Comparative Analysis of Brain Tumor Detection and Classification using Machine Learning Algorithms

By

Ihsan Mamraiz
ECI-IT-20-167

Muhammad Israr
ECI-IT-20-053

Zakir Hussain
ECI-IT-20-130

2025

Department of Computing

Faculty of Engineering Sciences and Technology
Hamdard University Islamabad Campus, Pakistan

Comparative Analysis of Brain Tumor Detection and Classification using Machine Learning Algorithms

By

Ihsan Mamraiz
ECI-IT-20-167

Muhammad Israr
ECI-IT-20-053

Zakir Hussain
ECI-IT-20-130

Under the supervision of
Dr. Muzamil Malik

2025

Faculty of Engineering Sciences and Technology

Hamdard University Islamabad Campus, Pakistan

Comparative Analysis of Brain Tumor Detection and Classification using Machine Learning Algorithms

By

Ihsan Mamraiz
ECI-IT-20-167

Muhammad Israr
ECI-IT-20-053

Zakir Hussain
ECI-IT-20-130

A project presented to the

Faculty of Engineering Sciences and Technology

In partial fulfillment of the

requirements for the degree of

Bachelor of Science
In
Artificial Intelligence

Faculty of Engineering Sciences and Technology

Hamdard University Islamabad Campus, Pakistan

Faculty of Engineering Sciences and Technology
Hamdard University Islamabad Campus, Pakistan

CERTIFICATE

This project "Comparative Analysis of Brain Tumor Detection and Classification using Machine Learning Algorithms" presented by **Ihsan Mamraiz, Muhammad Israr** and **Zakir Hussain** under the direction of **Dr. Muzamil Malik** approved by the project examination committee, has been presented to and accepted by the Faculty of Engineering and Science Technology, in partial fulfillment of the requirements for Bachelor of Artificial Intelligence.

Dr. Muzamil Malik (Supervisor)	Dr. Tahir Saleem Khattak (Examiner 1)
Dr. Tahir Saleem Khattak (Incharge, Department of Computing)	Dr. Shaheer Muhammad (Examiner 2)
Dr. Hassan Raza (Associate Dean, FES)	

ABSTRACT

Brain tumors are among the most critical and life-threatening conditions, with successful treatment outcomes heavily dependent on early and accurate detection. Traditional diagnostic methods, including magnetic resonance imaging (MRI) and biopsy, often identify tumors at advanced stages, complicating treatment. In this project, we propose a machine learning-based system for the early detection of brain tumors using MRI images. Our approach utilizes supervised machine learning techniques, focusing primarily on deep learning models such as Convolutional Neural Networks (CNNs), to classify and detect brain tumors with high precision. The system was trained using the publicly available Brain Tumor MRI images dataset on Kaggle, which includes annotated MRI scans. We implemented and evaluated few CNN architectures: Alex Net, VGG-19, and ML algorithms like Random Forest, Logistics regression. Each model was fine-tuned to enhance its performance in detecting brain tumors. CNN, VGG-19and Alex Net achieved the highest accuracy of 99% on the test dataset. To ensure Best-fit input quality, image preprocessing techniques such as resizing, normalization, and grayscale conversion were applied. The models were evaluated using precision, recall, F1-score, sensitivity. The results demonstrate the efficacy of deep learning models in improving the detection and classification of brain tumors, offering a non-invasive and efficient diagnostic tool for clinical applications.

DEDICATION

This thesis is dedicated to our families, whose unwavering love, support, and encouragement have been the cornerstone of our success and the driving force behind our achievements. To our parents, who have been our greatest source of strength and guidance, we owe an immeasurable debt of gratitude. Their sacrifices, patience, and belief in our abilities have provided us with the confidence and determination to overcome every challenge and reach this significant milestone. Their unconditional love and wise counsel have been our guiding light through the ups and downs of this journey. To our siblings, who have always stood beside us as our cheerleaders and confidants, your unwavering support and shared moments of joy have brought balance and positivity to our lives. Your words of encouragement and steadfast belief in our potential have been a source of motivation that kept us going even during the most demanding times. To our extended families, whose prayers, kind words, and unwavering faith in us have been a constant source of comfort and strength, we are deeply grateful. Your encouragement has not only inspired us but also reminded us of the importance of perseverance and dedication. This work is a testament to the collective love, care, and support that we have received from all of you. It is our sincere hope that this accomplishment serves as a source of pride for each of you, as it would not have been possible without your endless encouragement and belief in us.

ACKNOWLEDGEMENT

All glory and gratitude are due to Al-Mighty "**ALLAH**", the most compassionate, the most generous, the source of all human knowledge and wisdom, who gave us the mental strength and capacity to carry out this endeavor and explore the vast ocean of knowledge. Respects are due to our most cherished Holy Prophet, "**Hazrat MUHAMMAD (Peace Be Upon Him),**" whose example will serve as a constant source of wisdom for people.
We are grateful to the Faculty of Engineering Sciences and Technology for their support of this project, which is a significant accomplishment for us as undergraduates.

We wish to express our appreciation to our supervisor **Dr. Muzamil Malik** who served as our major advisor. We would like to express our heartiest gratitude for her keen guidance, sincere help and friendly manners which inspire us to do well on the project and make it a reality. Many people, especially our classmates and team members themselves, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our project. We thank all the people for their help directly and indirectly in completing our project.

Ihsan Mamraiz

Muhammad Israr

Zakir Hussain

PROJECT IN BRIEF

Project Title	Comparative Analysis of Brain Tumor Detection and Classification using Machine Learning Algorithms
Objective	To deliver a comprehensive literature on Brain Tumor Detection and classification through magnetic resonance imaging
Undertaken By	Ihsan Mamraiz, Muhammad Israr, Zakir Hussain
Supervised By	Dr. Muzamil Malik
Date Started	September, 2023
Date Completed	September, 2024
Tools Used	Python
System Used	Laptop Core i7 10th Generation, Kaggle, Google Colab

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CAD	Computer Assisted Diagnosis
CNN	Convolutional Neural Network
DL	Deep Learning
ML	Machine Learning
AI	Artificial Intelligence
RF	Random Forest
LR	Logistic Regression
VGG	Visual Geometry Group
HGG	High Grade Glioma
LGG	Loe Grade Glioma
CDSS	Clinical Decision support System
MRI	Magnetic Resonance Imaging
WML	White Matter Lesions
FL	Federated Learning
FLAIR	Fluid Attenuated Inversion Recovery
SVM	Support Vector Machines
K-NN	K-Nearest Neighbors
ReLU	Rectified Linear Unit
GraD-CaM	Gradient-weighted Class Activation Mapping
SHAP	Shapley Additive Explanations

CHAPTER 1
INTRODUCTION

1.1 Introduction

Improved technologies, especially machine learning, have brought about a revolution in healthcare in terms of disease detection and classification [1]. As in most industries, the application of technology in healthcare, particularly machine learning, has greatly improved the ability to detect and classify diseases [2]. Some of the machines learning approaches that are critical to the understanding of intricate datasets include neural networks, decision trees, supporting vector machines, and ensemble techniques [3].

