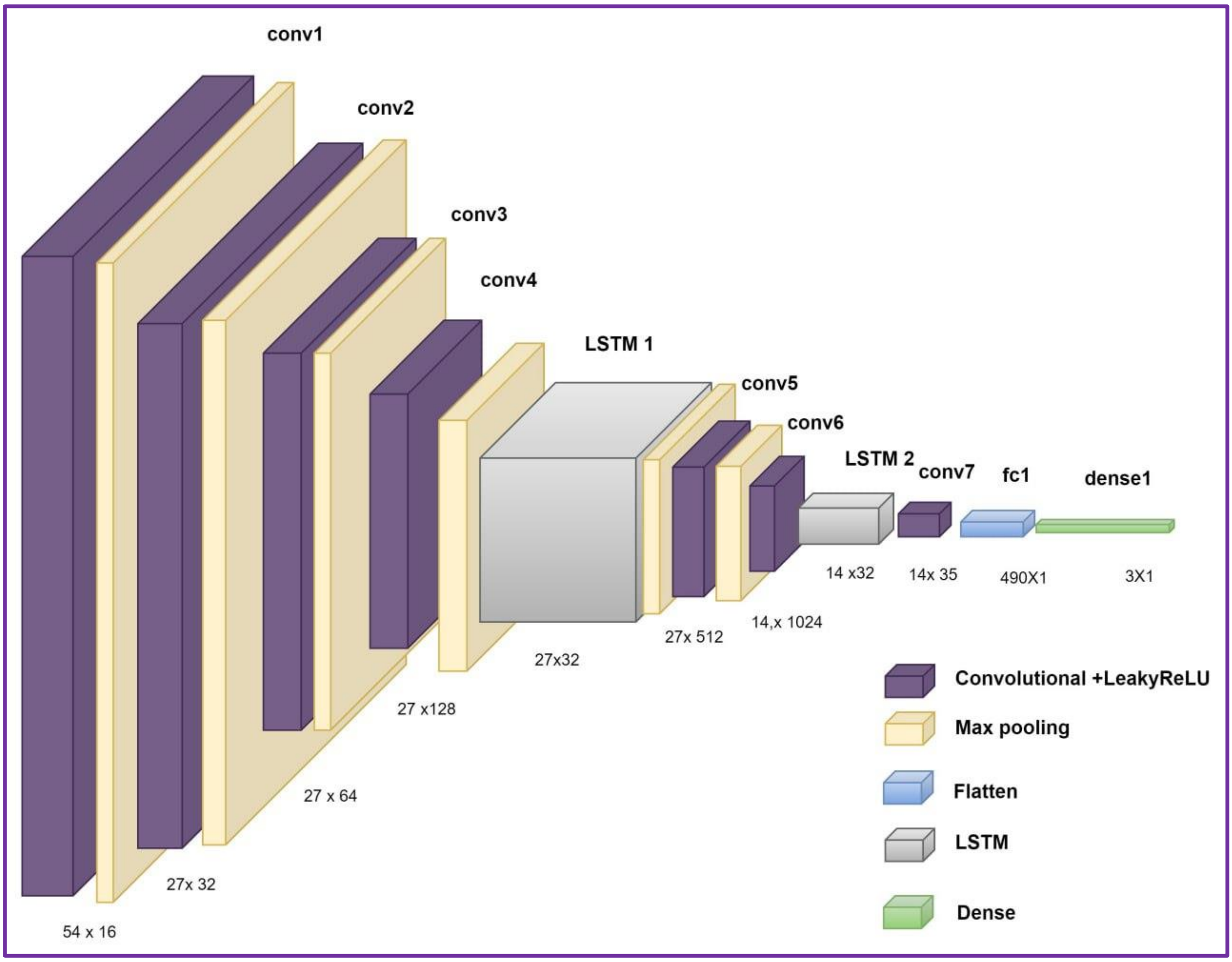


Figure (3.9): 1D CNN Triple Classification Model Layers.

Table (3.2): 1D CNN Triple Classification Model Layers**.**

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 54, 16)	64
leaky_re_lu_1 (LeakyReLU)	(None, 54, 16)	0
max_pooling1d_1 (MaxPooling1)	(None, 27, 16)	0
leaky_re_lu_2 (LeakyReLU)	(None, 27, 16)	0
conv1d_2 (Conv1D)	(None, 27, 32)	1568
max_pooling1d_2 (MaxPooling1)	(None, 27, 32)	0
conv1d_3 (Conv1D)	(None, 27, 64)	6208
leaky_re_lu_3 (LeakyReLU)	(None, 27, 64)	0
max_pooling1d_3 (MaxPooling1)	(None, 27, 64)	0
conv1d_4 (Conv1D)	(None, 27, 128)	24704
leaky_re_lu_4 (LeakyReLU)	(None, 27, 128)	0
max_pooling1d_4 (MaxPooling1)	(None, 27, 128)	0
lstm_1 (LSTM)	(None, 27, 32)	20608
leaky_re_lu_5 (LeakyReLU)	(None, 27, 32)	0
max_pooling1d_5 (MaxPooling1)	(None, 27, 32)	0
conv1d_5 (Conv1D)	(None, 27, 512)	49664
leaky_re_lu_6 (LeakyReLU)	(None, 27, 512)	0
max_pooling1d_6 (MaxPooling1)	(None, 14, 512)	0
conv1d_6 (Conv1D)	(None, 14, 1024)	1573888
leaky_re_lu_7 (LeakyReLU)	(None, 14, 1024)	0
lstm_2 (LSTM)	(None, 14, 32)	135296
conv1d_7 (Conv1D)	(None, 14, 35)	3395
flatten_1 (Flatten)	(None, 490)	0
dense_1 (Dense)	(None, 3)	1473

CHAPTER FOUR

EXPERIMENTAL RESULTSAND DISCUSSION

Chapter Four
Experimental Results and Discussion

4.1. Introduction

This chapter shows the experimental data and explains the findings of our ML and deep learning (DL) research. The primary focus of this chapter is on evaluating the efficacy of the proposed approaches and algorithms for detecting and categorizing brain tumors in MRI images. By conducting comprehensive trials and analysis, the objective is to investigate the potential benefits of ML and DL techniques in enhancing the diagnostic and decision-making capabilities of medical professionals. There are details on the data's origins, collection techniques, and early processing. The dataset includes the number of patients, types of tumors, and imaging modalities used in the experiment to provide you a thorough knowledge of the trial's methodology.

It also gives a thorough examination of the proposed ML and DL models' performance metrics. The models' ability to detect and classify brain tumors is assessed using a number of key parameters, including accuracy, precision, recall, and F1 score. These metrics are critical for evaluating the models' capacity to diagnose cancer and classify tumors into subgroups. The models' overall performance is also evaluated, and they are compared to current state-of-the-art methodologies. Also, thoroughly examine the models' capabilities and flaws to gain a better understanding of their performance. These investigations contribute to a comprehensive evaluation of the offered methodologies and provide useful development insights.

4.2 Hardware and Software Requirements

The process of creating, training, and testing the system calls for the use of specialized software and hardware.

4.2.1 Hardware Requirements

The suggested brain tumor categorization system makes use of HP computers with specifications like an Intel(R) Core i5- 1135G7U @ 2.42 GHz processor, 8 GB of RAM, Windows 10 operating system, and a 64-bit architecture.

4.2.2 Software Requirements

The suggested system was built using Python 3.6.5. Python makes it simple to write ANN programs. The system makes use of open-source libraries such as Open, Scikit-Learn, and Pandas. The Google TensorFlow framework (an open-source library designed for working with tensors) and the Keras library (an open-source neural network library built in Python based on TensorFlow) are also used. These libraries handle data analysis and machine learning tasks expertly. Java is used to create the graphical user interface.

4.3. Description of Brain Tumor Dataset

The study used a database of 7,023 MRI scans of the human brain that were categorized as glioma, meningioma, no tumor, or pituitary gland. The following images were discovered in each grouping: glioma (1,621), meningioma (1,645), no tumor (2,000), and pituitary (1,757) images. The images have 24-bit depth, different size like (225 pixel height with 215 pixel width) and horizontal resolution 96 dpi with vertical resolution 96 dpi

Table (4.1) summarizes the dataset used in brain tumor identification and classification from MRI images, which was created by combining these three datasets.

Table (4.1): A Brief Description of Brain Tumor Dataset.

Type	**No of Images**
glioma	**1621**
meningioma	**1645**
no tumor	**2000**
pituitary	**1757**

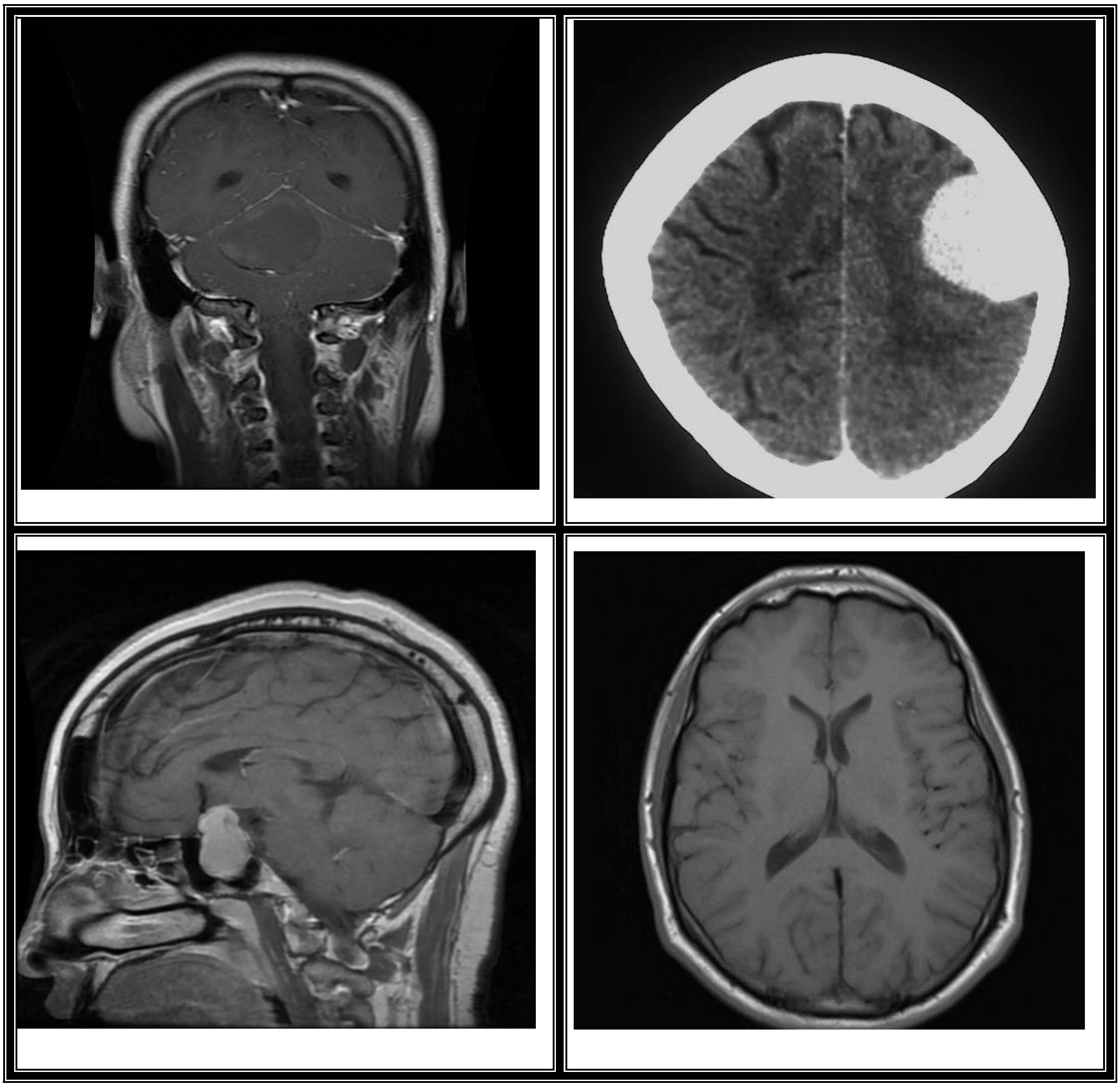


Figure (4.1): The Original Dataset.

To classify the proposed system based on whether a tumor is present or not, the data will be reorganized. If there is a tumor, it is then further classified into three types: glioma, meningioma, and pituitary; as demonstrated in tables (4.3) and (4.4).

Table (4.3): Dataset for Binary Classes (Tumor, No Tumor).

Type	No of Images
tumor	**5023**
No tumor	**2000**

Table (4.4): Dataset for Triple Classes (Tumor Types).

Type	No of Images
glioma	**1621**
meningioma	**1645**
pituitary	**1757**

4.4. Proposed System Results

The proposed system involves a sequence of three phases, with each individual phase comprising multiple steps. This section presents an overview of the entirety of these phases for the entire system.

4.4.1. Data Preprocessing Results

The system uses a variety of image processing techniques to preprocess the MRI data before using machine learning or deep learning algorithms for tumor detection and classification. For a normal dataset, three procedures are suggested: converting images to grayscale, histogram equalization, and image scaling. When abnormal dataset is subjected to all of the preceding steps, including gaussian blur preparation.

4.4.1.1. Convert Images to Gray Scale Results

The MRI images are transformed to grayscale in the early phases of processing, this procedure is described in Section 3.3.1.

Converting an image to grayscale, as shown in Figure (4.2), simplifies the representation while preserving important components for tumor detection and classification.

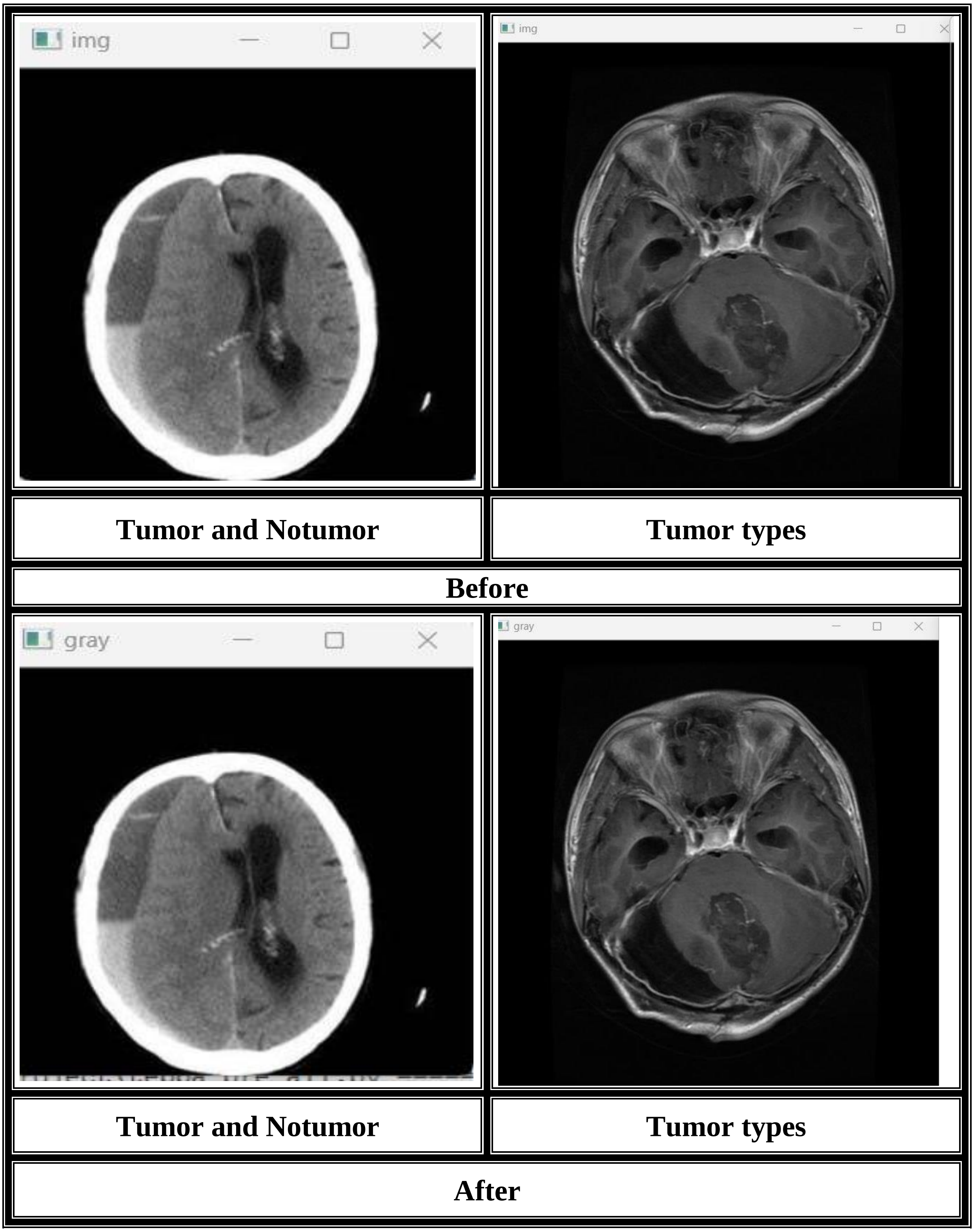
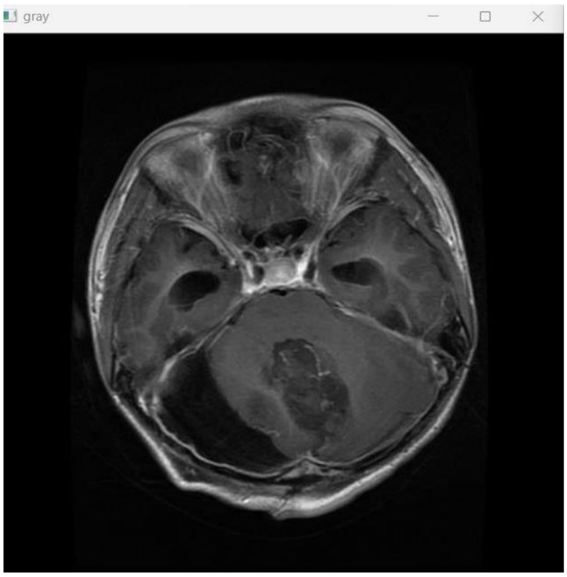


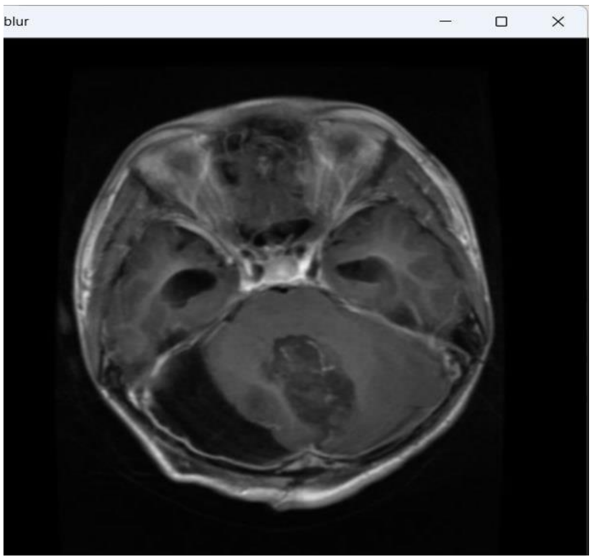
Figure (4.2): Convert Image to Gray Scale.

4.4.1.2. Gaussian Blur Results

The Tumor types images are then filtered using a Gaussian blur. The noise in an image can be diminished and small features or inconsistencies removed with the help of a smoothing technique called Gaussian blur. As can be seen in Figure (4.3), this procedure aids in reducing the effects of noise and other artifacts, leading to a more consistent and cleaner dataset for further study.



Tumor Types Before Applying Gaussian Blur



Tumor Types After Applying Gaussian Blur

Figure (4.3): Gaussian Blur Filter.

4.4.1.3. Histogram Equalization Results

After grayscale images are converted, histogram equalization is conducted. Histogram equalization improves image contrast by shifting the distribution of pixel intensities throughout the entire range. As shown in Figure (4.4), this strategy helps to make important details more visible and increases overall image quality, making it easier for machine learning and deep learning algorithms to extract meaningful information.

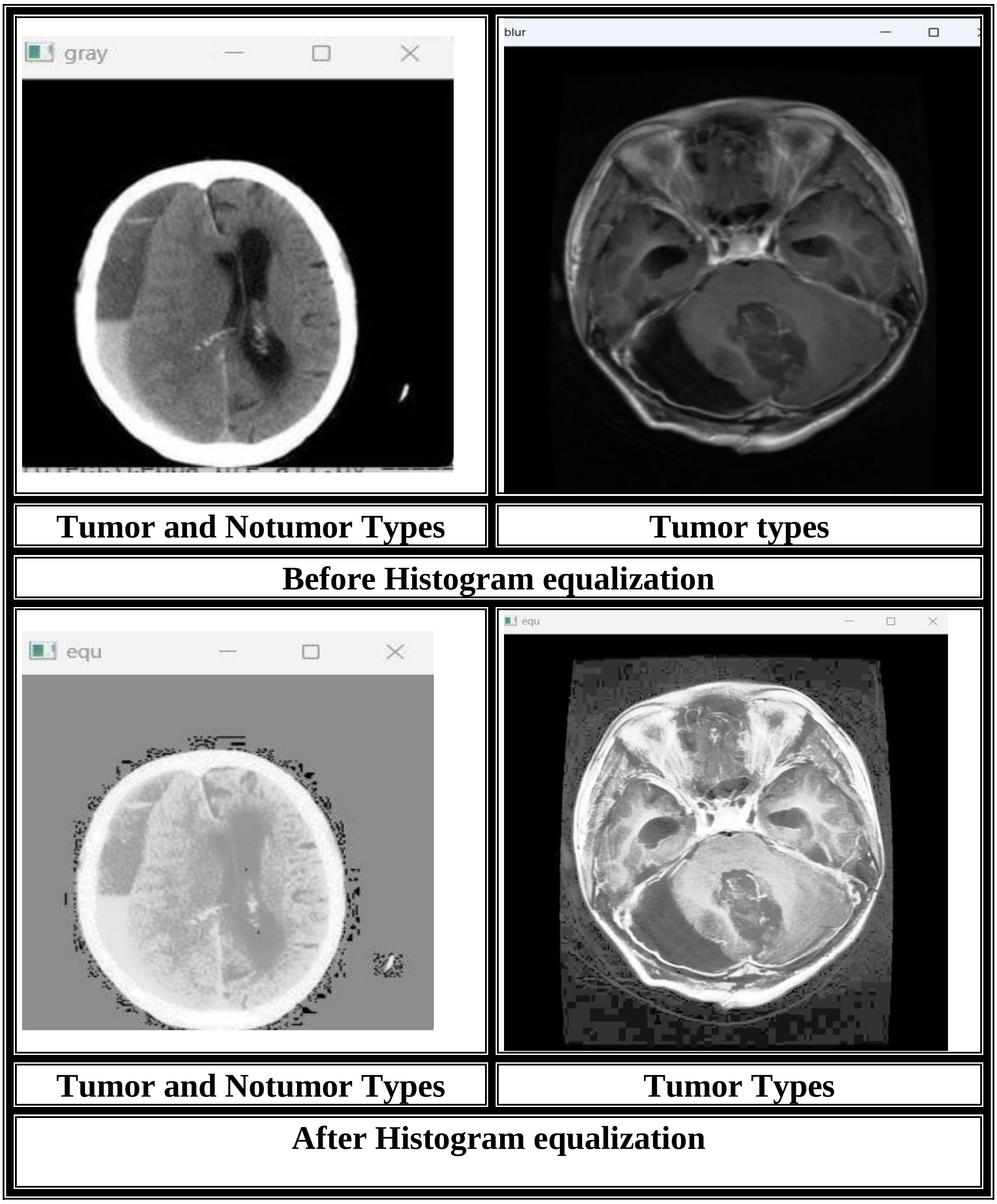
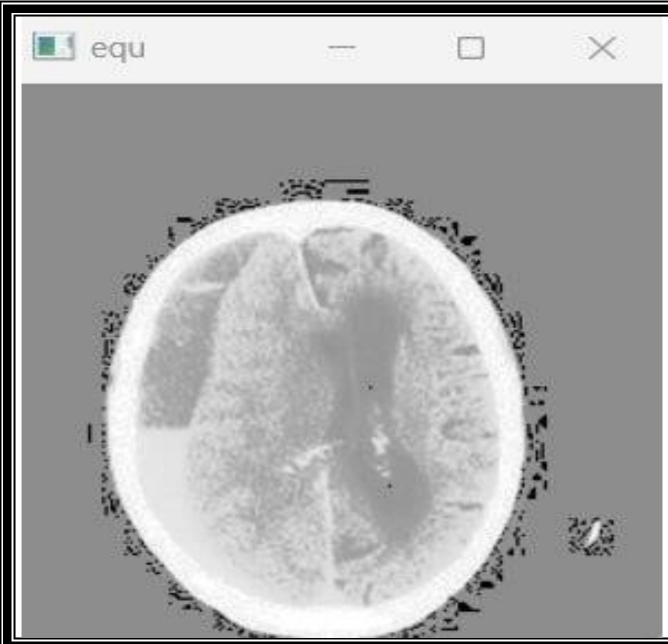


Figure (4.4): Dataset Histogram Equalization.

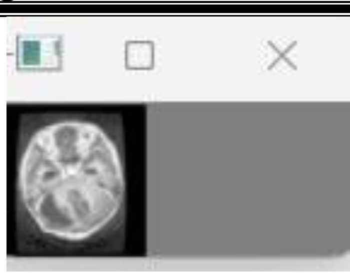
4.4.1.4. Image Resize Results

The processed images are reduced to a uniform 50x50 pixels in order to standardize their dimensions and decrease processing time. Downsizing the image size helps save on processing time without losing too much detail in the important areas. The resized images have 8-bit depth, 50-pixel height with 50-pixel width and horizontal resolution 96 dpi with vertical resolution 96 dpi, as can be seen in Figure (4.5). This procedure guarantees data uniformity and gets the images ready to be fed into ML and DL models.



Tumor and Notumor | **Tumor types**

(a) Before Resizing



Tumor and Notumor | **Tumor types**

(b) After Resizing

Figure (4.5): Dataset Image Resize(a) Before Resizing, (b) After Resizing.

4.4.2. Results of Features Extraction

This section, describes the outcomes of applying extracted feature approaches for the purpose of feature extraction in the context of brain tumor detection and classification from MRI images. This research makes use of the Fast Fourier Transform (FFT) and K-means, and used Tamura also for the purpose of feature extraction. These techniques are selected because of how effective they are at eliciting pertinent characteristics and cutting down the dimensionality of the dataset.

4.4.2.1. Feature Extraction using Fast Fourier Transform (FFT) and K-means

Fast Fourier Transform (FFT) and K-mean cluster algorithms were used to analyze MRI scans of brain tumors and extract useful information. In order to FFT convert a brain tumor MRI image from the spatial domain to the frequency domain. The result of FFT is 2D array (50×50) features of frequency domain for each image, as shown in Figure (4.6).

1.51E+02	1.29E+02	1.41E+02	1.34E+02	1.19E+02	1.15E+02	1.30E+02	1.52E+02	1.29E+02	1.55E+02	1.43E+02	1.4
1.22E+02	1.32E+02	1.33E+02	1.38E+02	1.28E+02	1.28E+02	1.23E+02	1.18E+02	1.28E+02	1.51E+02	1.25E+02	1.5
7.89E+01	1.23E+02	1.34E+02	1.36E+02	1.52E+02	1.38E+02	1.57E+02	1.41E+02	1.52E+02	1.37E+02	1.17E+02	1.1
1.48E+02	1.38E+02	1.42E+02	1.41E+02	1.52E+02	1.39E+02	1.48E+02	1.36E+02	1.49E+02	1.34E+02	1.21E+02	1.2
1.44E+02	1.19E+02	1.37E+02	1.43E+02	1.40E+02	1.29E+02	1.37E+02	1.51E+02	1.46E+02	1.48E+02	1.50E+02	1.4
1.41E+02	1.51E+02	1.40E+02	1.37E+02	1.37E+02	1.37E+02	1.41E+02	1.48E+02	1.39E+02	1.43E+02	1.42E+02	1.3
1.45E+02	1.28E+02	1.53E+02	1.43E+02	9.75E+01	1.40E+02	1.45E+02	1.09E+02	1.48E+02	1.37E+02	1.50E+02	1.4
1.54E+02	1.44E+02	1.37E+02	1.49E+02	1.54E+02	1.10E+02	1.48E+02	1.18E+02	1.40E+02	1.43E+02	1.52E+02	1.4
1.15E+02	1.45E+02	1.01E+02	1.10E+02	1.32E+02	1.28E+02	1.48E+02	1.53E+02	1.36E+02	1.28E+02	1.54E+02	1.5
1.44E+02	1.10E+02	1.58E+02	1.40E+02	1.31E+02	1.30E+02	1.47E+02	1.35E+02	1.40E+02	1.50E+02	1.51E+02	1.4
1.29E+02	1.16E+02	1.46E+02	1.43E+02	1.38E+02	1.49E+02	1.38E+02	1.28E+02	1.41E+02	1.20E+02	1.48E+02	1.5
1.38E+02	1.34E+02	1.37E+02	1.35E+02	1.17E+02	1.39E+02	1.24E+02	1.23E+02	1.50E+02	1.17E+02	1.40E+02	1.4
1.38E+02	1.45E+02	1.53E+02	1.60E+02	1.46E+02	1.32E+02	1.49E+02	1.48E+02	1.44E+02	1.29E+02	1.23E+02	1.4
1.37E+02	1.55E+02	1.55E+02	1.29E+02	1.48E+02	1.39E+02	1.47E+02	1.42E+02	1.43E+02	1.28E+02	1.53E+02	1.4
1.56E+02	1.41E+02	1.56E+02	1.54E+02	1.56E+02	1.50E+02	1.33E+02	1.50E+02	1.39E+02	1.45E+02	1.49E+02	1.1
1.49E+02	1.51E+02	1.28E+02	1.34E+02	1.47E+02	1.36E+02	1.49E+02	1.47E+02	1.33E+02	1.45E+02	1.67E+02	1.5
1.16E+02	1.34E+02	1.52E+02	1.42E+02	1.41E+02	1.50E+02	1.42E+02	1.50E+02	1.59E+02	1.21E+02	1.44E+02	1.5
1.52E+02	1.55E+02	1.58E+02	1.50E+02	1.43E+02	1.50E+02	1.53E+02	1.17E+02	1.60E+02	1.50E+02	1.51E+02	1.6
1.58E+02	1.39E+02	1.55E+02	1.43E+02	1.43E+02	1.42E+02	1.45E+02	1.49E+02	1.56E+02	1.45E+02	1.54E+02	1.3
1.51E+02	1.49E+02	1.52E+02	1.44E+02	1.50E+02	1.61E+02	1.49E+02	1.44E+02	1.55E+02	1.61E+02	1.46E+02	1.3
1.56E+02	1.52E+02	1.13E+02	1.26E+02	1.44E+02	1.52E+02	1.50E+02	1.35E+02	1.39E+02	1.35E+02	1.41E+02	1.5
1.20E+02	1.44E+02	1.35E+02	1.56E+02	1.32E+02	1.21E+02	1.54E+02	1.30E+02	1.58E+02	1.49E+02	1.53E+02	1.4
1.35E+02	1.41E+02	1.53E+02	1.55E+02	1.13E+02	1.31E+02	1.55E+02	1.26E+02	1.54E+02	1.41E+02	1.51E+02	1.5
1.04E+02	1.31E+02	1.61E+02	1.51E+02	1.55E+02	1.58E+02	1.49E+02	1.36E+02	1.61E+02	1.57E+02	1.63E+02	1.3
1.32E+02	1.45E+02	1.42E+02	1.45E+02	1.42E+02	1.50E+02	1.55E+02	1.57E+02	1.57E+02	1.49E+02	1.53E+02	1.2

Figure (4.6): The Output of FFT.

The frequencies are clustered using the K-means clustering (2D to 1D data transformation) technique after the FFT is completed. To divide the data into a predetermined number of blocks. Each pixel group represents a unique set of tumor-characterizing frequency components.

Overall, the feature extraction technique given by the combination of Fast Fourier Transform (FFT) and K-means clustering is extremely effective at simplifying MRI scans of brain tumors while retaining all crucial frequency information. The set of 50 features for each image is used as input for classification algorithms, assisting the overall success of the brain tumor detection and classification system.

4.4.2.2. Feature Extraction using Tamura

The Tamura feature extraction technique is applied to the MRI scans in the dataset, resulting in a condensed feature space with only four features. These features properly capture the texture and shape of the tumor, allowing subsequent classification models to differentiate between the various tumor types. This set of features was chosen due to their utility in extracting useful features of brain tumors from MRI data.

Furthermore, by combining Tamura with the integration of Fast Fourier Transform (FFT) and K-means clustering approaches, the feature extraction process is improved. Because of the combination of these numerous methods, the total number of attributes is now 54. Combining FFT, K-means, and Tamura offers a more accurate representation of the inherent properties of brain tumor images. The combination of FFT, K-means, and Tamura produces a set of 54 features that can be used in classification techniques. This comprehensive set of features benefits the brain tumor detection and classification procedure, allowing for more precision and efficiency.