One of the most critical healthcare issues, brain tumors continue to affect millions of people around the world. They are defined as abnormal growths of skin cells that develop in the brain, and these grow into benign or malignant tumors [4]. It is important to identify the brain tumors while they are at an early stage because this will lead to a higher chance of the patient responding to treatment [5]. However, the traditional methods of diagnosing conditions which rely heavily on sophisticated, often subjective processes such as reading MRI scans are not only tedious, but also lack efficacy. This has led to the attempt of using different machine learning (ML) algorithms for automating medical diagnostics, particularly in the field of tumor detection and classification [6].

Machine learning, [7] an application of artificial intelligence, has proved highly effective in recognizing intricate patterns in vast data sets, including medical imaging. ML algorithms are capable of automatically identifying brain tumors from healthy and cancerous tissues, as well as categorizing the tumor itself. A range of ML algorithms, ranging from Logistic Regression and Random Forests to Convolutional Neural Networks (CNNs), have been utilized for this task over the years [8]. Each algorithm has its own strengths and limitations, which makes it necessary to evaluate their performance across key metrics such as accuracy, precision, recall, and computational efficiency [9].

Some of the most common varieties of cancer found are included in the list above Cross-sectional analysis can reveal the presence of different subtypes which will need an alternate approach to diagnosis and treatment. One of the leading causes of death globally is cancer. The World Health Organization

(WHO) states that timely diagnosis is essential as it could potentially save lives [10].

There are different types of tumors; some are benign, some can be premalignant, and others are classified as malignant. Unlike malignant growths, benign tumors do not invade neighboring organs and tissues. This type of tumor is most likely diagnosed with the help of MRI, or Magnetic Resonance Imaging [11].

In the past, the diagnostics would not reach completion until it was ascertained if the tumor tissue was malignant or benign. Nowadays, with the current advancement in novel technologies every day, it is necessary to create a dependable and effective medicine tool development technology for subdivision and classification of growth tumors in MRI images [12].

This has a look at goals to behavior a comparative evaluation of different ML algorithms for brain tumor detection and category. By utilizing publicly available or clinically sourced datasets, the research will systematically evaluate algorithms to identify the best method for diagnosing mind tumors. This comparative evaluation will provide precious insights for researchers and healthcare practitioners, supporting them pick out the ideal set of rules for particular use cases. Furthermore, it'll make a contribution to the continued improvement of AI-based answers in medical imaging, in the long run assisting inside the early detection and treatment of brain [13].

Categorization and Detection of brain Tumor growing a gadget is appreciably important. Given that incorrect identification gives rise to malignant effects, the classification techniques have to achieve an excessive level of accuracy whilst performing tumor classification. Excessive accuracy with ML algorithms and techniques used for detection and type is often obtained with the proper approach of set of rules on statistics. Implementation of the precise type algorithm on facts ends in the great accuracy of algorithms. Different parameters are had to classify the magnificence of tumors [14]. Those parameters are then blended as they're depending on type. Day by day sort of tumor patients are in labs the usage of MRI/CT scans and the capabilities of tumors are stored because the data of sufferers. This information can then be used for analysis and prediction of different sufferers. In technical phrases device studying may be the answer to research and expect the elegance of tumor [15].

MRI or magnetic resonance imaging provides extremely important details about the anatomy of a diagnosed brain tumor, its cells, and the blood vessels associated with it, and therefore, it can be efficiently utilized for monitoring and treating the ailment. Non-destructive MRI is a method that enables the imaging and diagnosis of multiple medical issues without physically interacting with the patient’s body. The technique employs a strong magnet, radio wave transceivers, and a computer to capture images of the soft tissues, bones, and body parts. The results can be viewed on a computer screen, transferred through a network, printed out, or written to a CD. MRI, unlike x-ray machines, does not utilize radiation. Technological advancements have increased the accuracy of MR images, and they are now used to screen for many conditions.[16].

The most Automatic detection necessitates brain imaging segmentation which is arguably the most critical and challenging task in all computer aided clinical diagnostic tools. The MRI images of the brain have noise artifacts which are multiplicative noises and their reduction is a difficult task. From a clinical point of view, processes such as noise removing should not obliterate the tiny anatomical details. This makes proper segmentation of such images highly challenging. In any case, proper segmentation of the MRI images is very important and vital for right diagnosis to be made using computer aided clinical equipment [17].

Malignant tumors is one of the most frighting of all types of tumors. In adults Gliomas and lymphomas constitute close to eighty percent of malignant tumoral cases [18]. Gliomas encompass subtypes of primary tumor which range from low-grade to heterogeneous tumors (more infiltrative malignant tumors). They possess maximum prevalence with extremely high mortality rate. They can be classified into High Grade Glioma (HGG) and Loe Grade Glioma (LGG). HGG is low infiltrative and aggressive when compared to the LGG. HGG patients typically do not live more than fourteen months following the detection process. Radiotherapy and chemotherapy are the current HGG and LGG treatments [19]. Ischemic stroke is a cerebral vascular insult that is a common cause of mortality and disability in the world. The damaged brain tissue (stroke lesion) goes through multiple stages of the evolution of the disease which are classified as sub-acute (24h-2w), chronic (>2w), and acute (0-24h) [20].

Segmentation and subsequent quantitative lesions analysis of clinical images provides valuable information for evaluating brain pathologies that are important for treatment mapping strategies, predicting and tracking disease progression, and tracking patient outcomes. In addition, specific locations of specific injury correlate to specific deficits based on the brain structure involved. Functional deficits from stroke lesions are correlated with volume of damage to specific brain areas. Furthermore, accurate delineation of pathology is perhaps the single most important step in brain tumor cases where an estimation of tumor volume and subsequent sub-components is critical in making more informed treatment strategies.

Unlike clinical images used to assess stroke deficits, segmentation of lesion regions in multidimensional images presents more weighty challenges. Heterogeneous inconsistencies may occur in the lesions including variation in frequency, shape, location and volume which all makes the steps in segmentation design less efficient. Furthermore, it is not completely trivial to rehabilitate the contusions, edema, and hemorrhages that occur in substructures of brain tumors like the necrotic core and proliferating cells. The arguably better segmentation could be achieved by using the long, tedious, and expensive process of humans delegating written explanations. And, it is completely impractical for more studies including more inter observer variation. The more effective automatic method of tumor extraction is one of primary objectives in medical images computing which produces reproducible, objective and scalable techniques for quantitative assessment of brain tumor. MS and stroke lesions appear identical hyper-intense on FLAIR and other WML sequences. It is generally difficult to obtain statistically before information regarding lesion shape and appearance [21].

Investigations around artificial intelligence (AI), and in particular developments in machine learning (ML) and deep learning (DL), led to revolutionary breakthroughs in radiology, pathology, genomics and many other fields.

Although state-of-the-art DL models can have millions of parameters to learn (from sufficiently large, high-quality curated datasets) to match clinicians in accuracy, and to ensure safety, fairness, equity and generalizability to new data2–5. For example, to train a detector of tumors using AI, we need huge databases that capture the full extent of potential variations of anatomy, pathology, and by types of possible input data. Obtaining data like this is difficult because health data can be highly sensitive and its use is tightly controlled6. Even if anonymity allow us to overcome restrictions, it is well known that simply stripping identifying information such as patient name and date of birth will unlikely keep any reasonable expectation of anonymity7. For instance, it is possible to regenerate an identifiable image of a patient face from computed tomography (CT) or magnetic resonance imaging (MRI)8. Part of the reason that data sharing is not readily found in health care is simply that gathering, curating, and maintaining a useful high-quality data set takes considerable time, effort, and money. Therefore, these sets of data can be valuable for business and hence are unlikely to be freely shared. Data collectors instead typically maintain fine-grained control over the data collected by them [22].

Federated learning (FL)9–11 is a learning paradigm seeking to address the problem of data governance and privacy by training algorithms collaboratively without exchanging the data itself. Originally developed for different domains, such as mobile and edge device use cases12, it more recently picked up momentum for health care uses13–20. FL allows insight to be obtained in collaboration, i.e., as a form of consensus model, without transferring patient data outside the firewalls of the institutions they are located in. Rather, the ML process is carried out locally at each involved institution and only model features (e.g., parameters, gradients) are exchanged as shown in Fig. 1. It has been demonstrated in recent studies that FL models can reach performance levels that are on par with centrally hosted data sets and better than models seeing only single-institutional isolation data [23].

A successful deployment of FL could therefore present enormous potential for advancing precision medicine and at the population-level in which models could have unbiased answers, optimally capture an individual's physiology, be responsive to atypical disease and consider governance as well as patient confidentiality concerns. However, FL also requires rigorous technical scrutiny to ensure that the algorithm is moving forward in the most prudent way possible, without compromising safety or patient confidentiality. Nonetheless, it could possibly overcome a variety of restrictions associated with techniques that require a single centralized repository of data. We foresee a federated future for digital health and with this vision paper, we describe our shared vision in an attempt to give context and overview for the community on the opportunities and impact of FL for medical aims (section "Data driven medicine requires federated efforts"), and to illustrate key considerations and challenges of implementing FL for digital health (section "Technical considerations") [24].

1.2 Overview of Brain and Brain Tumor

The human brain is the main part of the human nervous system. It occupies the space in the human head and is enclosed in the skull. The brain has the function to control the entire human body. It is one type of organ that allows humans to tolerate and accept all aspects of the environment. The human brain allows humans to take action and communicate the thoughts and feeling. In this section we will discuss the anatomy of the brain for helping understand the basic things. Structure of the human brain is shown in Figure 1.1

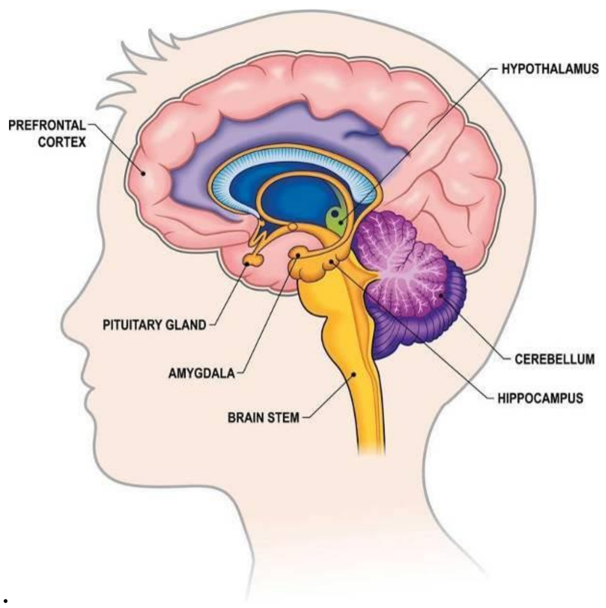


Figure 1.1: Human Brain Image.

The brain tumors are categorized into mainly two types: number one brain tumor (benign tumor) and secondary mind tumor (malignant tumor). The benign tumor is one sort of cell grows slowly inside the brain and form of brain tumor is gliomas. It originates from non-neuronal mind cells referred to as astrocytes. Essentially, number one tumors are less competitive, however these tumors have a great deal stress on the mind and because of that, the brain stops operating properly [25]. The secondary tumors are more aggressive and more likely to spread into other tissue. Secondly, mind tumor originates through different parts of the body. These types of tumors have a cancer cell in the body that is metastatic that spreads into different areas of the body like brain, lungs and so forth. Secondary mind tumor is very malignant. The origin for secondary brain tumors is chiefly due to lungs cancer, kidney cancer, bladder cancer and so forth. [26].

1.3 Magnetic Resonance Imaging (MRI)

Raymond V. Damadian discovered the first magnetic image in 1969. By 1977 we were able to visualize the first MRI image of the human body and the most ideal technique. Thanks to MRI, we are able to visualize the details of the internal structure of the brain and through that we can observe the various types of tissues of the human body. When compared to conventional imaging modalities such as X-ray and computer tomography, Magnetic Resonance Imaging (MRI) images of high quality. The only other imaging modality that compares to MRI is Positron Emission Tomography (PET). Also, MRI is an excellent method to help understand the brain tumor in the human body. Different MRI imaging techniques for mapping tumor induced Changes includes T1 weighted, T2 weighted, and FLAIR (Fluid attenuated inversion recovery) weighted images as shown in Figure 1.2.

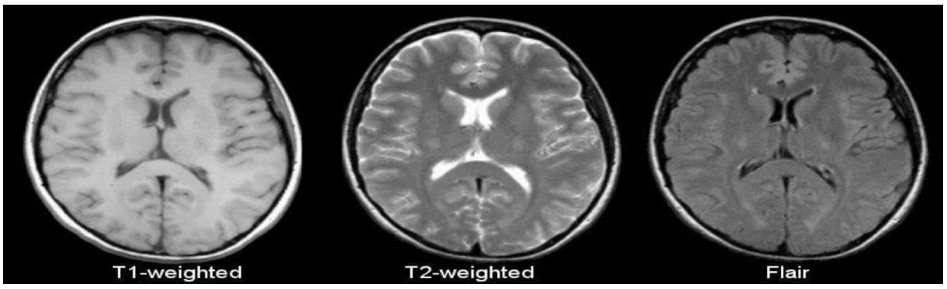


Figure 1.2: Brain MRI Images.