Figures (4.7) and (4.8) display the results derived from the extracted feature techniques, showing promising outcomes in differentiating between various forms of brain tumors. When used together, FFT and Tamura allow for the extraction of both frequency and morphological information that can help characterize tumor patterns.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
988	0	138.6395	140.4144	135.2993	140.2079	139.622	140.1353	143.9815	143.2145	140.2063	142.5054	142.3914	142.3581	142.3333	142.4259	147.2418	147.5501	145.0328	148.1422	147.2585	151.354	148.92
989	0	141.7868	137.9065	135.911	139.738	138.8546	142.4737	136.9049	141.1393	137.4864	140.5257	144.8388	139.7085	144.5809	148.856	147.8646	147.5594	149.7844	150.9573	150.0018	149.1804	150.03
990	0	137.8133	138.6204	139.8413	141.1347	140.1986	139.9278	137.086	136.9535	141.39	140.3136	137.6251	143.8546	141.7898	146.3433	148.2431	152.3935	152.731	153.0591	150.3024	152.218	153.2
991	0	140.239	141.7133	146.3374	139.2082	142.3986	143.6867	145.3773	140.3141	143.0501	143.1055	142.1483	142.5936	139.9886	143.483	145.7001	142.6185	143.3975	145.5613	142.0528	146.7085	145.74
992	0	138.4488	140.4796	134.561	135.2973	140.1677	143.5201	137.9229	138.1414	141.6825	141.238	140.1531	143.9821	145.508	143.7969	150.6442	150.5436	147.2604	147.0399	147.3342	147.4597	149.02
993	0	153.0104	148.7397	150.602	149.8092	152.7047	151.3234	154.0443	153.9331	158.1367	156.1663	154.0829	152.6282	151.0764	154.0952	156.4898	153.6407	153.7079	153.107	157.3699	154.3063	157.51
994	0	141.399	140.9303	135.1581	134.4925	140.2565	142.0027	137.4037	139.5442	141.2021	141.3537	140.0715	142.3299	144.9316	143.7061	149.8862	150.8434	147.2092	147.2681	147.8927	146.4229	148.69
995	0	136.6488	135.0422	135.6814	136.8001	138.8852	138.93	141.0945	138.266	147.2489	143.9643	146.182	146.1283	144.8463	142.7952	144.6308	149.4577	147.8532	145.3646	143.0965	140.9488	146.63
996	0	143.1394	142.6736	144.0837	143.0217	143.5232	147.94	148.1318	149.8791	146.3215	150.0711	147.6737	149.1257	148.1304	148.9503	147.1753	144.7208	149.8212	153.2346	153.9308	157.24	153.98
997	0	138.1466	132.4918	132.1474	135.685	138.4563	137.5187	132.278	134.3685	134.6293	136.3363	138.3408	137.9269	138.2516	133.8576	140.7406	138.0726	139.6416	143.9088	144.3567	142.8176	147.56
998	0	135.9108	142.001	134.7933	137.9396	138.1717	138.5486	138.8182	141.9733	142.6796	139.4415	143.8007	142.1696	146.7808	144.9975	144.8725	145.1547	142.5436	146.8058	149.5703	147.8559	146.06
999	0	132.9145	139.1497	134.3979	139.9503	137.0759	137.9843	136.7674	139.3232	143.0007	138.1974	141.124	141.0415	137.7512	140.9335	144.5761	142.3414	140.7369	145.9358	147.9644	147.8148	150.23
000	0	147.2248	148.3078	150.3234	150.5838	149.9015	150.5637	149.9305	151.8691	145.2051	149.4463	147.2562	147.7686	148.7001	148.7026	151.0306	151.6196	148.8531	152.9697	149.2995	153.5369	153.53
001	0	135.0722	134.8022	137.9231	135.7505	136.61	139.5357	135.1054	140.3672	141.296	145.5359	141.2086	144.2498	144.05	147.3855	147.9842	148.968	150.184	148.3422	142.8357	147.6019	149.60
002	1	131.5424	134.3664	134.8673	135.4528	134.936	136.5277	134.6828	137.9722	137.7065	138.7772	139.9048	138.4885	139.2783	137.7898	142.8248	140.0906	142.6147	144.8257	145.3982	145.6942	145.45
003	1	131.6768	136.1934	134.7019	135.5625	135.3768	135.1925	136.2317	134.9688	138.7935	138.618	140.4394	140.834	137.9599	143.118	142.2752	142.2799	145.0801	148.0033	145.01	148.3566	147.78
004	1	143.5389	136.7158	142.0702	141.7907	134.6138	138.3925	136.6856	137.3505	138.029	140.114	137.04	138.9173	138.6861	142.199	138.7422	140.4392	142.9122	142.1192	142.5619	139.9259	142.93
005	1	130.3486	128.6409	133.2315	132.608	130.3288	132.9139	133.573	131.8495	131.7539	134.2538	137.5971	137.9711	140.2272	138.7163	142.8775	139.1184	139.5259	140.0596	144.0572	142.6807	147.53
006	1	128.8143	134.7056	136.5352	134.4958	131.0626	136.2864	136.8088	133.8205	136.3589	138.2503	138.6895	138.028	140.0182	139.0784	138.5711	140.3441	144.9883	142.6769	145.7539	146.5129	150.72
007	1	130.067	129.7088	129.3746	129.3481	127.2202	132.524	133.4035	132.9782	134.8405	133.4018	135.2448	136.8423	136.1887	137.9999	133.9473	138.2933	136.4092	139.408	139.3562	143.0894	139.54
008	1	132.3648	132.9817	135.6612	134.2219	133.7028	134.9101	134.5223	134.6097	133.9261	134.8994	134.7042	137.7252	141.8191	140.4242	138.0766	143.5311	147.4038	143.0524	143.0584	145.6968	147.01
009	1	138.3924	138.0858	136.2912	132.8814	136.6945	138.5294	137.8561	137.8944	137.3732	138.2369	141.2768	138.9351	142.8518	143.2683	144.7735	141.7777	143.4856	143.4622	144.189	147.9207	149.57
010	1	140.3252	138.2152	139.8513	137.7674	142.0792	142.365	140.9426	138.6	141.1148	139.7214	140.8482	139.5707	141.7729	143.5913	145.2647	144.3197	143.1681	143.8263	147.6065	146.7935	146.77
011	1	136.299	136.5416	135.8647	136.5449	140.81	141.4249	138.7412	141.8859	142.0637	139.3453	142.3807	142.6329	139.3142	143.7452	146.1278	144.1231	145.7533	146.7186	144.6741	144.9189	148.80
012	1	132.965	137.3222	135.7164	133.5291	134.4666	138.1469	137.0887	132.9691	135.9311	140.9854	138.2085	138.4801	141.4495	136.3243	141.9794	143.6962	143.9544	143.2781	144.9547	146.6191	147.51
013	1	132.9068	136.9088	137.3018	133.2278	133.4542	136.7848	132.5012	135.861	131.9531	135.5943	139.4889	139.6683	135.3737	138.8005	141.0666	142.2144	142.8964	143.9972	144.1112	144.9052	148.2

Figure (4.7): Samples of Resulted Features Extracted Using FFT, Tamura and K-Means for Binary Classification.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
3257	1	130.7405	128.4944	129.7639	129.2667	132.2645	136.1598	129.9832	130.865	133.492	134.9726	135.5247	136.8182	134.3442	134.6727	134.7245	134.9241	132.9769	136.6554	141.4597
3258	1	134.6292	135.9521	140.4833	137.7678	132.2232	138.0836	137.4587	139.6224	136.5025	139.5498	138.4148	138.3543	136.8287	137.4246	140.06	139.6295	139.9946	139.4827	143.2845
3259	1	131.6742	132.278	128.3939	130.2305	131.333	133.8388	132.8145	138.1394	134.755	137.4835	137.9293	137.7088	138.3674	141.5003	143.6454	140.5815	138.3632	140.662	140.1377
3260	1	130.7252	131.4777	132.502	130.81	129.3059	131.119	130.8692	128.171	133.4757	132.0248	132.207	130.0163	133.4945	134.4627	131.3069	132.4939	135.3922	135.9209	136.4749
3261	1	134.2605	131.4728	130.1971	129.4548	131.8938	129.3886	129.4125	132.7651	131.3948	131.6866	133.3705	130.9891	129.4425	132.937	131.1938	135.576	137.3169	137.9503	137.9269
3262	1	130.876	132.2868	133.6912	131.4461	131.3921	133.184	134.3015	131.6969	132.9747	132.5955	130.11	132.6444	132.2421	132.4668	135.9278	134.7451	135.6993	138.3808	140.4191
3263	1	136.3749	138.0509	133.9855	139.7721	135.9697	134.7671	139.5034	139.6106	137.2121	134.3589	134.7238	136.8569	136.5608	137.0137	140.1449	137.1299	140.577	137.7917	139.6029
3264	1	127.3783	131.8997	132.8714	130.5874	132.4484	129.8299	133.866	130.5756	135.4026	136.8227	135.5748	136.5979	136.7298	139.0596	142.8196	140.9077	141.9485	140.6515	137.5398
3265	1	132.8554	127.6199	133.7311	130.6004	128.6731	129.9992	135.4176	134.2132	137.4857	136.9108	136.7528	139.2332	138.2882	137.1528	137.5851	139.9396	142.668	142.3243	141.8979
3266	1	127.8	131.5479	130.7375	131.8007	129.1049	126.843	129.7449	130.0904	135.5213	131.8418	135.0088	131.4893	130.48	133.333	135.136	136.3044	133.9803	138.0649	138.2483
3267	1	132.7245	132.0823	128.1362	130.4182	128.2154	135.7249	131.63	136.5044	137.412	133.8579	133.2208	135.8348	135.5856	134.8561	139.0008	139.7181	141.845	143.7776	143.2202
3268	2	134.4794	137.5353	138.8778	137.3556	137.126	137.6679	140.5994	138.8304	140.3017	138.6634	142.9065	140.5315	144.5621	148.2806	147.4505	147.4702	148.3356	149.0626	152.7337
3269	2	138.0546	140.9657	138.9939	137.878	136.5241	140.3662	133.8863	138.2655	138.9753	139.7012	143.308	140.0501	140.988	143.2695	143.3331	145.1264	145.7033	145.851	146.4499
3270	2	132.6615	137.0132	133.5049	134.2062	135.5784	134.8354	135.8696	136.5547	140.3824	137.1214	138.3736	135.919	138.9405	136.766	142.5037	139.9765	142.6748	141.995	143.7529
3271	2	142.6574	140.6531	138.238	137.4511	137.5079	138.5064	139.8539	136.8686	143.1406	144.1294	140.0293	144.8054	141.7399	148.2448	148.1454	150.713	149.8403	151.2223	154.3836
3272	2	138.2068	140.721	139.147	142.2461	140.7343	142.3683	139.6831	140.639	139.1141	141.9837	144.3988	144.7939	144.5585	145.6226	148.923	147.7274	150.1344	149.9717	153.6839
3273	2	136.7435	136.6216	138.3001	141.6554	137.264	141.2689	140.1412	140.4983	143.9236	144.222	144.7125	144.4264	144.5921	148.278	147.1801	152.1168	152.311	154.0788	155.3405
3274	2	139.3349	140.4477	139.0812	141.9892	138.748	134.9957	138.0401	140.7347	142.1036	142.0922	139.7585	142.7387	142.5977	144.6428	146.8371	142.8699	147.3944	142.8668	148.8472
3275	2	145.2521	141.5033	140.8276	143.8199	147.0145	145.5377	142.5384	145.0673	143.4138	145.9389	142.9431	149.6478	147.7719	148.9392	152.3836	148.9313	154.4797	151.9511	152.3697
3276	2	140.2777	142.1807	141.6928	144.8547	140.5755	140.5763	143.9968	141.7494	141.6957	143.5167	144.1992	142.6627	142.4844	148.7942	147.5793	148.237	150.293	150.3063	151.6551
3277	2	143.3352	140.1766	138.6328	136.5903	140.8819	137.526	138.5568	140.5325	139.7858	142.8892	142.3097	141.3695	143.5374	143.0278	146.9895	143.2123	146.6613	150.3949	150.2381
3278	2	132.4196	132.1258	129.789	133.5944	134.4565	135.5261	132.6141	132.3838	136.2904	138.5405	139.2607	136.1319	134.9075	140.7839	136.7336	137.7401	143.4013	141.8519	145.6104
3279	2	141.7242	140.4314	134.2415	141.1403	138.4678	138.0989	135.6955	137.146	141.4769	143.4604	144.5682	141.7067	142.2058	144.6911	144.5309	149.045	146.7723	147.5046	150.7277
3280	2	127.9412	131.7932	128.6209	133.5116	132.1162	128.9668	137.3276	134.3535	137.5635	135.8954	134.1652	136.8931	137.6507	138.0294	137.0688	140.685	138.5203	144.059	143.1315
3281	2	137.901	132.0617	133.2158	133.9639	136.2493	137.1952	138.1392	140.1278	137.9598	137.3152	137.1777	139.3358	139.9305	140.209	142.9749	146.6752	144.8505	147.0112	147.3739
3282	2	137.7077	135.1535	138.3469	137.2066	131.8443	138.8604	139.2792	140.0878	136.3329	139.3748	141.4729	141.0436	142.2426	143.305	145.9533	147.1896	148.1422	151.1953	150.474
3283	2	132.5319	133.3173	132.1839	131.9195	132.7644	133.2832	136.6184	137.1217	135.1837	136.9164	136.8497	142.9804	141.7657	144.7385	143.2853	147.639	147.2354	149.2289	151.5996
3284	2	130.6971	135.719	135.7726	138.7713	136.429	137.4371	136.8438	143.2425	143.033	141.2425	136.0479	142.7757	142.3599	142.5829	145.5192	148.5422	148.4626	145.5596	146.4292
3285	2	129.5143	134.7241	134.7591	136.4087	136.6497	133.7295	134.636	139.0786	139.142	139.7183	139.3793	140.8721	138.2715	138.9895	141.1983	144.2608	146.2103	143.4453	147.7634

Figure (4.8): Samples of Resulted Features Extracted Using FFT, Tamura and K-Means for Triple Classification.

Feature distributions for binary classes, in this case between tumor and nontumor cases, are shown in Table (4.5). The table breaks down the total number of samples by category. Class 0 has 5023 data points representing instances without tumors, whereas class 1 contains 2000 data points indicating cases with tumors.

The kinds of tumors in the sample are reflected in the distribution of characteristics for triple classes shown in Table (4.6). The number of samples for each tumor category is listed in the table. There are 1621 samples of one kind of tumor in class 0, 1645 samples of another kind of tumor in class 1, and 1757 samples of still another kind of tumor in class 2. Insights into the balance and distribution of data across the various tumor types are provided in the table, which is useful for further research and categorization.

Table (4.5): Features for Binary Classes (Tumor, No Tumor).

Class	**Number Of Samples**
0	**5023**
1	**2000**

Table (4.6): Features for Triple Classes (Tumor Types).

Type	**Number of Samples**
0	**1621**
1	**1645**
2	**1757**

4.4.3. Result of Classification Stage using ML

This part will look at the prediction model's results, which include the outputs of machine learning models (Random Forest and KNN) for brain tumor diagnosis and classification. The findings are divided into two categories: binary classification (tumor or no tumor) and triple classification (tumor types). The algorithms' overall effectiveness was ranked using accuracy, precision, recall, and F1-score.

4.4.3.1. Random Forest Classification Model

The Random Forest classification algorithm is used to assess how effectively our chosen brain tumor properties predicted each class. After features of interest have been identified, the data is then classified using a random forest. A maximum of 100 trees with one leaf and used Gini impurity spilt method are used in the classification algorithm. Ratio of (70:30) splits of the dataset are used in training and testing phases.

The Random Forest model that is used to classify the data received (54 features) from the feature extraction stage may be analyzed by computing the accuracy for each group utilized as input into the classification models. The models were highly accurate in categorizing brain MRI images as tumor or not tumors. For binary classification, the Random Forest model is evaluated at a 97% accuracy level as shown in Table (4.7) and Figure (4.9)

Table (4.7): Model Results for Binary Classification.

Method	Accuracy	Precision	Recall	F1-Score
RF	0.97	0.95	0.97	0.96

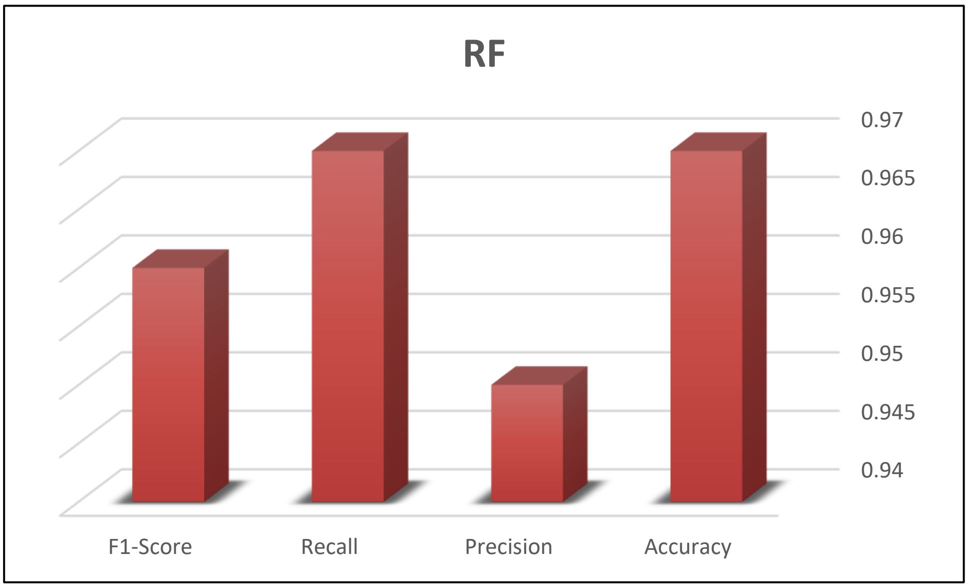


Figure (4.9): RF Model Results for Binary Classification.

The confusion matrices demonstrate how the ML models categorize data as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The model's efficacy in recognizing and categorizing cases of brain tumors can be further evaluated using statistical analysis of these data.

In the binary classification issue, the confusion matrix for Random Forest illustrates the percentages of correct and wrong classifications. According to the results, the model correctly detected "Tumor" in 335 occurrences and "Tumor" in 1208 instances. However, it identified "Tumor" as "Tumor" six times and "No Tumor" 36 times as "Tumor."

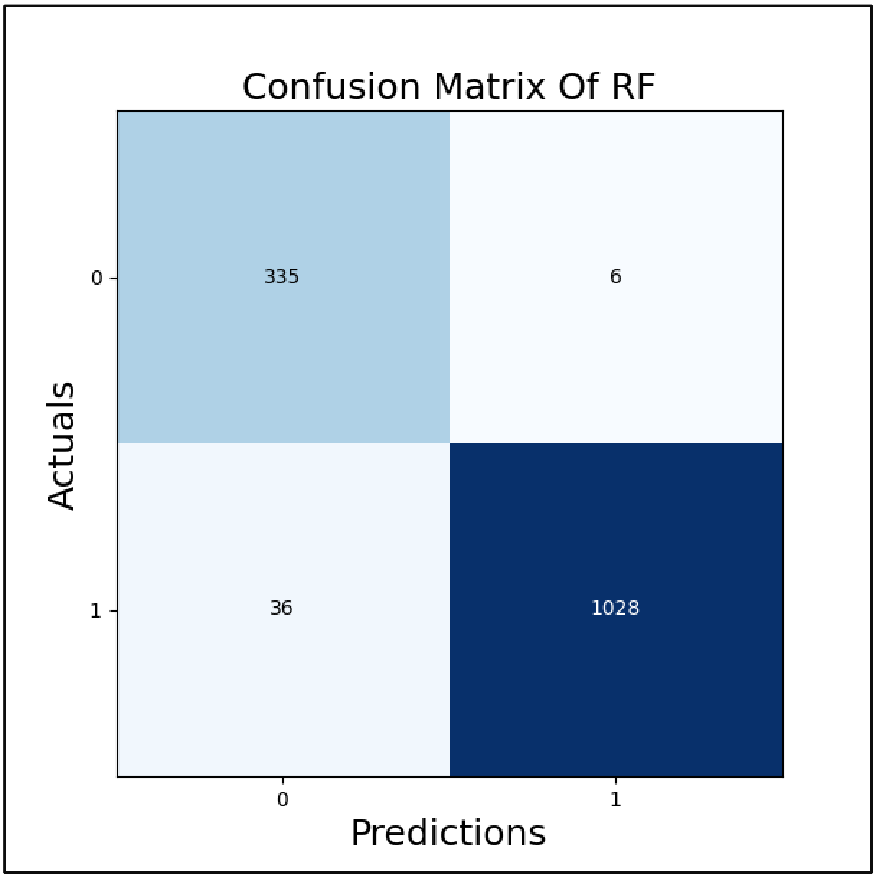


Figure (4.10): Confusion Matrix for Random Forest (Binary Classification).