1.4 Aim and Objectives

Aim and objectives of this research work are discussed below.

The aim of the project is:

To identify the most effective algorithm for enhancing diagnostic accuracy, reducing false positives/negatives, and improving Algorithmic efficiency in brain tumor detection and classification.

The primary objectives of the project are:

- To select a set of popular machines learning algorithms, including traditional methods (e.g. Logistic regression, Random Forest, vgg19, Alex net) and modern techniques (e.g., CNN), for comparative analysis.
- Provide performance-based insights to guide researchers and healthcare professionals in choosing the most effective algorithms.
- To identify best-fit solution for brain tumor classification.

1.5 Scope of the Project

This project has numerous future applications in healthcare and technology. It can be used to develop diagnostic support tools for radiologists, aiding in faster and more accurate tumor detection. Early detection algorithms can be integrated into public health screening programs for high-risk populations. Research labs can use the project's framework to refine ML models or extend it to other medical imaging tasks. Clinicians could adopt it to personalize treatment plans based on tumor types, enhancing precision medicine. Additionally, this work could support telemedicine and remote diagnostics, making healthcare accessible in underserved areas. It also has potential commercial applications in HealthTech startups, integrating AI tools into hospital systems or mobile platforms. Beyond healthcare, the methodologies developed here can inspire multi-disease diagnostic tools and AI training frameworks [27].

1.6 Problem Statement

Brain tumors are among the most severe and life-threatening medical conditions, posing a significant challenge to healthcare professionals worldwide. The timely and accurate diagnosis of brain tumors is critical for effective treatment and improving patient survival rates. The major method for finding and diagnosing brain cancers is Magnetic Resonance Imaging (MRI) scans. The present procedure depends very much on manual interpretation of MRI results by radiologists and medical professionals. Limited by the number of competent experts available, this manual investigation is time-consuming, subject to human error, and leads to possible delays in therapy and diagnosis.

Machine learning (ML) and artificial intelligence (AI)'s arrival offers a chance to transform medical imaging by automatically and efficiently classifying brain tumors with great accuracy. With the ability to quickly examine huge quantities of MRI data, machine learning models can find patterns and deviations not readily discernible by the human eye. This has sparked a growing fascination with investigating and contrasting different ML techniques in order to establish their accuracy in brain tumor classification.

This study's main aim is to assess several machine learning techniques and select the best model for brain tumor classification. The research intends to answer a fundamental question: Can machine learning models deliver timely and precise classifications that outstrip those of traditional diagnostic techniques? Based on primary performance indicators including accuracy, precision, recall, F1 score, specificity, and sensitivity, the evaluation process consists of evaluating several algorithms. A thorough evaluation of all model’s advantages and constraints guarantees a grounded comparison.

The investigation will include many machine learning methods ranging from conventional classifiers like Support Vector Machines (SVM), Random Forests, and Decision Trees to deep learning technologies such Convolutional Neural Networks (CNNs) and Transfer Learning models. Every one of these methods has distinct benefits in feature extraction, pattern recognition, and classification accuracy. Particularly in medical image analysis, Convolutional Neural Networks have shown phenomenal results since they can automatically learn and extract pertinent features from MRI scans.

Data preprocessing, too, is a main factor since the accuracy of input information greatly affects model performance. To guarantee the models receive top-quality data for training and validation, MRI pictures must be preprocessed to include noise reduction, contrast enhancement, and segmentation. Additionally, to increase computational efficiency and model generalization, feature selection and dimensionality reduction methods such as Principal Component Analysis (PCA) can be used.

The data set used in this study will be split into training, checking, and testing sets to guarantee a fair and impartial assessment. To avoid overfitting and guarantee the generalizability of the models, cross-validation methods including k-fold cross-validation will be used. Hyperparameter tuning, along with programming for each algorithm, will help to maximize performance and get the best available outcomes.

The most appropriate algorithm for real-world application in hospital settings will be found by means of comparative analysis of the models. A model should have excellent precision, recall, F1 score, and accuracy since it less false negatives and false positives, hence enhancing diagnostic integrity. Furthermore, taken into account will be the interpretability and explanation complexity of the models; this will guarantee that doctors can rely on and grasp the decision-making process of artificial intelligence-driven systems.

Beyond theoretical examination, the possible influence of this study reaches to practical applications since a good machine learning model properly deployed could greatly improve the diagnostic process for brain cancers. Automated classification helps radiologists by speeding up the identification of tumor kinds, lowering diagnostic accuracy, and offering second opinions. This in turn enables early intervention and more individualized treatment planning, hence better patient results.

Though machine learning has much promise, several issues need be tackled in research. Among these are data availability, class imbalance in data sets, computing resource needs, and moral issues concerning AI-driven medical diagnosis. Training strong models depends on careful acquisition of good labeled sets and guarantee of data variety. Furthermore, critical is cooperation with medical professionals to confirm the practical application of the developed models in real life clinical situations.

The assessment of computational efficiency and scalability is another important feature of this research. Deploying machine learning models in real-world medical environments requires given the computational resources needed for training and inference. Especially in under-resourced settings, not all medical settings might be appropriate for models requiring much GPU processing power. The research will thus consider the practicality of portable models that can run well with little processing power.

In addition, the research will examine how federated learning and edge computing can improve brain tumor classification models' accessibility and security. Federated learning enables AI models to be trained on various decentralized devices without the exchange of sensitive patient data with central servers. The method increases patient privacy and compliance with data protection laws while still facilitating effective model training. Edge computing, conversely, supports real-time classification through the local processing of MRI scans on medical imaging equipment, minimizing latency and enhancing diagnostic efficiency.

In addition to performance measures, the explainability and interpretability of machine learning models are essential for establishing trust among medical practitioners. Black-box models like deep neural networks tend to be opaque in their decision-making, which makes them hard to implement in clinical settings. Methods like Grad-CAM (Gradient-weighted Class Activation Mapping) and SHAP (Shapley Additive Explanations) will be explored in this research to improve model interpretability so that radiologists can see which features affected a particular classification decision.

The paper also recognizes the ethical repercussions of AI-assisted medical diagnosis. Machine learning may improve accuracy, but it cannot substitute human knowledge. AI must be considered an aid and not a substitute for radiologists. Clearly defined guidelines need to be established regarding the integration of AI-made classifications within clinical processes to ensure final decisions remain within the purview of medical professionals. Furthermore, data bias concerns need to be taken into consideration since biased training data may cause inconsistencies in model performance across patient populations.