For Trible classification, the Random Forest model is evaluated at an 80% accuracy level as shown in Table (4.8).

Table (4.8): Model Results for Triple Classification.

Method	Accuracy	Precision	Recall	F1-Score
RF	0.80	0.81	0.81	0.80

Figure (4.11) provides the following conclusions about categorization based on the RF model:

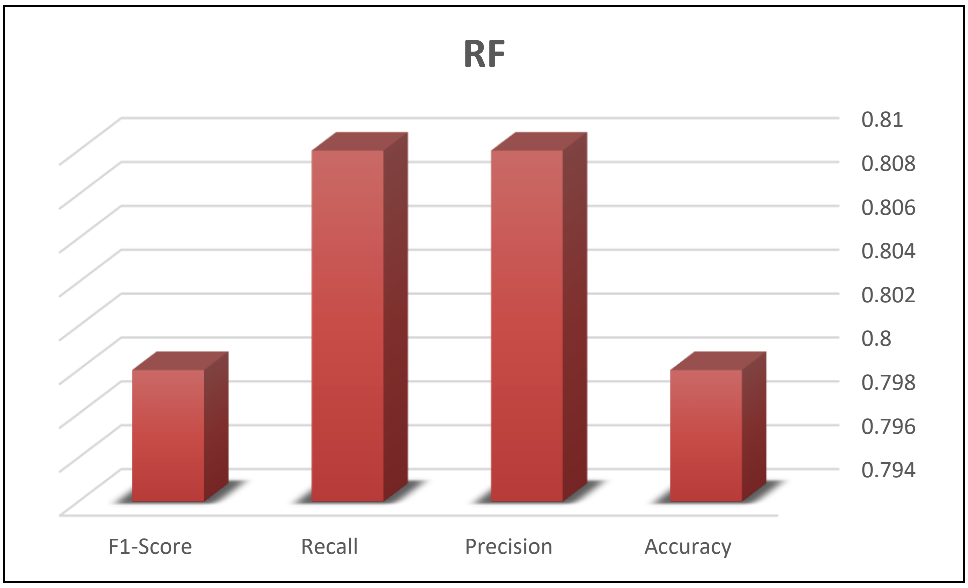


Figure (4.11): RF Model Results for Triple classification.

The confusion matrix for Random Forest in Triple classification problem as shown in Figure (4.12), the program was able to accurately predict.

- 210 cases as (glioma), 35 instance (glioma) classify incorrectly as (meningioma) and 13 instance (glioma) classify incorrectly as (pituitary).
- 272 cases classify true as (meningioma), 66 instance (meningioma) classify incorrectly as (glioma) and 10 instance (meningioma) classify incorrectly as (pituitary).
- 332 cases classify true as (pituitary), 43 instance (pituitary) classify incorrectly as (glioma) and 24 instance (pituitary) classify incorrectly as (meningioma).

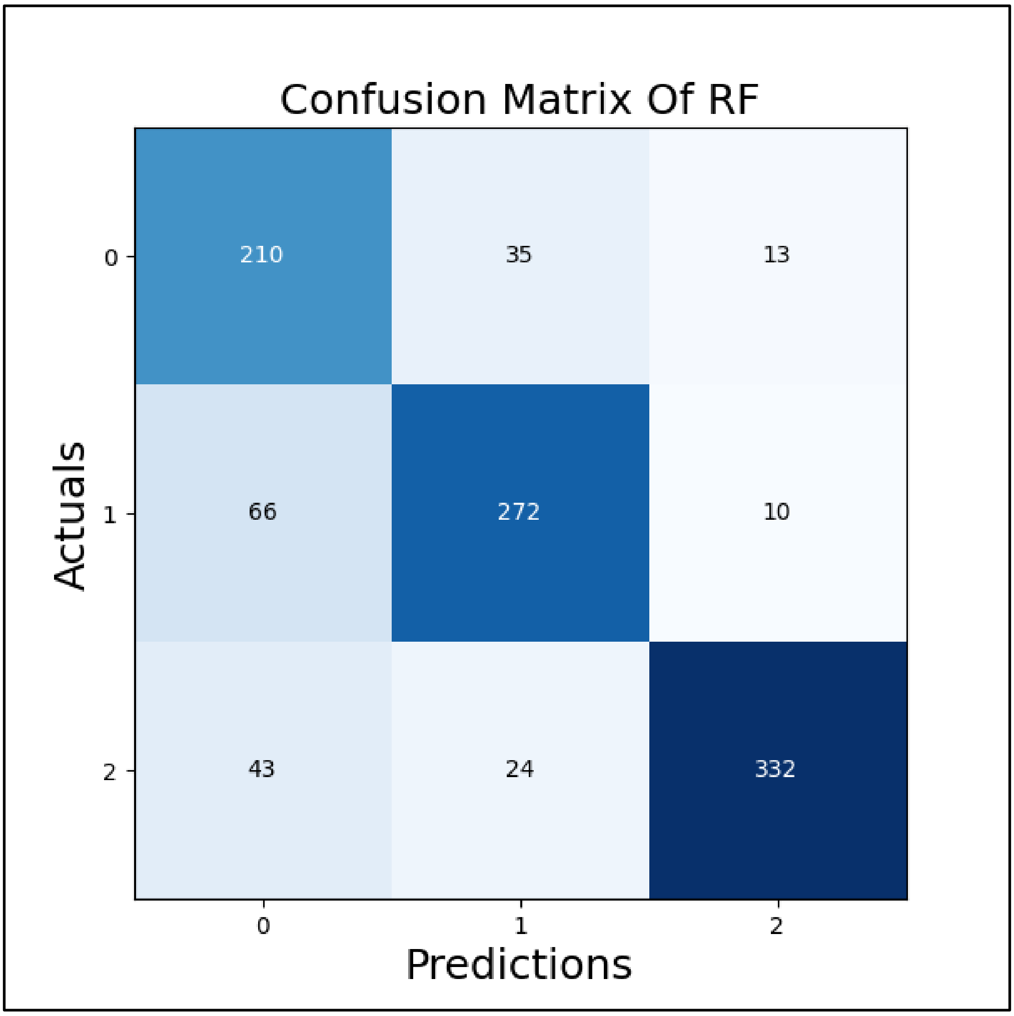


Figure (4.12): Confusion Matrix for Random Forest (Triple Classification).

4.4.3.2. Results of KNN Classification Model

The thesis uses KNN with a k-value of 5, which means that the categorization is determined by the combined votes of the five nearest neighbors with Minkowski method. In addition, a 70:30 split is made between the training and test sets of the dataset. This partitioning guarantees that the KNN model gets trained with 70% of the data, allowing it to discover the latent connections between characteristics and tumor types. To gauge the trained model's generalization abilities, the remaining 30% of the dataset is used for testing.

The features that are extracted from the feature extraction step was used to detect the tumor into using the KNN model.

The K-Nearest Neighbors (KNN) model for binary classification in table (4.9) performs well. The model labeled 93% of instances correctly. In addition, its precision score (0.90) indicates that the model predicted some condition 90% of the time. The model caught 92% of 100% true positives, according to recall. In the F1-Score of 0.91 shows this fine balance between precision and recall. In conclusion, the KNN model is suitable for binary classification tasks due to its fair precision/recall trade-off and average dataset accuracy. However, this model must be tested on other datasets and in real life before being proven effective.

Table (4.9): KNN Model Results for Binary Classification.

Method	Accuracy	Precision	Recall	F1-Score
KNN	0.93	0.90	0.92	0.91

The result of the KNN model for binary classification is shown in Figure (4.12).

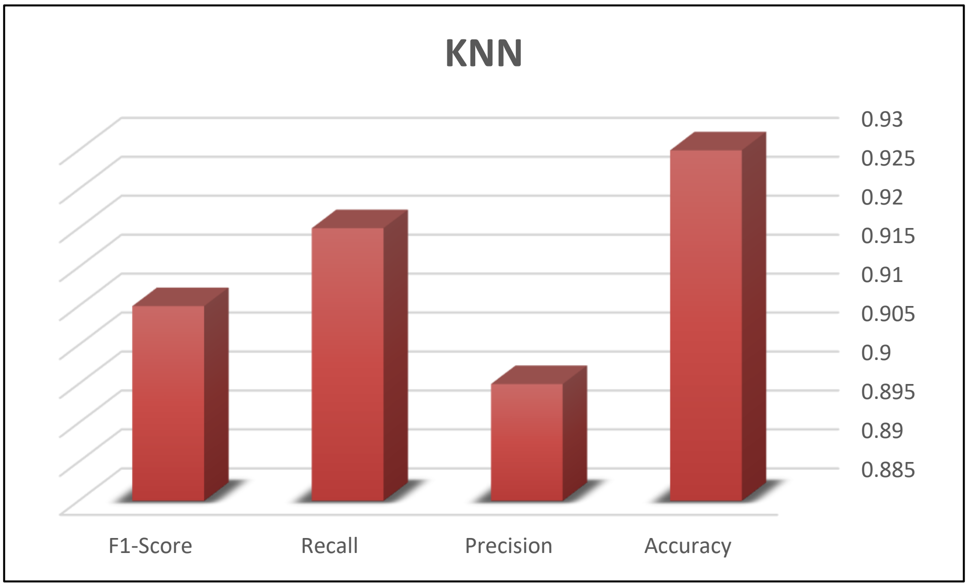


Figure (4.13): Confusion Matrix for KNN (Binary Classification).

The matrix displays the number of correctly classified cases (true positives and true negatives) and wrongly classified cases (false positives and false negatives).

There were 311 instances where the model correctly identified "Tumor," and 996 instances where it correctly identified "No Tumor," as shown by the confusion matrix for KNN in binary classification. There were a total of 60 false positives, out of which 38 were for tumors.

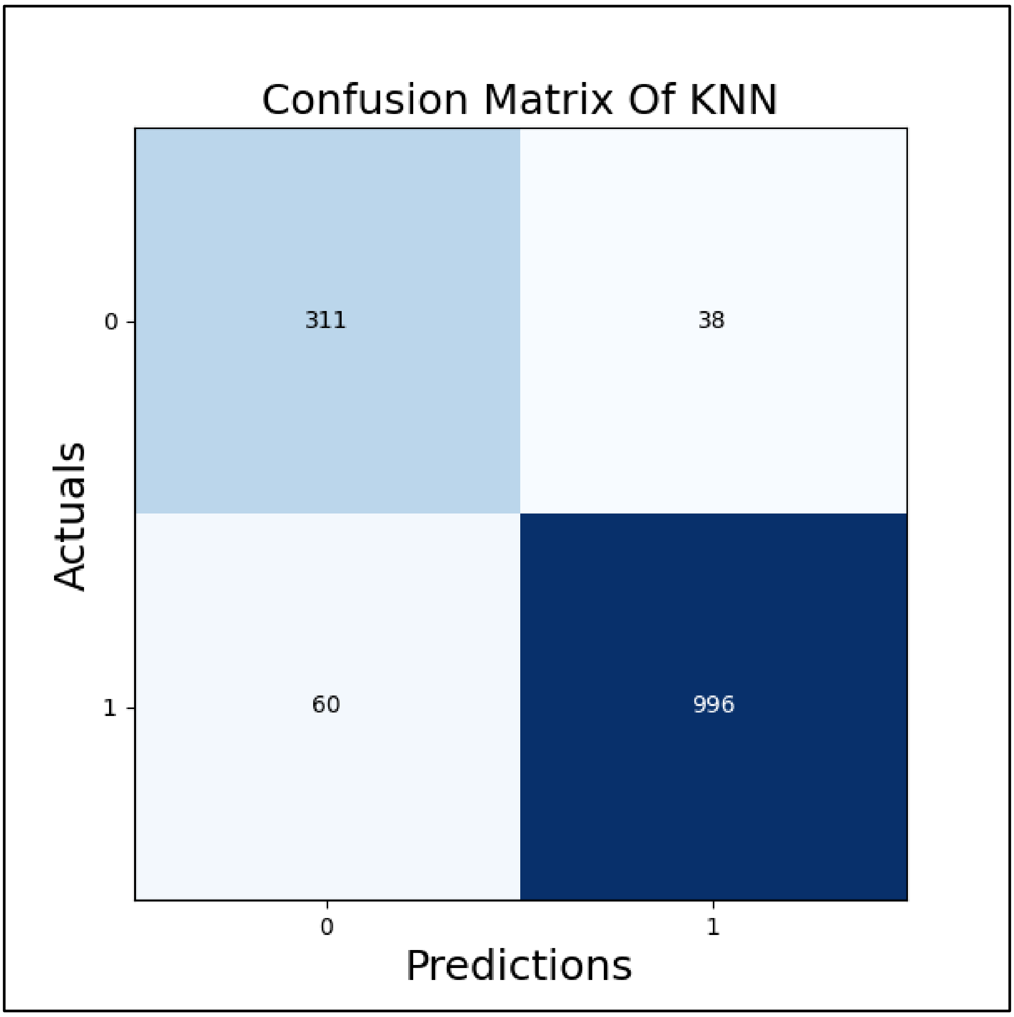


Figure (4.14): Confusion Matrix for KNN (Binary Classification).

The K-Nearest Neighbors (KNN) model used to perform anomaly classification from table (4.10), achieved an accuracy of 0.72 meaning it correctly identified 72% of these cases as abnormal. Precision, recall and macro F1-score are reported as .9433 (balanced), .85 (balanced) and .8764 (balanced). According to these results, the KNN model exhibits satisfactory performance for detecting outliers in the dataset. But a score of 0.72 means there's still significant room for improvement, and a less-than-perfectly-scoring model probably won't meet the needs of applications demanding a higher measure of precision or recall. Future work could involve further optimization and possibly more complex models to improve classification performance.

Table (4.10): KNN Model Results for Abnormality Classification.

Method	Accuracy	Precision	Recall	F1-Score
KNN	0.72	0.72	0.72	0.72

The accuracy of the KNN model came in at 72 % as shown in Figure (4.15)

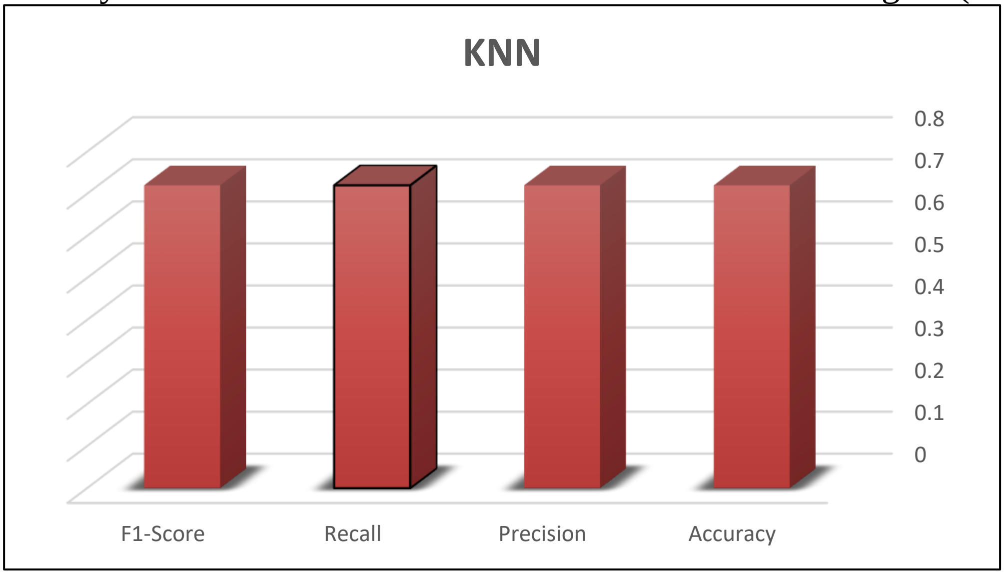


Figure (4.15): KNN Model Results (Abnormal Classification).

The K-Nearest Neighbors (KNN) confusion matrix shows how well the model performs when classifying brain tumors.

- 198 cases as (glioma), 75 instance (glioma) classify incorrectly as (meningioma) and 36 instance (glioma) classify incorrectly as (pituitary).
- 222 cases classify true as (meningioma), 93 instance (meningioma) classify incorrectly as (glioma) and 12 instance (meningioma) classify incorrectly as (pituitary).
- 307 cases classify true as (pituitary), 28 instance (pituitary) classify incorrectly as (glioma) and 34 instance (pituitary) classify incorrectly as (meningioma).

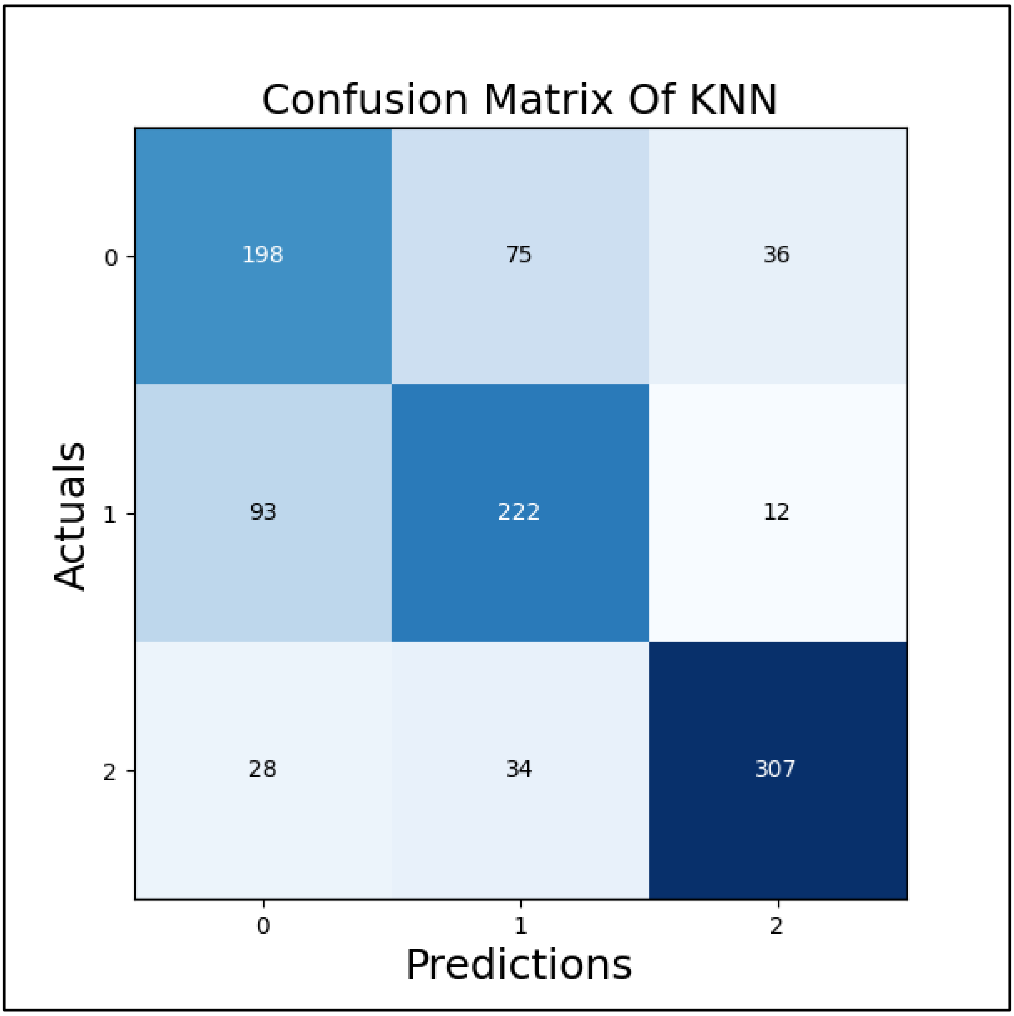


Figure (4.16): Confusion Matrix for KNN (Abnormal Classification).

4.4.3.3. Results of Hybrid Convolution Neural Network (Hybrid CNN) Model

A CNN model is used in this work to predict the existence of brain tumors. The Keras Python package is used to build the model because it provides a high-level neural network framework that allows for rapid prototyping and experimentation. Seven convolutional layers, seven LeakyReLU activation levels, six max-pooling layers, two long short-term memory layers, and one fully linked layer are used in this model.

The first layer is a convolutional layer with a 1x3 kernel size and 16 filters, followed by a LeakyReLU activation function. The next layer is a max-pooling layer. The activation function LeakyReLU is then used, followed by a second convolutional layer with a 1x3 kernel and 32 filters. The next layer is a max-pooling layer, the third convolutional layer uses a 1x3 kernel and 64 filters. Then a LeakyReLU activation function is used, followed by a max-pooling layer and a fourth convolutional layer with a 1x3 kernel and 128 filters, followed by a LeakyReLU activation function. Following that is a max-pooling layer, followed by an LSTM layer with 32 units and a LeakyReLU activation function. The fifth layer, after a max-pooling layer, is a convolutional layer with a 1x3 kernel and 512 filters. The LeakyReLU activation function is employed, followed by a max-pooling layer, a sixth convolutional layer with 1x3 kernel size and 1024 filters using the LeakyReLU activation function.

An LSTM layer with 32 units is next applied, followed by a seventh convolutional layer with a 3x3 kernel and 35 filters, and finally. Before making the final prediction in a softmax layer and a Dense activation function, the data is simplified in a Flatten layer. To train the CNN model, the loss function to use (binary cross entropy) for (binary class) and (categorical cross entropy) for (triple class) is optimized. The accuracy of the model is used to assess its performance, and an adaptive moment estimation (ADAM) optimizer with a learning rate of 0.001 is chosen for further investigation. The model is subjected to 100 epochs.

As evaluated by the precision, recall, F1-score, and support measures, the CNN model scored remarkably well on the binary classification issue. The precision score of 1 indicates that class 0 (no tumor) was properly classified in every instance expected. Class 0 had a recall score of one, indicating that the model correctly identified all tumor-free samples. The F1-score, accuracy, and recall for class 0 are all 1, indicating a satisfactory degree of both, the model showed perfect accuracy, recall, and F1-score for class 1 (tumor).

To examine its performance in this situation, the CNN model is tested on a dataset of three tumor classes (class 0, class 1, and class 2) for Triple Classification. Precision, recall, F1-score, and support are calculated for each category. The precision ratings of 1 show that the model successfully classified all examples into their appropriate classes. The model correctly identifies examples of all classes (as evaluated by the recall score of 1), which is 1. These findings demonstrate that the CNN model is very accurate and useful for recognizing and categorizing various types of cancers. The model's accuracy for Triple classification was 1.

4.4.4. Evaluating the Proposed Model

According to the prediction model, the three models employed to categorize brain MRI images (Random Forest, KNN, and CNN) all achieved outstanding accuracies for both binary and abnormality classification tasks. The Random Forest and KNN models achieved 97% and 93% accuracy rates for binary classification, respectively. Furthermore, the models' triple categorization accuracies ranged from approximately 72% to 80%. As evidenced by the results obtained for binary and triple classifications, the CNN model performed remarkably well in detecting and categorizing brain tumors and abnormalities. The model correctly detected tumor and tumor type cases with 100% accuracy and 0% false positives or false negatives, attaining high precision, recall, and F1-scores across the board. Overall, the CNN model's success in predicting the presence of brain tumors and abnormality kinds is further supported by the fact that it achieved an accuracy of 100% across both tests.

These findings show the promise of deep learning approaches, particularly CNN models, for medical image processing and brain tumor identification. The CNN model's ability to automatically learn discriminative features from preprocessed MRI images is a significant component in its outstanding performance. The obtained results indicate that the proposed ML and DL framework can be a valuable resource for medical practitioners in accurately diagnosing and classifying brain tumors and abnormalities. We will address additional evaluation and comparison with various prediction models and methodologies in the following sections to evaluate the robustness and generalizability of the proposed CNN model.

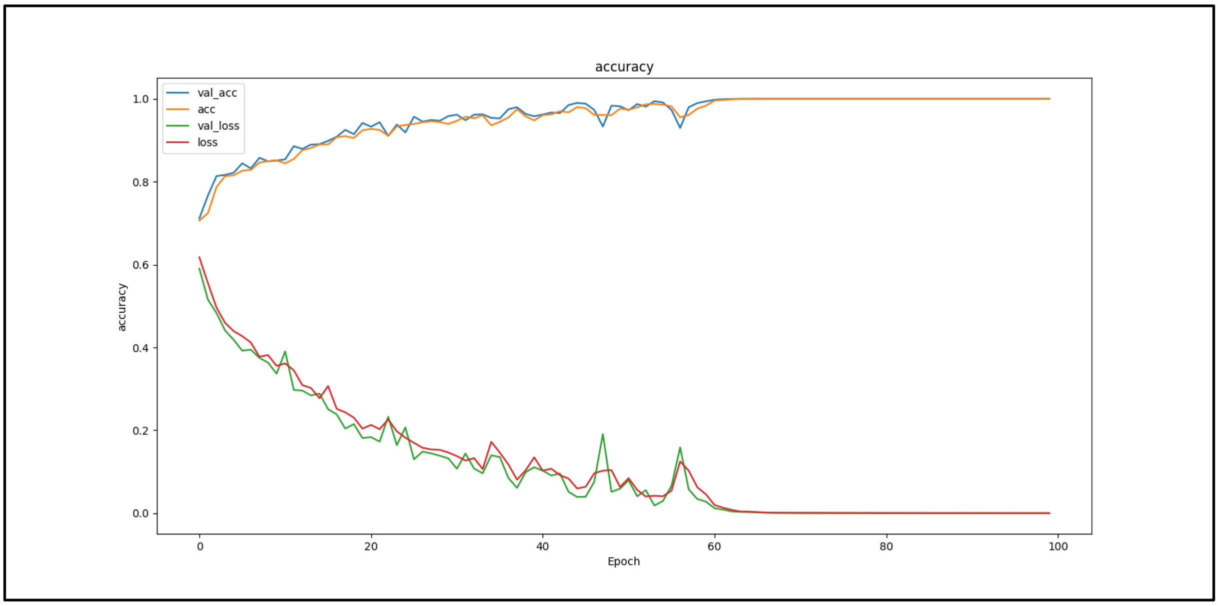


Figure (4.17): Accuracy and Loss for Binary Classification.

Figure (4.17) depicts the evolution of accuracy and loss on the binary classification task across 100 epochs. The slope of the accuracy curve indicates the model's ability to distinguish between tumor and no tumor. Lower values on the loss curve, which represents the model's optimization process, imply better performance. Examining this graph reveals the convergence of the model's accuracy and the tendency toward reduced loss as training progresses.

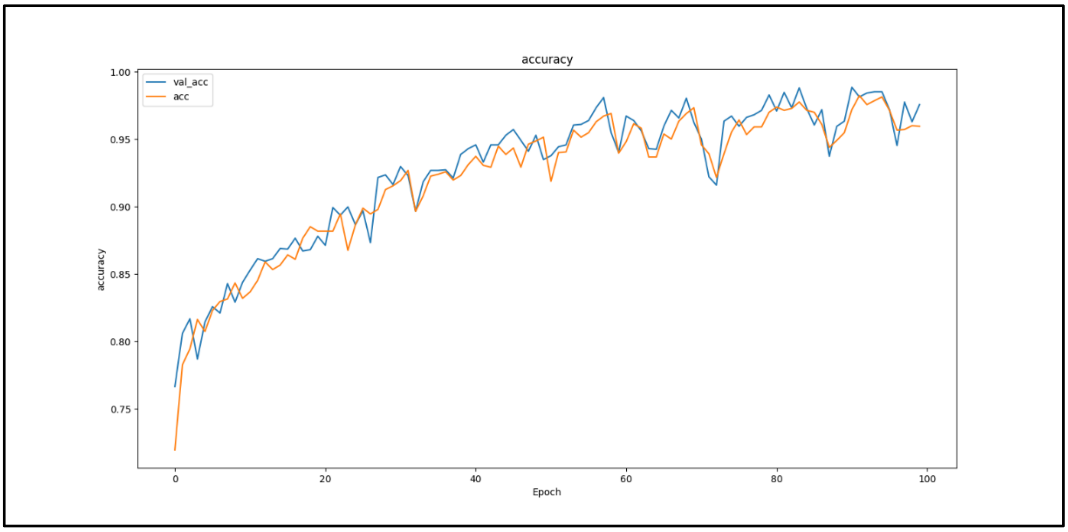


Figure (4.18): Accuracy for Binary Classification.

Figure (4.18), which focuses on the accuracy measure for binary classification, clearly shows the model's capacity to differentiate between tumor and no tumor.

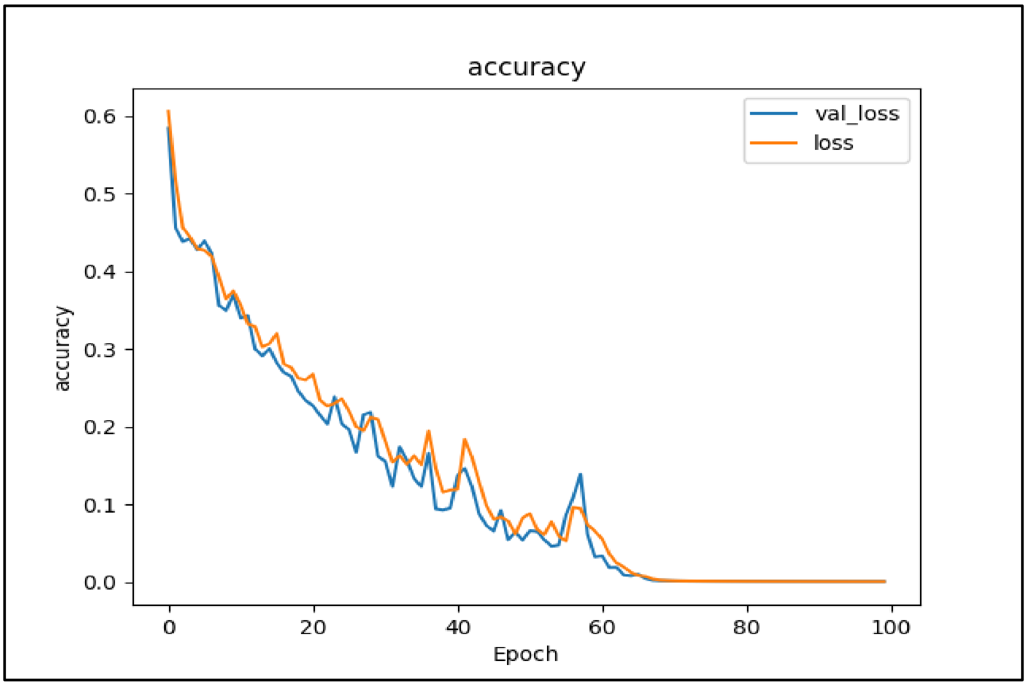


Figure (4.19): Loss and Val for Binary Classification.

Figure (4.19) depicts the binary classification loss and validation loss curves. One curve represents the training loss, while another curve represents how well the model performs on new data. If the validation loss is steady relative to the training loss, it indicates that the model can successfully generalize to new samples.

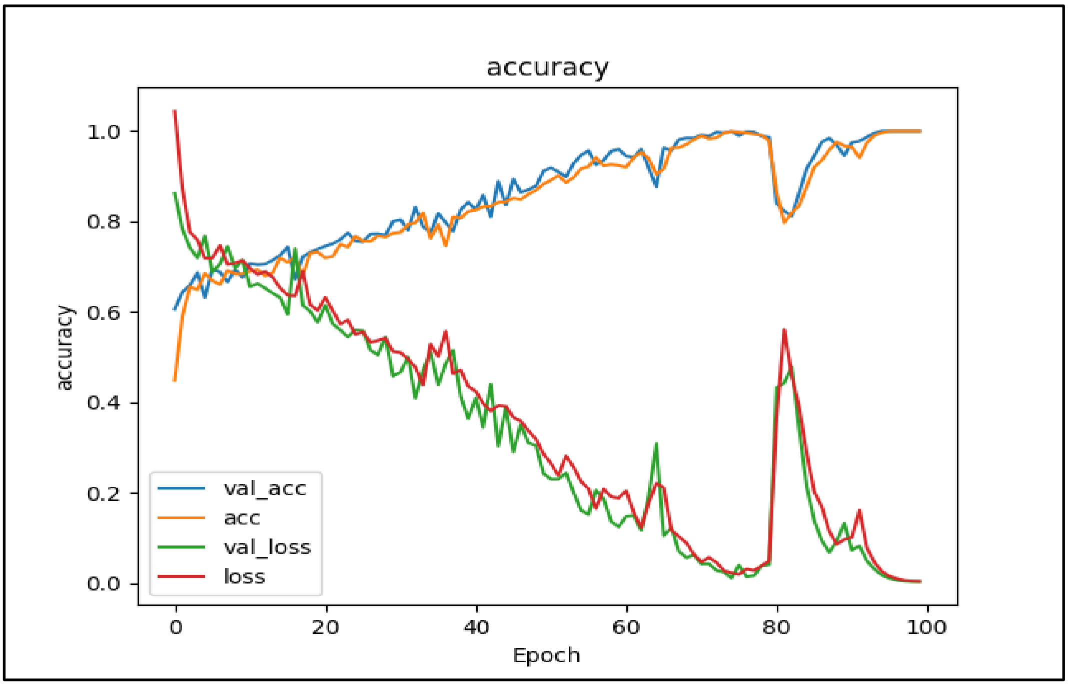


Figure (4.20): Accuracy and Loss for Triple Classification.

Figure (4.20) depicts the accuracy and loss patterns for the triple classification job. It demonstrates the model's ability to distinguish between numerous abnormalities seen in brain tumors. The accuracy curve shows how well a model detects different types of irregularities. The loss curve provides insight into the model's optimization process during training, with lower values signifying greater performance.

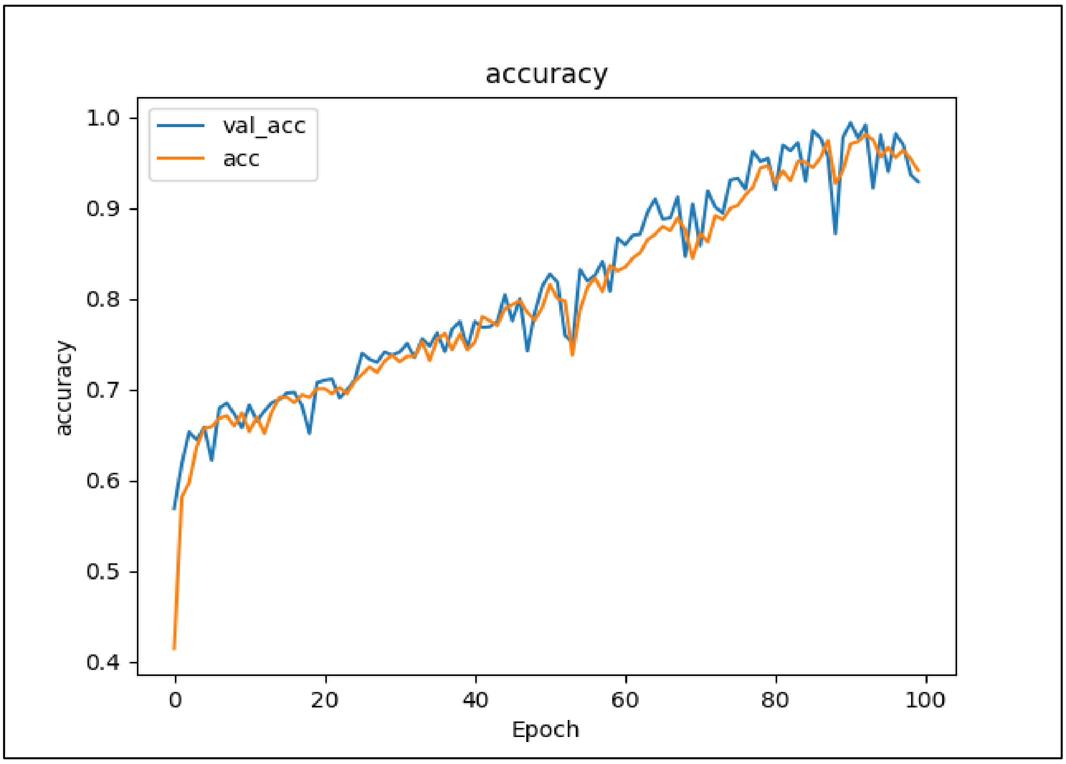


Figure (4.21): Accuracy for Triple Classification.