The ultimate vision of this research is to help ensure the integration of AI within healthcare in a safe, effective, and patient-and-clinician-beneficial manner.

Machine learning models can be integrated into CDSS if they produce higher performance, which can improve the workflow of radiologists, decrease workload, and limit diagnostic variations. This could result in earlier and more precise diagnosis, which in turn could improve patient outcomes and survival rates.

1.7 Organization of the Report:

Chapter 01: This chapter provides an introduction to the research topic, which is the importance of classifying brain tumors and the ability of MRI to assist in diagnosis. The chapter discusses brain tumors in an overview, why early detection and diagnosis is important, aims and objectives, and scope of the study. Furthermore, this chapter will highlight the problem statement and the need for machine learning-based solutions in medical imaging.

Chapter 02: This chapter provides an extensive review of the literature on brain tumor classification through machine learning. It describes the types of methodologies, previously used datasets, and algorithms, along with their advantages and disadvantages. Another section discusses deep learning developments, and CNN architectures, including comparisons of current and traditional diagnostic classifications.

Chapter 03: This section explains the method adopted in the study, discussing data collection method, image preprocessing methods, and the machine learning models selected. It describes the specific preprocessing methods(state) of converting the images to gray scale, resizing the images, and normalizing the images to exploit as much image quality as possible. The sections detailing the selected models VGG-19, AlexNet, Logistic Regression, Random Forest, and CNN, include an explication of their predicted performance expectations. The results from model evaluations are presented in paragraph form to summaries the findings.

Chapter 4: We researched and developed an effective machine-learning model for brain tumor identification and classification using MRI images. We were able to leverage advanced deep learning architectures such as VGG-19, CNN, Alex Net, Logistic Regression, and Random Forest, to provide strong performance on this difficult task. Results show that the models we proposed have an excellent performance as the CNN model performed the best overall on the evaluation metrics we used. This model was able to achieve a testing accuracy of 96.48%, indicating that it generalized well to unseen data. In addition, the CNN model outperformed other models on precision, recall, F1-score, sensitivity, and specificity, showing it was a robust and trustworthy model for brain tumor identification and classification.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

Recent advancements in Artificial Intelligence (AI) have led to improved detection and classification of brain tumors from Magnetic Resonance Imaging (MRI) and Machine Learning (ML). AI and ML-based methods, including deep learning and radiomics, have the potential to extract valuable features from MRI images in order to determine the presence and extent of brain tumors in the future [28].

Machine learning models have become key to automated brain tumor detection, reducing dependency on human interpretations of scans acquired from an MRI machine. Early investigations involved traditional methods such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees. These models were early proof-of-concept of the potential scientific process of computational methods applied to medical images but had issues with scaling and generalization [29].

For example, uses K-means and fuzzy C-means clustering for brain tumor segmentation in MRI images. Although these algorithms were successful in separating tumor areas, their dependence on hand-engineered features hindered their consistency across different datasets. Likewise, used k-NN for medical image classification with decent accuracy but struggled with high-dimensional data [30].

One major challenge is the requirement for big, diverse, and heavily annotated datasets. Such datasets are needed to train and test AI models. Acquiring such datasets is challenging owing to privacy, annotating complexities of medical images, and inconsistencies in imaging protocols used in various institutions. A further challenge lies in the AI models' generalizability. Nonetheless, some challenges persist, such as the requirement for large and highly annotated datasets and how to tackle the generalizability of AI models to new, unseen data. Deep learning models, particularly Convolutional Neural Networks (CNNs), have obtained higher performance metrics on image-based tasks but retain low interpretability as a concern [31].

In addition, research such as has centered its attention on segmentation methods to enhance brain tumor detection accuracy with high precision using K-means clustering. Likewise, proved fuzzy C-means clustering useful for MRI-based brain tumor classification [32].

Despite these advancements, enhancing model interpretability and integrating AI into clinical workflows remain pivotal for achieving widespread adoption. Future studies should emphasize the development of explainable AI models and real-world applicability in diverse clinical settings.AI-assisted brain tumor diagnosis can also help to standardize the diagnosis process, reducing variability and improving consistency across different institutions. By addressing the challenges associated with this technology, AI can revolutionize the field of radiology and improve patient care [33].

Table 2.1: Summary of Literature Review.

Reference	**Major Findings**	**Dataset**	**Accuracy**	**Algorithm Model**
Goyal & Sharma (2021)	Segmentation techniques for improved classification.	MRI images	92.5%	K-means algorithm
Selvakumar et al. (2012)	K-means, fuzzy -means for brain tumor segmentation	MRI images	87%	K-means, Fuzzy C-means
Jia & Chen (2020)	Identification and classification using deep learning techniques	Public MRI dataset	95.6%	CNN
Amin et al. (2018)	Use of deep CNNs for brain tumor detection with high precision.	MRI images	96%	Deep CNN

Soheila Saeedi, Soraya Rezai, Hamidreza Keshavarz, sheared Kalhor, Article number 16(2023)	pituitary gland tumors, in addition to healthful brains without tumors, using magnetic resonance brain snap shots to enable physicians to detect high accuracy tumors in early tiers.	MRI- images	95.63% 86% 28%	2D CNN KNN MLP
Gore & Deshpande (2020)	Comparison of deep learning methods for tumor detection.	MRI datasets	96%	Deep Learning
Bauer et al. (2013)	Survey of MRI-based studies for tumor analysis.	MRI volumes	88%	Various ML models
Lütjens et al. (2017)	Overview of deep learning in medical image analysis.	Medical images	97%	CNN
Hazra, Animesh, Sujit Kumar Gupta, *(ICECDS)*, pp. 425-430. IEEE, 2017.	the k-means clustering algorithm is used to clustering the image. The work is done by using MATLAB.	MRI images	89%	K-means algorithm
Bauer et al. (2013)	Survey of MRI-based studies for tumor analysis.	MRI- images	87.46%	Various ML models
Saeedi et al. (2023)	MRI-based classification using 2D CNN and KNN.	MRI images	95.6%	CNN, KNN

Soltani Nejad et al. (2017)	Super pixel-based brain tumor.	FLAIR MRI	89%	Random Forest
Szilagyi et al. (2015)	Automatic segmentation using fuzzy c-means.	MRI volumes	86%	Fuzzy c-means
Ramteke & Yashawant (2012)	Automatic medical image classification using k-NN.	Medical images	88%	k-NN
Mohsen et al. (2018)	Classification using deep learning neural networks.	MRI datasets	94%	CNN
Ankit Ghosh 1 and Alok Kole 2 1jadavpur university 2aliation not available October 26, 2021	A Comparative Study of Enhanced Machine Learning Algorithms for Brain Tumor Detection and Classi cation	MRI datasets	85% 84% 84%	SVM Logistic Regression KNN
Alaa Ahmed Abbood1, Qahtan Makki Shallal2, Mohammed A. Fadhel3	Automated brain tumor classification using various deep learning models: a comparative study	MRI (magnetic resonance imaging), and CT (computed tomography)	82% 86% 91% 95.%	AlexNet VGG16 GoogleNet ResNet
Samia Mushtaq1, *, Apash Roy1 and Tawseef Ahmed Teli2	A Comparative Study on Various Machine Learning Techniques for Brain Tumor Detection Using MRI	MRI and CT	92%	CNN

T. Lakshmi Prasanthi, N. Neelima	Improvement of Brain Tumor Categorization using Deep Learning: A Comprehensive Investigation.	MRI datasets	92% 92% 95% 95%	Densenet RestNet VGG16 VGG19
Ashraf m. h. taha, dr. syaiba baldish binti Arifin, Samy s. Abu Naser	Aa systematic literature review of deep and machine learning algorithms in brain tumor and meta-analysis	Distributions Of Data Set Used	80% 79.40%	KNN SVM
Soheila Saeedi, Soraya Rezayi, Hamidreza Keshavarz and Sharareh R. Niakan Kalhori1,	MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques	CT scan	96% 95%	2D CNN Convolution al auto-encoder
MD Ashif Raja, Anzar Hussain Lone, Manpreet Kaur3, Preet Kaur 4, Rajinder Singh Sodhi, Neha	Exploring Machine Learning Approaches for Predicting Brain Tumors: A Comparative Study	MRI brain imaging	97% 97% 98% 98%	Regression KNN SVM Random Forest

Comparative analysis, such as those by [34] illustrate that CNNs outperform traditional methods in tasks like tumor segmentation and classification, achieving accuracy levels exceeding 95%. In addition, ensemble methods such as Random Forests and hybrid methods involving the blending of 2D CNN and KNN have also been promising, especially in improving robustness on a wide range of datasets. issues of data diversity, model explainability, and incorporation into clinical workflow remain, which highlights the necessity for ongoing innovation and verification under real-world conditions [35]. This work not only helps in the enhancement of diagnostic consistency but also in transforming radiological practice by means of AI-supported frameworks. These have been used in brain tumor detection using different ML algorithms, ranging from the conventional classifiers like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Decision Trees, and assembling techniques like Random Forests and Gradient Boosting. Deep learning architectures, particularly Convolutional Neural Networks (CNNs), have also been investigated in recent developments because of their high performance in image-based applications [36].

Machine learning methods, and particularly Convolutional Neural Networks (CNNs), have vastly revolutionized the field of brain tumor detection and classification, signaling a major development in its trajectory. Nevertheless, issues like model interpretability and data diversity need to be further addressed through more research to continue improving AI integration and performance within clinical settings [37].

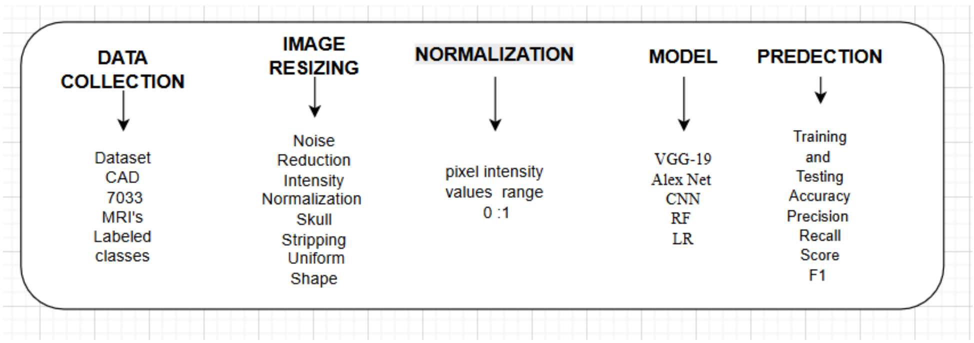
In addition, developments in multimodal imaging and hybrid ML algorithms have further enhanced the field by allowing the blending of MRI, CT scans, and clinical data for more advanced diagnostics. Research such as illustrates the application of ensemble methods in bringing together information from different sources to achieve better accuracy and reliability. These advances highlight the need to harness varied data modalities and novel algorithmic architectures to overcome existing limitations in brain tumor detection and classification. Future work must focus on creating scalable, clinically translatable models that can effectively process complex real-world situations [38].

Studies have demonstrated that ML models not only improve diagnostic accuracy but also reduce processing time and assist healthcare professionals in clinical decision-making. Traditional algorithms, such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees, laid the foundation for automated tumor detection, achieving modest success in classification tasks [39]. However, they often faced challenges with scalability and generalization. The advent of deep learning models, particularly Convolutional Neural Networks (CNNs), revolutionized this field by offering superior performance in feature extraction and classification accuracy, as evidenced by their application in segmentation and diagnostic support [40].

CHAPTER 3
METHODOLOGY

3.1 Proposed Methodology

In proposed methodology it will read the MRI image in Figure 3.1Grayscale format, then it will resize the image according to the model, normalize the image, give the model and give us the prediction.



DATA COLLECTION
Dataset CAD 7033 MRI's Labeled classes
IMAGE RESIZING
Noise Reduction Intensity Normalization Skull Stripping Uniform Shape
NORMALIZATION
pixel intensity values range 0 :1
MODEL
VGG-19 Alex Net CNN RF LR
PREDECTION
Training and Testing Accuracy Precision Recall Score F1

Figure 3.1: Proposed Model Workflow.

3.2 Data Collection

The Brain Tumor Detection Dataset is a highly curated collection of MRI images designed to facilitate research and development in brain tumor detection and classification using machine learning models. This dataset plays a crucial role in advancing computer-aided diagnostic (CAD) systems, particularly in the medical imaging domainFigure3.2[41]. Its detailed annotations and diversity of images provide a strong foundation for building reliable and robust deep learning models for early brain tumor diagnosis.

The dataset contains 7033 MRI images, each labeled with bounding boxes for four different classes of brain tumors: Glioma, Meningioma, No Tumor, and Pituitary. The images were thoroughly handpicked, labeled, and validated to maintain data quality and integrity. The annotations were done by a specialized team using the Labeling tool, which ensures high accuracy in detecting and classifying brain tumors. A multi-class organization facilitates effective discrimination among brain tumor types, making the dataset relevant for developing CAD systems for supporting clinical workflow. The MRI images are displayed in different views, such as sagittal, axial, and coronal planes, allowing for complete insight into the anatomy of the brain [42]. The diversity improves the robustness of the dataset, allowing models to generalize more effectively across different imaging directions.