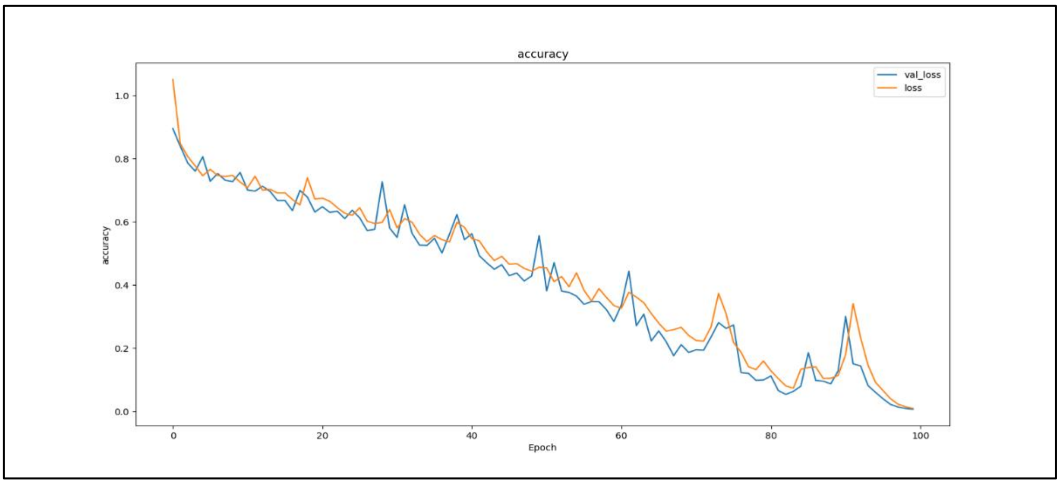


Figure (4.22): Loss and Val for Triple Classification.

Figures (4.21) and (4.22) show the loss and accuracy curves for triple classification. It displays how well the model can classify different types of tumors. The accuracy curve demonstrates how well the model can detect various types of problems. The loss curve demonstrates how the model is optimized over time.

4.5. Result Comparison with Other Studies

Machine learning and deep learning have been the subject of multiple studies in the field of brain tumor identification and classification. The table (4.11) below compares the methods utilized and the accuracy reached by some of the foundational publications in this field with the results provided in this thesis.

Table (4.11): Comparison of Related Works Performance**.**

References and work date	Dataset	Classification Method	Accuracy
Gómez-Guzmán et al. (2023) [14]	Brain Tumor MRI dataset Msoud	InceptionV3	97.12%
		ResNet50	96.97%
		InceptionResNetV2	96.78%
		Xception	95.67%
		MobileNetV2	95.45%
		EfficientNetB0	90.88%
		Generic CNN	81.08%
Patil and Kirange (2023) [15]	MRI brain image dataset	CNN-LSTM	99.1%
Nayak et al. (2022) [16]	Figshare	Dense Efficient-Net	99.97%
Raza et al. (2022) [17]	Figshare	DeepTumorNet	99.67%
Maqsood et al. (2022) [18]	Figshare	modified MobileNetV2 and Multiclass SVM	98.92%
	BraTS 2018		97.47%
Ullah et al. (2022) [20]	SARTAJ	InceptionResNetV2	98.91%
Deepak et al. (2021) [22]	Figshare	CNN and SVM	95.82%
Proposed	Brain Tumor MRI dataset Msoud	CNN	100%
Proposed	Brain Tumor MRI dataset Msoud	KNN	(93% , Binary) (72%, Triple)
Proposed	Brain Tumor MRI dataset Msoud	RF	(97% , Binary) (80%, Triple)

This experimental work together to improve the accuracy to diagnose and classify brain cancers. Despite variances in datasets, models, and methodologies, all of these methods indicate the promise of machine learning and deep learning in this domain. The proposed solution outperformed some of the mentioned research in terms of accuracy, proving the superiority of the proposed methodology. The findings demonstrated that the proposed algorithms could accurately classify pictures of brain tumors, which might aid in both diagnosis and therapy.

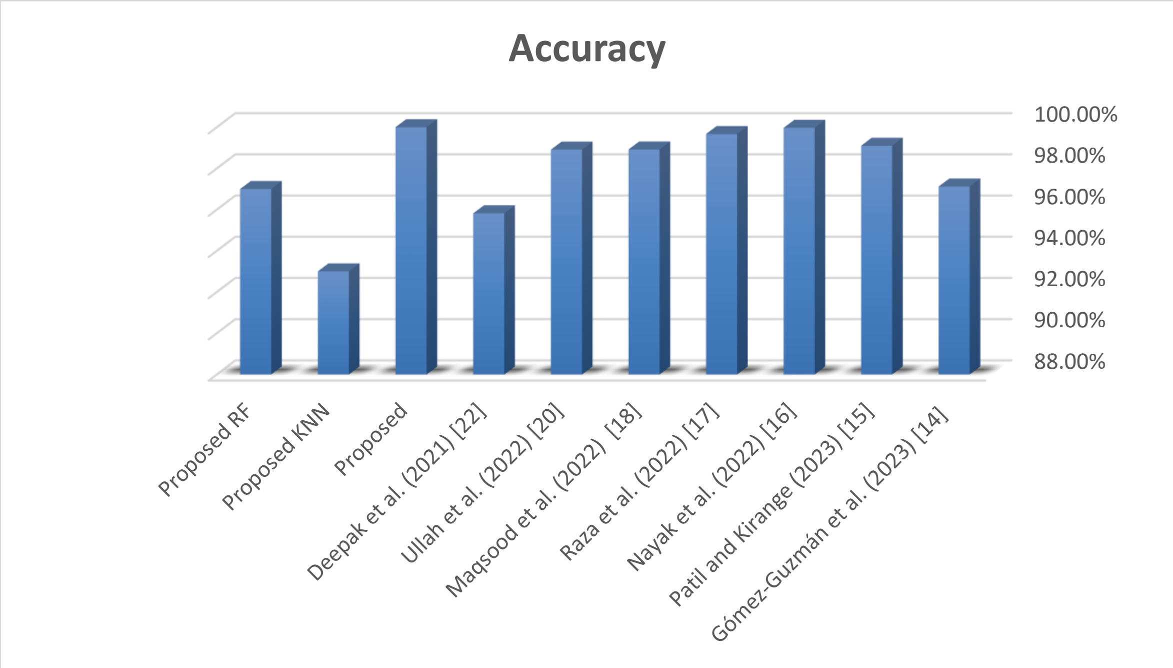


Figure (4.26): Accuracy Performance for Various Studies**.**

CHAPTER FIVE

CONCLUSIONS AND FUTUREWORKS

Chapter Five

Conclusions and Future Works

5.1 Conclusions

Through the examination of MRI imaging data, this thesis studies the use of machine learning (ML) and deep learning (DL) methods to obtain precise categorization of brain cancers. Through thorough testing and analysis, valuable insights into the usefulness of these algorithms in diagnosing brain tumors and measuring their performance using various metrics have been gained.

1. Fast Fourier Transform (FFT), K-means clustering, and Tamura were used to classify brain tumor MRI data using machine learning and deep learning algorithms. The methods collected important tumor characteristics and simplified the dataset.

2. Random Forest and K-nearest neighbors have shown promise in distinguishing tumors from non-tumors with accuracy rate of 93% and 97%. However, these methods' accuracy are 80% and 72% in classifying glioma, meningioma, and pituitary tumors varied.

3. Feature extraction and deep learning algorithms helped categorize MRI brain tumor data. The CNN model excelled at binary classification and anomaly detection, demonstrating its ability to accurately distinguish tumors with accuracy rate of 100%.

5.2. The Future Works

While the results from the established brain tumor detection and classification system are encouraging, there are still many areas that might benefit from further study and development.

1. By combining the results of numerous models, classification accuracy can be increased through the use of ensemble methods like bagging and boosting. Our brain tumor classification system should benefit from the use of ensemble approaches, which have demonstrated encouraging results in a variety of domains.

2. The system's ability to generalize can be improved by increasing the size and variety of the training dataset with photos of brain tumors. With access to additional information, models can pick up on more nuanced trends, leading to more precise classifications of abnormalities.

3. More work needs to be done on the system to make it possible to use it in real time and to integrate it with other clinical systems. This would allow doctors to more effectively incorporate the technology into their routines, ultimately leading to more precise diagnoses of brain tumors.

This thesis finishes with a thorough examination of the possibilities of ML and DL algorithms for detecting and classifying brain tumors using MRI data. These findings proved the feasibility of accurate and rapid brain tumor classification utilizing a variety of models and feature extraction methodologies. We can increase the system's performance and contribute to improvements in brain tumor identification and therapy by following the aforementioned future research routes.

References

[1] F. Zulfiqar, U. Ijaz Bajwa, and Y. Mehmood, “Multi-class classification of brain tumor types from MR images using EfficientNets,” *Biomed Signal Process Control*, vol. 84, 2023, doi: 10.1016/j.bspc.2023.104777.

[2] S. Banerjee, S. Mitra, F. Masulli, and S. Rovetta, “Brain tumor detection and classification from multi-sequence MRI: study using convnets,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-11723-8_17.

[3] M. H. G. Raouf, A. Fallah, and S. Rashidi, “Use of Discrete Cosine-based Stockwell Transform in the Binary Classification of Magnetic Resonance Images of Brain Tumor,” Institute of Electrical and Electronics Engineers (IEEE), Mar. 2023, pp. 293–298. doi: 10.1109/icbme57741.2022.10052875.

[4] R. Mehrotra, M. A. Ansari, R. Agrawal, and R. S. Anand, “A Transfer Learning approach for AI-based classification of brain tumors,” *Machine Learning with Applications*, vol. 2, 2020, doi: 10.1016/j.mlwa.2020.100003.

[5] T. Shelatkar, D. Urvashi, M. Shorfuzzaman, A. Alsufyani, and K. Lakshmanna, “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.

[6] A. H. Khan *et al.*, “Intelligent Model for Brain Tumor Identification Using Deep Learning,” *Applied Computational Intelligence and Soft Computing*, vol. 2022, 2022, doi: 10.1155/2022/8104054.

[7] Z. Jia and D. Chen, “Brain Tumor Identification and Classification of MRI images using deep learning techniques,” *IEEE Access*, 2020, doi: 10.1109/access.2020.3016319.

[8] M. Angulakshmi and M. Deepa, “A Review on Deep Learning Architecture

and Methods for MRI Brain Tumour Segmentation," *Current Medical Imaging Formerly Current Medical Imaging Reviews*, vol. 17, no. 6, 2021, doi: 10.2174/1573405616666210108122048.

[9] H. Chellakh, A. Moussaoui, A. Attia, and Z. Akhtar, "MRI Brain Tumor Identification and Classification Using Deep Learning Techniques," *Ingénierie des systèmes d information*, vol. 28, no. 1, 2023, doi: 10.18280/isi.280102.

[10] K. Pranitha, N. Vurukonda, and R. K. Nayak, "A Comprehensive Survey on MRI Images Classification for Brain Tumor Identification using Deep Learning Techniques," in *3rd International Conference on Smart Electronics and Communication, ICOSEC 2022 - Proceedings*, 2022. doi: 10.1109/ICOSEC54921.2022.9951913.

[11] G. Karayegen and M. F. Aksahin, "Brain tumor prediction on MR images with semantic segmentation by using deep learning network and 3D imaging of tumor region," *Biomed Signal Process Control*, vol. 66, 2021, doi: 10.1016/j.bspc.2021.102458.

[12] M. Ali, S. O. Gilani, A. Waris, K. Zafar, and M. Jamil, "Brain Tumour Image Segmentation Using Deep Networks," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3018160.

[13] G. Karayegen and M. F. Aksahin, "Brain Tumor Prediction with Deep Learning and Tumor Volume Calculation," in *TIPTEKNO 2021 - Tip Teknolojileri Kongresi - 2021 Medical Technologies Congress*, 2021. doi: 10.1109/TIPTEKNO53239.2021.9632861.

[14] M. A. Gómez-Guzmán *et al.*, "Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks," *Electronics (Switzerland)*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12040955.

[15] S. Patil and D. Kirange, "Ensemble of Deep Learning Models for Brain Tumor Detection," *Procedia Comput Sci*, vol. 218, 2023, doi: 10.1016/j.procs.2023.01.222.

[16] 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.

[17] A. Raza *et al.*, "A Hybrid Deep Learning-Based Approach for Brain Tumor Classification," *Electronics (Switzerland)*, vol. 11, no. 7, 2022, doi: 10.3390/electronics11071146.

[18] S. Maqsood, R. Damaševičius, and R. Maskeliūnas, "Multi-Modal Brain Tumor Detection Using Deep Neural Network and Multiclass SVM," *Medicina (Lithuania)*, vol. 58, no. 8, 2022, doi: 10.3390/medicina58081090.

[19] N. Ullah, M. S. Khan, J. A. Khan, A. Choi, and M. S. Anwar, "A Robust End-to-End Deep Learning-Based Approach for Effective and Reliable BTD Using MR Images," *Sensors*, vol. 22, no. 19, 2022, doi: 10.3390/s22197575.

[20] N. Ullah *et al.*, "An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning," *Applied Sciences (Switzerland)*, vol. 12, no. 11, 2022, doi: 10.3390/app12115645.

[21] E. A. Abbood and T. A. Al-Assadi, "GLCMs Based multi-inputs 1D CNN Deep Learning Neural Network for COVID-19 Texture Feature Extraction and Classification," *Karbala International Journal of Modern Science*, vol. 8, no. 1, 2022, doi: 10.33640/2405-609X.3201.

[22] S. Deepak and P. M. Ameer, "Automated Categorization of Brain Tumor from MRI Using CNN features and SVM," *J Ambient Intell Humaniz Comput*, vol. 12, no. 8, pp. 8357–8369, Aug. 2021, doi: 10.1007/s12652-020-02568-w.

[23] D. Moitra and R. Kr. Mandal, "Classification of non-small cell lung cancer using one-dimensional convolutional neural network," *Expert Syst Appl*, vol. 159, 2020, doi: 10.1016/j.eswa.2020.113564.

[24] J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical imaging using machine learning and deep learning algorithms: A review," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies, iCoMET 2019*, 2019. doi: 10.1109/ICOMET.2019.8673502.

[25] M. G. Hart, R. Romero-Garcia, S. J. Price, and J. Suckling, "Global Effects of Focal Brain Tumors on Functional Complexity and Network Robustness: A

Prospective Cohort Study," *Clinical Neurosurgery*, vol. 84, no. 6. 2019. doi: 10.1093/neuros/nyy378.

[26] J. S. Miser, B. C. Shia, Y. W. Kao, Y. L. Liu, S. Y. Chen, and W. L. Ho, "The Health Care Utilization and Medical Costs in Long-Term Follow-Up of Children Diagnosed With Leukemia, Solid Tumor, or Brain Tumor: Population-Based Study Using the National Health Insurance Claims Data," *JMIR Public Health Surveill*, vol. 9, 2023, doi: 10.2196/42350.

[27] P. A. T. E. Board, "Adult Central Nervous System Tumors Treatment (PDQ®)," *PDQ Cancer Information Summaries*, 2022.

[28] L. Rogers *et al.*, "Meningiomas: Knowledge base, treatment outcomes, and uncertainties. A RANO review," *Journal of Neurosurgery*, vol. 122, no. 1. 2015. doi: 10.3171/2014.7.JNS131644.

[29] F. Petralia *et al.*, "Integrated Proteogenomic Characterization across Major Histological Types of Pediatric Brain Cancer," *Cell*, vol. 183, no. 7, 2020, doi: 10.1016/j.cell.2020.10.044.

[30] D. N. Louis *et al.*, *World Health Organization Classification of Tumours WHO Classification of Tumours of the Central Nervous System*, vol. 131, no. 6. 2016.

[31] D. N. Louis *et al.*, *The 2016 WHO classification of tumors of the central nervous system*, vol. 131, no. 6. 2016.

[32] Dr. T. V. kumar, "CLASSIFICATION OF BRAIN CANCER TYPE USING MACHINE LEARNING," *Journal of Artificial Intelligence and Capsule Networks*, vol. 2019, no. 2, 2019, doi: 10.36548/jaicn.2019.2.006.

[33] U. K. Malviya, "Tumor Detection in MRI Images using Modified Multi-level Otsu Thresholding (MLOT) and Cross-Correlation of Principle Components," in *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, 2020. doi: 10.1109/ICCMC48092.2020.ICCMC-00026.

[34] N. B. Bahadure, A. K. Ray, and H. P. Thethi, "Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired

BWT and SVM,” *Int J Biomed Imaging*, vol. 2017, 2017, doi: 10.1155/2017/9749108.

[35] P. Jyotiyana and S. Maheshwari, “Maximal Stable Extremal Region Extraction of MRI Tumor Images Using Successive Otsu Algorithm,” in *Lecture Notes in Networks and Systems*, 2019. doi: 10.1007/978-981-13-0586-3_67.