Analysis was conducted by specialists so that the high quality and reliability required for training machine learning algorithms were ensured. The data set is divided into training and validation sets in order to achieve a balanced case distribution by tumor type, maximizing the model's capability for tumor detection and correct classification. Through the use of this dataset, developers and researchers can develop and test machine learning models that greatly enhance early brain tumor detection, eventually enabling timely and effective medical interventions.

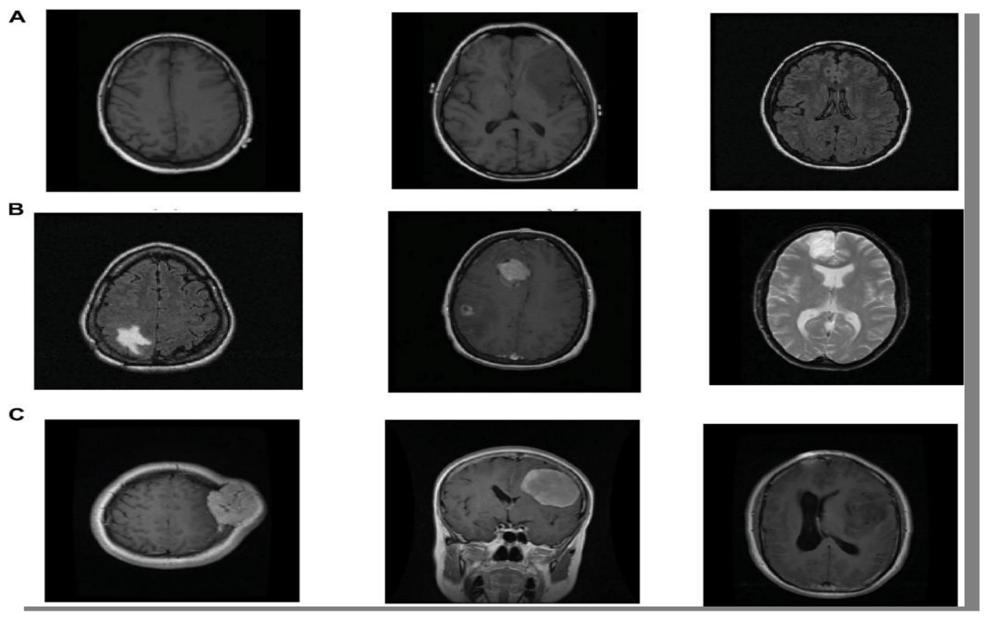


Figure 3.2: Annotations on Images.

3.3 Image Preprocessing

Brain tumor detection image preprocessing normalizes MRI images by reducing noise, normalizing intensity, and stripping the skull. These methods improve the visibility of tumors and enhance the performance of machine learning algorithms such as CNNs and K-means clustering. Proper preprocessing allows for an unbiased comparison of various classification algorithms.

3.3.1 Converting to Grayscale

After the MRI scans have been loaded, the initial vital preprocessing step is converting them to grayscale. Grayscale images are extremely powerful in medical imaging, particularly for applications such as brain tumor segmentation. This is due to the fact that grayscale images are optimal at emphasizing vital diagnostic features, such as tissue density differences, necessary for spotting abnormalities like tumors.

In RGB images, data is spread over three channels, red, green, and blue. While grayscale images combine the data into one intensity channel. This not only makes the data simpler but also makes computation easier. Even with the lower dimensionality, grayscale images maintain the critical structural and anatomical information of brain tissues. This is a crucial step, as the difference between various brain tissue types and possible tumors is mainly recorded through intensity variations instead of color. This enables the machine learning algorithms to concentrate on detecting the slight variations in tissue density, which are the most important signs of tumors.

3.3.2 Image Resizing

MRI scans exist in a broad variety of resolutions and dimensions as a result of the varying imaging equipment and acquisition parameters applied in their capture. To ensure the data is normalized to the same shape for equal analysis, the images have to be resized into a uniform form prior to being inputted into the deep learning models. This is a critical step in pre-processing medical images since models trained on inputs of different sizes can fail to perform well because of the inconsistency in the data [42].

By resizing the images to a standard size, we provide uniform input data that the neural networks can handle better. Image size standardization ensures that every image is represented in a similar way, and the task of the deep learning model to identify patterns and features throughout the dataset becomes more efficient and accurate [40]. This also serves to minimize the memory usage of the images, thereby making it more manageable to process larger datasets, which is quite important in medical imaging applications.

3.3.3 Image Normalization

Normalization is another important preprocessing step that maintains consistency between all input images. In medical imaging, intensity levels of pixels can differ considerably between various scans based on differences in Figure 3.3 MRI machine settings, lighting setups, or patient anatomy. Without normalization, such intensity differences could result in subpar model performance since the neural network would have a hard time generalizing patterns across varying inputs.

Image normalization functions by rescaling pixel intensity values to a standardized range, for example, between 0 and 1, or scaling them relative to the mean and standard deviation of the data. This makes it easier to lower the effect of extreme pixel values or noise in the images, By normalizing the intensity values, we ensure that all the images have an equal range of pixel intensities so that the model can easily concentrate on meaningful differences pertaining to brain tumors instead of extraneous variations in image quality.

All these preprocessing operations gray scaling, resizing, and normalization are crucial in the process of preparing MRI images for deep learning models, which improves the quality and consistency of the input data. This eventually enhances the performance of models in brain tumor detection and classification, resulting in more accurate and dependable diagnostic outcomes.

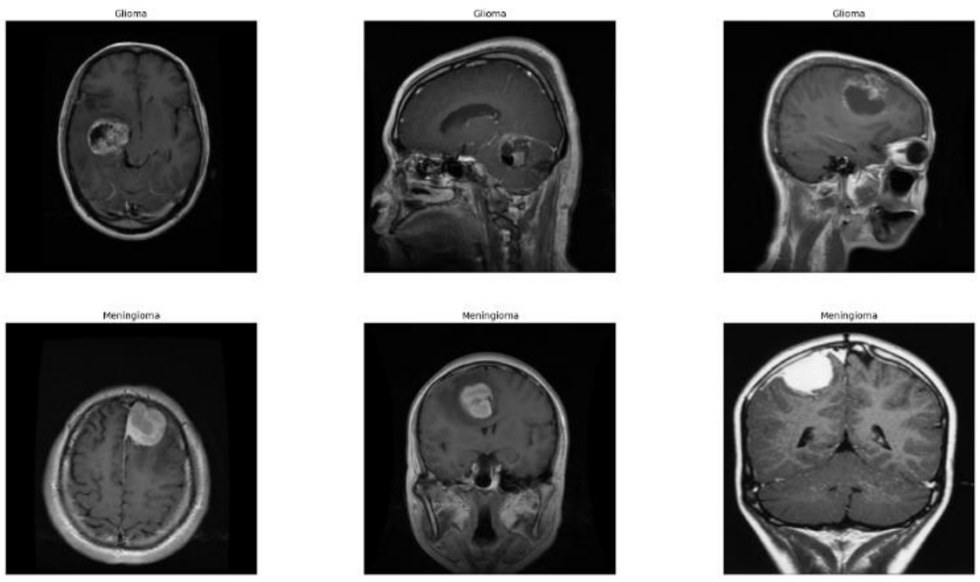


Figure 3.3: After Image Preprocessing.