[36] X. Zhang, H. Tian, Z. Wang, and S. Nie, “Segmentation of brain glioma on MRI using multiple densely connected 2D-CNNs,” *Guangxue Jishu/Optical Technique*, vol. 46, no. 5, 2020.

[37] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim, and A. Islam, “TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3179577.

[38] S. A. Hantoush Alrubaie and A. H. Hameed, “Dynamic Weights Equations for Converting Grayscale Image to RGB Image,” *JOURNAL OF UNIVERSITY OF BABYLON for Pure and Applied Sciences*, vol. 26, no. 8, 2018, doi: 10.29196/jubpas.v26i8.1677.

[39] H. Mutahira, B. Ahmad, M. S. Muhammad, and D. R. Shin, “Focus Measurement in Color Space for Shape from Focus Systems,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3098753.

[40] K. Kumar, R. K. Mishra, and D. Nandan, “Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC,” in *Procedia Computer Science*, 2020. doi: 10.1016/j.procs.2020.04.215.

[41] M. Ahmad *et al.*, “Pliable Algorithm RGB image Convert in Gray image Using Transformation Equation,” 2019. [Online]. Available: www.IJCSSE.org

[42] A. Güneş, H. Kalkan, and E. Durmuş, “Optimizing the color-to-grayscale conversion for image classification,” *Signal Image Video Process*, vol. 10, no. 5, 2016, doi: 10.1007/s11760-015-0828-7.

[43] P. Parida and R. Rout, “Transition region-based approach for skin lesion

segmentation," *Electronic Letters on Computer Vision and Image Analysis*, vol. 19, no. 3, pp. 28–39, 2020, doi: 10.5565/rev/elcvia.1177.

[44] P. Mahajan, P. Abrol, and P. K. Lehana, "Effect of blurring on identification of aerial images using convolution neural networks," in *Lecture Notes in Electrical Engineering*, Springer, 2020, pp. 469–484. doi: 10.1007/978-3-030-29407-6_34.

[45] T. G. Devi, N. Patil, S. Rai, and C. S. Philipose, "Gaussian Blurring Technique for Detecting and Classifying Acute Lymphoblastic Leukemia Cancer Cells from Microscopic Biopsy Images," *Life*, vol. 13, no. 2, Feb. 2023, doi: 10.3390/life13020348.

[46] S. H. Gangolli, A. Johnson Luke Fonseca, and R. Sonkusare, "Image Enhancement using Various Histogram Equalization Techniques," in *2019 Global Conference for Advancement in Technology, GCAT 2019*, Institute of Electrical and Electronics Engineers Inc., Oct. 2019. doi: 10.1109/GCAT47503.2019.8978413.

[47] Rao, B. S. (2020, April 1). *Dynamic Histogram Equalization for contrast enhancement for digital images*. Applied Soft Computing; Elsevier BV. https://doi.org/10.1016/j.asoc.2020.106114

[48] I. W. Adiyasa, A. P. Prasetyono, A. Yudianto, P. P. W. Begawan, and D. Sultantyo, "Bilinear interpolation method on 8x8 pixel thermal camera for temperature instrument of combustion engine," in *Journal of Physics: Conference Series*, 2020. doi: 10.1088/1742-6596/1700/1/012076.

[49] D. Khaledyan, A. Amirany, K. Jafari, M. H. Moaiyeri, A. Z. Khuzani, and N. Mashhadi, "Low-Cost Implementation of Bilinear and Bicubic Image Interpolation for Real-Time Image Super-Resolution," in *2020 IEEE Global Humanitarian Technology Conference, GHTC 2020*, 2020. doi: 10.1109/GHTC46280.2020.9342625.

[50] O. Rukundo, "Normalized Weighting Schemes for Image Interpolation Algorithms," *Applied Sciences (Switzerland)*, vol. 13, no. 3, Feb. 2023, doi:

10.3390/app13031741.

[51] U. R. Acharya *et al.*, "Automated Detection of Alzheimer's Disease Using Brain MRI Images– A Study with Various Feature Extraction Techniques," *J Med Syst*, vol. 43, no. 9, Sep. 2019, doi: 10.1007/s10916-019-1428-9.

[52] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, "Feature Extraction Methods: A Review," in *Journal of Physics: Conference Series*, 2020. doi: 10.1088/1742-6596/1591/1/012028.

[53] M. A. A. Hamid and N. A. Khan, "Investigation and Classification of MRI Brain Tumors Using Feature Extraction Technique," *J Med Biol Eng*, vol. 40, no. 2, pp. 307–317, Apr. 2020, doi: 10.1007/s40846-020-00510-1.

[54] G. Kumar and P. K. Bhatia, "A detailed review of feature extraction in image processing systems," in *International Conference on Advanced Computing and Communication Technologies, ACCT*, Institute of Electrical and Electronics Engineers Inc., 2014, pp. 5–12. doi: 10.1109/ACCT.2014.74.

[55] V. Nair, M. Chatterjee, N. Tavakoli, A. S. Namin, and C. Snoeyink, "Optimizing CNN using Fast Fourier Transformation for Object Recognition," in *Proceedings - 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020*, 2020. doi: 10.1109/ICMLA51294.2020.00046.

[56] V. S. Tutatchikov, "Two-dimensional fast Fourier transform: Batterfly in analog of Cooley-Tukey algorithm," in *Proceedings - 2016 11th International Forum on Strategic Technology, IFOST 2016*, 2017. doi: 10.1109/IFOST.2016.7884163.

[57] P. N. Tackie Ammah and E. Owusu, "Robust medical image compression based on wavelet transform and vector quantization," *Inform Med Unlocked*, vol. 15, 2019, doi: 10.1016/j.imu.2019.100183.

[58] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?," *Pattern Recognit*, vol. 93, 2019, doi: 10.1016/j.patcog.2019.04.014.

[59] A. Bakshi *et al.*, "Performance Augmentation of Cuckoo Search Optimization

Technique Using Vector Quantization in Image Compression," *Mathematics*, vol. 11, no. 10, May 2023, doi: 10.3390/math11102364.

[60] L. Na, X. Zhiyong, D. Tianqi, and R. Kai, "Automated brain tumor segmentation from multimodal MRI data based on Tamura texture feature and an ensemble SVM classifier," *International Journal of Intelligent Computing and Cybernetics*, vol. 12, no. 4, pp. 466–480, Oct. 2019, doi: 10.1108/IJICC-04-2019-0031.

[61] B. Imran, "Content-Based Image Retrieval Based on Texture and Color Combinations Using Tamura Texture Features and Gabor Texture Methods," *American Journal of Neural Networks and Applications*, vol. 5, no. 1, p. 23, 2019, doi: 10.11648/j.ajnna.20190501.14.

[62] N. Li and Z. Xiong, "Automated brain tumor segmentation from multi-modality MRI data based on Tamura texture feature and SVM model," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2019. doi: 10.1088/1742-6596/1168/3/032068.

[63] D. Painuli, S. Bhardwaj, and U. köse, "Recent advancement in cancer diagnosis using machine learning and deep learning techniques: A comprehensive review," *Computers in Biology and Medicine*, vol. 146. 2022. doi: 10.1016/j.compbiomed.2022.105580.

[64] J. Manhas, R. K. Gupta, and P. P. Roy, "A Review on Automated Cancer Detection in Medical Images using Machine Learning and Deep Learning based Computational Techniques: Challenges and Opportunities," *Archives of Computational Methods in Engineering*, vol. 29, no. 5. 2022. doi: 10.1007/s11831-021-09676-6.

[65] A. Aldahiri, B. Alrashed, and W. Hussain, "Trends in Using IoT with Machine Learning in Health Prediction System," *Forecasting*, vol. 3, no. 1, 2021, doi: 10.3390/forecast3010012.

[66] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans Cybern*, vol. 44, no. 12, 2014, doi: 10.1109/TCYB.2014.2307349.

[67] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Comput Sci*, vol. 2, p. 160, 2021, doi: 10.1007/s42979-021-00592-x.

[68] E. Antoniou *et al.*, “EEG-based eye movement recognition using the brain–computer interface and random forests,” *Sensors*, vol. 21, no. 7, Apr. 2021, doi: 10.3390/s21072339.

[69] J. Y. Kim *et al.*, “Development of random forest algorithm based prediction model of alzheimer’s disease using neurodegeneration pattern,” *Psychiatry Investig*, vol. 18, no. 1, pp. 69–79, 2021, doi: 10.30773/pi.2020.0304.

[70] A. Singh, R. Kumar, and A. K. Tiwari, “Prediction of Alzheimer’s Using Random Forest with Radiomic Features,” *Computer Systems Science and Engineering*, vol. 45, no. 1, pp. 513–530, 2023, doi: 10.32604/csse.2023.029608.

[71] A. S. M. Shafi, M. M. I. Molla, J. J. Jui, and M. M. Rahman, “Detection of colon cancer based on microarray dataset using machine learning as a feature selection and classification techniques,” *SN Appl Sci*, vol. 2, no. 7, 2020, doi: 10.1007/s42452-020-3051-2.

[72] A. Cutler, D. R. Cutler, and J. R. Stevens, “Random Forests,” in *Ensemble Machine Learning*, New York, NY: Springer New York, 2012, pp. 157–175. doi: 10.1007/978-1-4419-9326-7_5.

[73] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, “Efficient kNN classification with different numbers of nearest neighbors,” *IEEE Trans Neural Netw Learn Syst*, vol. 29, no. 5, 2018, doi: 10.1109/TNNLS.2017.2673241.

[74] W. Xing and Y. Bei, “Medical Health Big Data Classification Based on KNN Classification Algorithm,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2955754.

[75] G. Alimjan, T. Sun, Y. Liang, H. Jumahun, and Y. Guan, “A New Technique for Remote Sensing Image Classification Based on Combinatorial Algorithm of SVM and KNN,” *Intern J Pattern Recognit Artif Intell*, vol. 32, no. 7, pp. 1–23, 2018, doi: 10.1142/S0218001418590127.

[76] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, “Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction,” *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-10358-x.

[77] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning-Based Text Classification,” *ACM Computing Surveys*, vol. 54, no. 3. 2021. doi: 10.1145/3439726.

[78] Y. Zhao, Q. Chen, W. Cao, W. Jiang, and G. Gui, “Deep Learning Based Couple-like Cooperative Computing Method for IoT-based Intelligent Surveillance Systems,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2019. doi: 10.1109/PIMRC.2019.8904229.

[79] F. J. Díaz-Pernas, M. Martínez-Zarzuela, D. González-Ortega, and M. Antón-Rodríguez, “A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network,” *Healthcare (Switzerland)*, vol. 9, no. 2, Feb. 2021, doi: 10.3390/healthcare9020153.

[80] H. Greenspan, B. Van Ginneken, and R. M. Summers, “Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5. 2016. doi: 10.1109/TMI.2016.2553401.

[81] F. Alrasheedi, X. Zhong, and P. C. Huang, “Padding Module: Learning the Padding in Deep Neural Networks,” *IEEE Access*, vol. 11, pp. 7348–7357, 2023, doi: 10.1109/ACCESS.2023.3238315.

[82] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech Syst Signal Process*, vol. 151, 2021, doi: 10.1016/j.ymssp.2020.107398.

[83] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving Deep Convolutional Neural Networks for Image Classification,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, 2020, doi: 10.1109/TEVC.2019.2916183.

[84] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif Intell Rev*, vol. 53, no. 8, 2020, doi: 10.1007/s10462-020-09825-6.

[85] R. A. Pratiwi, S. Nurmaini, D. P. Rini, M. N. Rachmatullah, and A. Darmawahyuni, “Deep ensemble learning for skin lesions classification with convolutional neural network,” *IAES International Journal of Artificial Intelligence*, vol. 10, no. 3, 2021, doi: 10.11591/ijai.v10.i3.pp563-570.

[86] V. B. Codreanu, S. Bv, S. Aigner, V. Weinberg, D. Podareanu, and G. Caspar Van Leeuwen, “Best Practice Guide - Deep Learning,” 2019, doi: 10.13140/RG.2.2.31564.05769.

[87] V. H. Phung and E. J. Rhee, “A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Applied Sciences (Switzerland)*, vol. 9, no. 21, 2019, doi: 10.3390/app9214500.

[88] A. Saxena, “An Introduction to Convolutional Neural Networks,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, 2022, doi: 10.22214/ijraset.2022.47789.

[89] M. Woźniak, J. Siłka, and M. Wieczorek, “Deep neural network correlation learning mechanism for CT brain tumor detection,” *Neural Comput Appl*, vol. 35, no. 20, 2023, doi: 10.1007/s00521-021-05841-x.

[90] R. D. Rakshit, D. R. Kisku, P. Gupta, and J. K. Sing, “Cross-resolution face identification using deep-convolutional neural network,” *Multimed Tools Appl*, vol. 80, no. 14, 2021, doi: 10.1007/s11042-021-10745-y.

[91] C. Versloot, “What are max pooling, average pooling, global max pooling and global average pooling, ,” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2022.

[92] R. L. Kumar, J. Kakarla, B. V. Isunuri, and M. Singh, “Multi-class brain tumor classification using residual network and global average pooling,” *Multimed Tools Appl*, vol. 80, no. 9, 2021, doi: 10.1007/s11042-020-10335-4.

[93] A. Kost, W. A. Altabey, M. Noori, and T. Awad, “Applying neural networks for tire pressure monitoring systems,” *SDHM Structural Durability and*

Health Monitoring, vol. 13, no. 3, 2019, doi: 10.32604/sdhm.2019.07025.

[94] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138. Elsevier Ltd, pp. 14–32, Jun. 01, 2021. doi: 10.1016/j.neunet.2021.01.026.

[95] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences (Switzerland)*, vol. 10, no. 5, Mar. 2020, doi: 10.3390/app10051897.

[96] K. Vijayaprabakaran and K. Sathiyamurthy, "Towards activation function search for long short-term model network: A differential evolution based approach," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2637–2650, Jun. 2022, doi: 10.1016/j.jksuci.2020.04.015.

[97] D. Yi, J. Ahn, and S. Ji, "An effective optimization method for machine learning based on ADAM," *Applied Sciences (Switzerland)*, vol. 10, no. 3, 2020, doi: 10.3390/app10031073.

[98] W. E. L. Ilboudo, T. Kobayashi, and K. Sugimoto, "Robust Stochastic Gradient Descent with Student-t Distribution Based First-Order Momentum," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 3, 2022, doi: 10.1109/TNNLS.2020.3041755.

[99] R. Tiwari, "Stabilizing the training of deep neural networks using Adam optimization and gradient clipping," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 07, no. 01, 2023, doi: 10.55041/ijsrem17594.

[100] G. Memarzadeh and F. Keynia, "Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm," *Electric Power Systems Research*, vol. 192, 2021, doi: 10.1016/j.epsr.2020.106995.

[101] T. El Idriss, A. Idri, I. Abnane, and Z. Bakkoury, "Predicting blood glucose using an LSTM neural network," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*,

Institute of Electrical and Electronics Engineers Inc., Sep. 2019, pp. 35–41. doi: 10.15439/2019F159.

[102] F. Aksan, Y. Li, V. Suresh, and P. Janik, "CNN-LSTM vs. LSTM-CNN to Predict Power Flow Direction: A Case Study of the High-Voltage Subnet of Northeast Germany," *Sensors*, vol. 23, no. 2, 2023, doi: 10.3390/s23020901.

[103] H. Li, J. Han, S. Li, H. Wang, H. Xiang, and X. Wang, "Abnormal Driving Behavior Recognition Method Based on Smart Phone Sensor and CNN-LSTM," *International Journal of Science and Engineering Applications*, vol. 11, no. 1, 2022, doi: 10.7753/ijsea1101.1001.

[104] A. Bibi *et al.*, "Spam Mail Scanning Using Machine Learning Algorithm," *J Comput (Taipei)*, vol. 73, no. 2, 2020, doi: 10.17706/jcp.15.2.73-84.

[105] H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "A Framework for Predicting Airfare Prices Using Machine Learning," , Volume 22, Issue 3, 2022.

[106] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Appl Sci*, vol. 1, no. 12, 2019, doi: 10.1007/s42452-019-1356-9.

Appendix A

GUI for Brain Tumor Diagnosis and Classification

To make the brain tumor classification approach more user-friendly and accessible, a Graphical User Interface (GUI) was developed. Doctors can easily enter patient data, upload MRI images, and obtain reliable classification results thanks to the graphical user interface (GUI). The user interface has defined areas for entering critical medical data, allowing for full documentation and efficient record administration. As illustrated in Figure (A.1), first name, last name, age, place of birth, father's name, and occupation are all included. Collecting this information assists medical personnel in appropriately identifying and classifying patients, as well as allowing them to preserve complete patient records for future use.

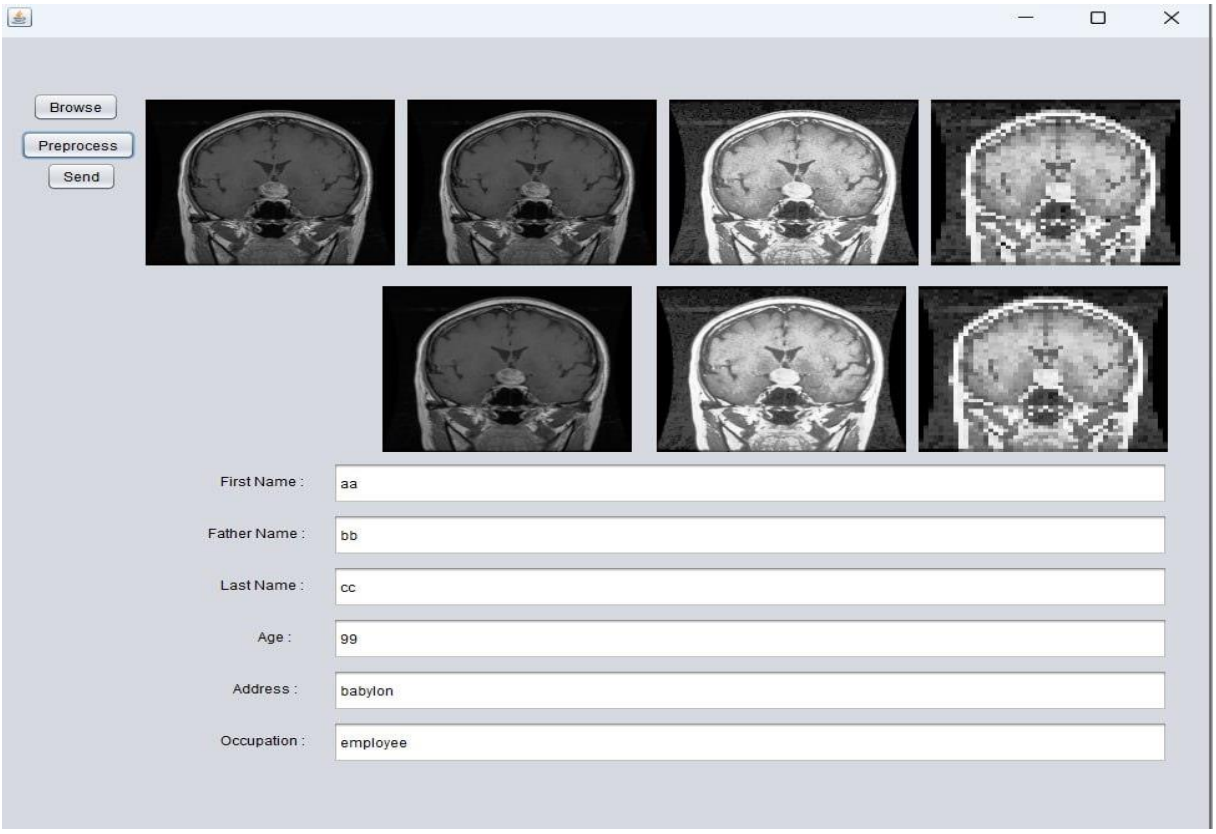


Figure (A.1): Graphical User Interface.

The graphical user interface allows you to import brain MRI scans. After being selected, the image is automatically routed through the categorization process. The image is preprocessed to ensure consistency and conformity with the learned models by converting it to grayscale, performing histogram equalization, adding Gaussian blur, and scaling it to a standard format (for example, 50x50 pixels). After the pretreatment stages are completed, the GUI initiates the categorization procedure. As demonstrated in Figure (A.2), the Convolutional Neural Network (CNN) model is utilized to categorize images into two groups. Based on the image data, the models determine whether or not a brain tumor exists.

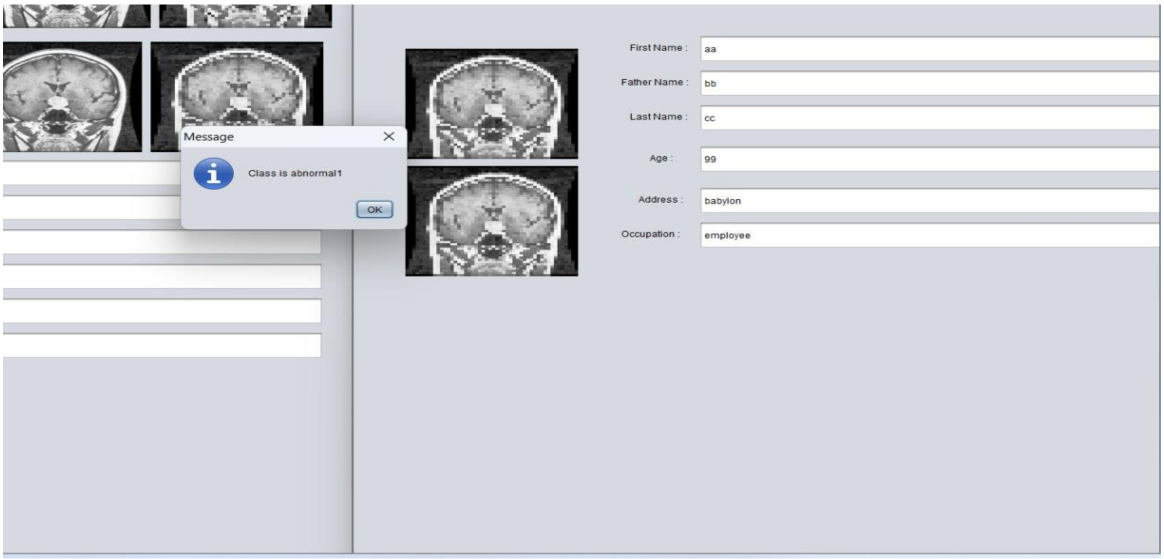


Figure (A.2): Graphical User Interface Binary Classification Procedure.

The GUI use the CNN model to detect the many types of anomalies linked with brain tumors. The model examines the image and categorizes it as one of numerous anomalies. After classification, the GUI displays the results in an intelligible manner. This includes presenting the patient's information, the uploaded MRI image, and the categorization findings. For binary classification, the GUI indicates whether the image has been classified as normal or abnormal.

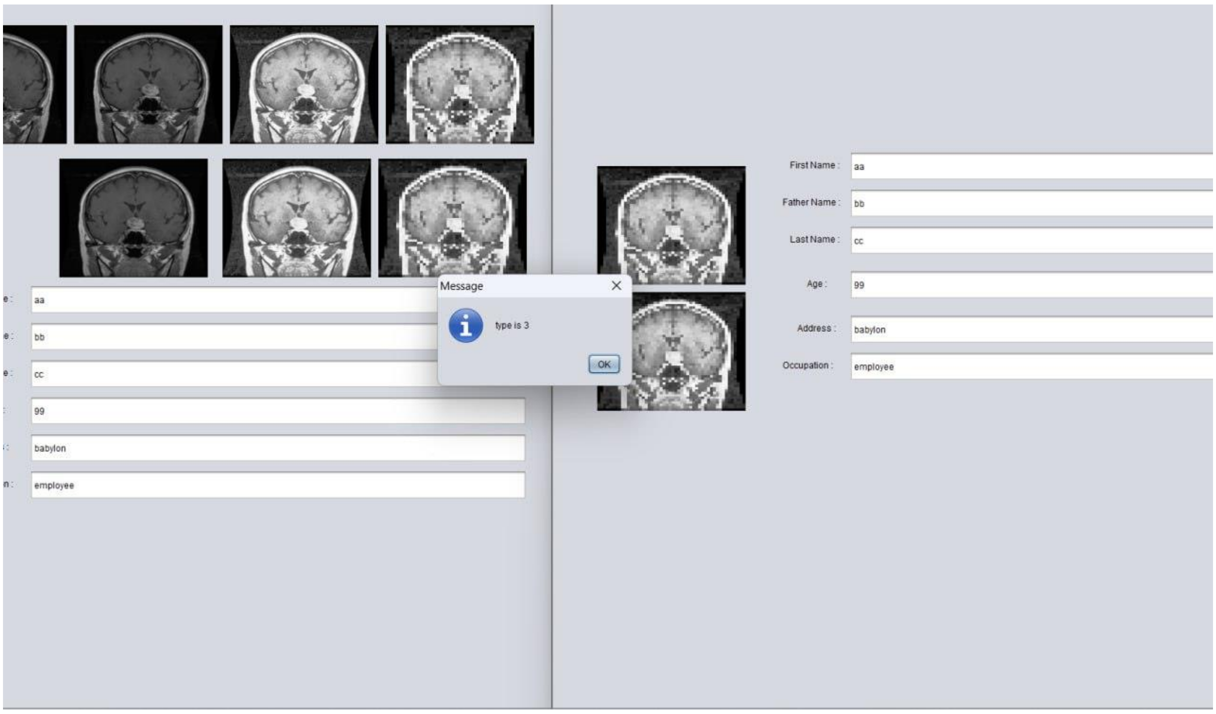


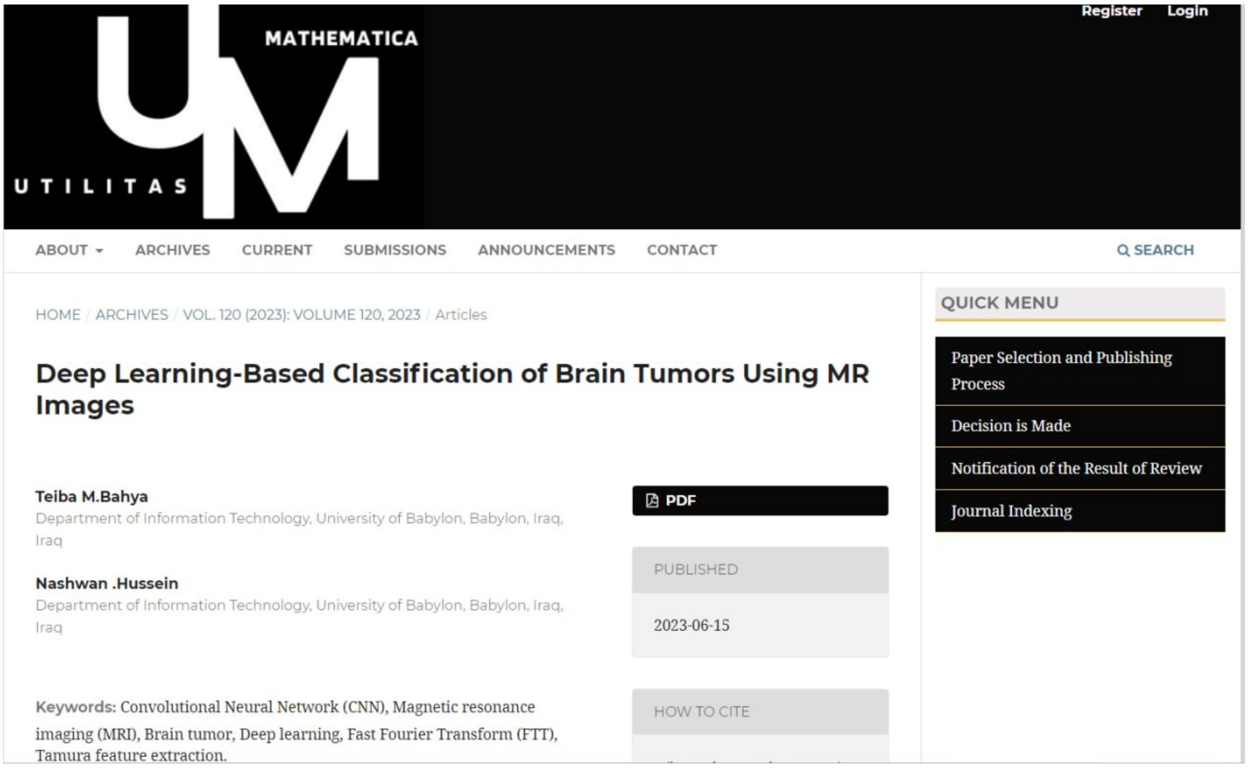
Figure (A.3): Graphical User Interface Abnormality Classification Procedure.

The GUI displays the specific type of abnormality found in the brain tumor image. This information assists clinicians in better interpreting classification results, which improves the quality of care they can provide.

The graphical user interface also allows the results to be exported for use in other medical systems or electronic health records. The GUI streamlines the workflow of the brain tumor classification system by merging patient information entry, image upload, classification, and results presentation. It increases diagnosis speed and accuracy, assists in making well-informed treatment decisions, and ultimately benefits patients.

Appendix B

The Published Paper



Register Login

MATHEMATICA

UTILITAS

ABOUT ARCHIVES CURRENT SUBMISSIONS ANNOUNCEMENTS CONTACT SEARCH

HOME / ARCHIVES / VOL. 120 (2023): VOLUME 120, 2023 / Articles

Deep Learning-Based Classification of Brain Tumors Using MR Images

Teiba M.Bahya
Department of Information Technology, University of Babylon, Babylon, Iraq, Iraq

Nashwan .Hussein
Department of Information Technology, University of Babylon, Babylon, Iraq, Iraq

PDF

PUBLISHED
2023-06-15

HOW TO CITE

Keywords: Convolutional Neural Network (CNN), Magnetic resonance imaging (MRI), Brain tumor, Deep learning, Fast Fourier Transform (FTT), Tamura feature extraction.

QUICK MENU

- Paper Selection and Publishing Process
- Decision is Made
- Notification of the Result of Review
- Journal Indexing

Appendix C

The Accepted Paper

The Islamic University

33
04/May/2023

LETTER OF ACCEPTANCE

(Machine Learning Techniques to Classify Brain Tumor)

Teba Bhya and Nashwan Hussein

It has been accepted for presentation in the **"The Sixth International Iraqi Conference on Engineering Technology and its Applications (6th IICETA 2023)"**. The final decision of publication in IEEE explore is subject terms and conditions of Conference Scientific Committee and IEEE.

Sincerely,

Dr Ahmed Alkhayyat

Assist. Prof. Dr. Ahmed Alkhayyat
Chairman of the 6th IICETA 2023

Prof. Dr. Sattar B. Sadkhan
Representative of IEEE/ Iraq

The Sixth International Iraqi Conference on Engineering Technology and its Applications (6th IICETA 2023)

جمهورية العراق

Ministry Of Health Babylon Health Directorate Email:- Babel_Healthmoh@yahoo.com Tel:282628 or 282621		وزارة الصحة والبيئة دائرة صحة محافظة بابل المدير العام مركز التدريب والتنمية البشرية لجنة البحوث

استمارة رقم: ٢٠٢٣/٠٢٧

رقم القرار :

تاريخ القرار :

وزارة الصحة
دائرة صحة بابل
مركز التدريب والتنمية البشرية
لجنة البحوث

قرار لجنة البحوث

تحية طيبة ...

درست لجنة البحوث في دائرة صحة بابل مشروع البحث ذي الرقم (٢٠٢٣/٠٢٧/ بابل) المعنون (تصنيف أورام الدماغ باستخدام لغة الاله) والمقدم من الباحثة (طبيه مصطفى كاظم) إلى وحدة أدارة البحوث والمعرفي مركز التدريب والتنمية البشرية في دائرة صحة بابل بتاريخ ٢٠٢٣/٢/٢٠ وقررت :

قبول مشروع البحث أعلاه كونه مستوفيا للمعايير المعتمدة في وزارة الصحة والخاصة بتنفيذ البحوث ولا مانع من تنفيذه في مؤسسات الدائرة .

مع الاحترام

الدكتور
محمد عبد الله عجرش
رئيس لجنة البحوث
/ / ٢٠٢٣

وزارة الصحة
دائرة صحة بابل
مركز التدريب والتنمية البشرية
لجنة البحوث

نسخة منه إلى :

- مكتب المدير العام / مركز التدريب والتنمية البشرية / وحدة إدارة البحوث ... مع الأوليات.

المستخلص

يجب تحديد وتصنيف الأورام الدماغية بدقة للكشف المبكر والعلاج الفعّال. تفسير صور التصوير بالرنين المغناطيسي (MRI) يدويًا معقد ومعرض للأخطاء. تجمع طريقة استخراج السمات الجديدة بين الشبكات العصبية التصاعدية والشبكات القصيرة الأجل (LSTM) لمعالجة هذه المشكلات.

تستخدم هذه الرسالة أساليب تعلم الآلة لتصنيف الأورام الدماغية في صور MRI. يتم استخدام Tomura، الذي يصف النسيج والهيكل استنادًا إلى الخشونة والتباين والاتجاه والخشونة، لاستخراج السمات ذات الصلة من صور MRI. تمتزج هذه السمات مع تحويل فورييه سريع وتجميع K-means لإنشاء مجموعة سمات ذات 54 بعدًا. تلتقط هذه الميزات أنماط الأورام الدماغية المميزة للتحليل والتصنيف.

أولاً، تم تصميم نماذج تعلم الآلة مثل Random Forest (RF) و K-Nearest Neighbors (KNN) للتصنيف الثنائي (طبيعي مقابل غير طبيعي) وأيضًا تصنيف الشذوذ (أنواع مختلفة من الشذوذ). أظهرت النتائج أن كلاً من نماذج Random Forest و KNN حققت دقة عالية بنسبة 97٪ و 93٪ في التمييز بين الصور الطبيعية والمغايرة للأورام الدماغية. علاوة على ذلك، أداءت هذه النماذج بفعالية في تحديد فئات معينة من الشذوذ بدقة تبلغ 80٪ و 72٪ على التوالي.

ثم، تقوم الرسالة بدراسة فعالية شبكات النسيج التصاعدي (CNN) في تصنيف الأورام الدماغية. حققت الشبكة CNN نتائج استثنائية ليس فقط في التصنيف الثنائي ولكن أيضًا في مجال مهام تصنيف الشذوذ. لمعالجة الطبيعة التسلسلية للبيانات، تم تحسين النموذج من خلال دمج شبكة LSTM. تستخدم هذه الهندسة المعمارية LSTM خلايا الذاكرة والبوابات لالتقاط التبعيات طويلة المدى والأنماط الزمنية في بيانات الأورام الدماغية. تتبع الطبقات المتصلة بالكامل وطبقات الإسقاط لزيادة دقة التنبؤ والحد من التجاوز.

لختصار، تقدم هذه الدراسة منهجًا شاملاً لتصنيف الأورام الدماغية في صور MRI. يشمل النهج استخدام طرق استخراج السمات وشبكات CNN وشبكات LSTM. من خلال تنفيذ هذه التقنيات المتقدمة للتعلم العميق، تحقق الدراسة دقة 100٪ في تحديد وتصنيف الأورام..

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
قسم برمجيات تقنية المعلومات

كشف وتصنيف ورم الدماغ باستخدام التعلم الآلي

أطروحة
مقدم إلى مجلس كلية تقنية المعلومات للدراسات العليا بجامعة بابل في استيفاء جزئي لمتطلبات درجة الماجستير في تكنولوجيا المعلومات / البرمجيات

اعداد الطالبة
طيبة مصطفى كاظم جاسم

اشراف
أ. م. د. نشوان جاسم حسين

1444 هـ **2023 م**