3.4 Selected Models

Proposed Models are based on CNN (Convolutional Neural Network), and ML models Random Forest, Logistic regression the following are:

3.4.1 VGG-19

In Figure 3.4: Developed by the University of Oxford's Visual Geometry Group (VGG), VGG-19 is a deep convolutional neural network. It has 19 layers total: 5 max-pooling layers, 3 fully linked layers, and 16 convolutional layers with 3x3 filters. The network is easy to create because it has consistent architecture and employs modest filters to collect minute features. While preserving crucial information, the max-pooling layers aid in reducing the spatial dimensions of feature maps.

Because of its depth and ability to extract precise features, VGG-19—which is well-known for its excellent accuracy on picture classification tasks—performs well on benchmarks like ImageNet. However, it requires a lot of memory and processing power due to its deep architecture and many parameters. Using previously trained models for a variety of specialized tasks is a common application of transfer learning.

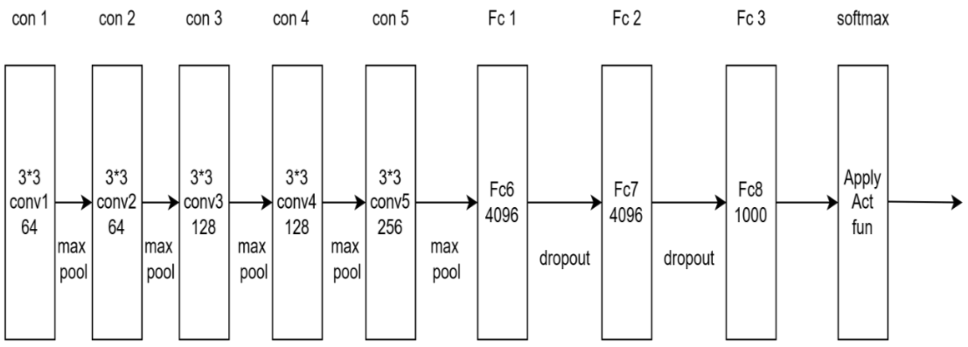


Figure 3.4: VGG-19 Architecture.

3.4.2 AlexNet

Alex Net is a pioneering deep studying version added in 2012 with the aid of Alex Hrushevsky, Ilya Stuever, and Geoffrey Hinton. It played a key function in revolutionizing the sector of laptop vision by demonstrating the energy of convolutional neural networks (CNNs) on large-scale picture class obligations. The architecture consists of 5 convolutional layers, followed with the aid of 3 absolutely connected layers. In Figure 3.5the convolutional layers use ReLU activation to introduce non-linearity, and max-pooling layers to down pattern the feature maps, decreasing spatial dimensions and computation. Alex

Net additionally employs dropout in its completely linked layers to save you overfitting, which changed into a great improvement over previous version.

One of the key breakthroughs of Alex Net was its use of GPUs for training on the ImageNet dataset, enabling the model to handle millions of images and parameters efficiently. The model's use of data augmentation techniques like image translations and reflections further enhanced its generalization. Alex Net significantly outperformed previous methods on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), achieving a top-5 error rate of 15.3% compared to the previous best of 26.2%. This success sparked widespread interest in deep learning and CNNs, leading to rapid advancements in the field of artificial intelligence.

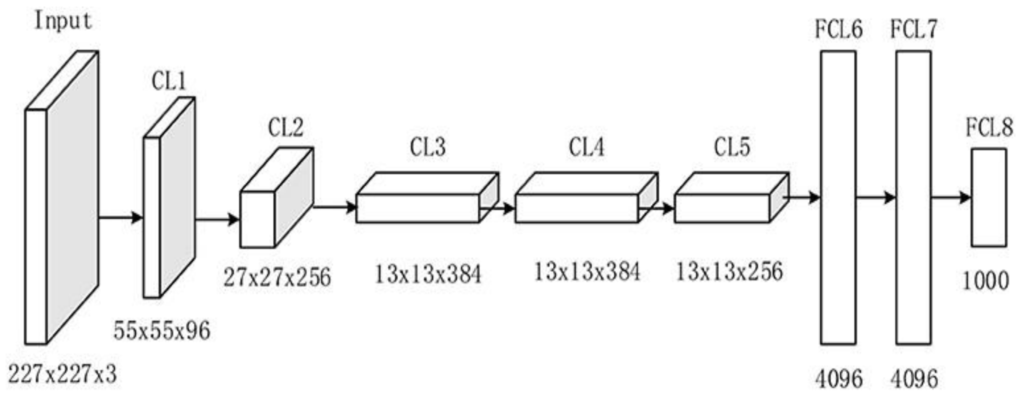


Figure 3.5: AlexNet Architecture.

3.4.3 Convolution Neural Network (CNN)

Deep learning Convolutional Neural Networks Convolutional Neural Networks A CNN, or a Convolutional Neural Network, is a type of deep learning model that processes and analyzes data with grid-like topology. In Figure 3.6 the best images are suited to be handled by the CNN on account of automatically learning spatial hierarchies in images through a series of convolutional layers, pooling layers, and fully connected layers. In convolutional layers, filters slide across the input image to detect edges, textures, and increasingly complex patterns as we go down the layers. These features are then condensed in pooling layers to retain only the most relevant information, thus reducing computation and helping prevent overfitting. Finally, fully connected layers integrate these features to classify or make predictions. This ability to learn spatial features and hierarchies makes CNNs suitable for applications such as object recognition, facial recognition, and most importantly, medical image analysis.

Advanced techniques on CNNs have been exhibited and developed as very effective tools for the identification and classification of tumors in MRI images. Traditionally, the classification of tumors is a time-consuming process that has always involved the work of radiologists. With MRI datasets, CNNs detect differences in subtle contrasts and textures that characterize different types of tumors, ranging from gliomas, meningiomas, to pituitary tumors and no tumor. Data Preprocessing: Normalization and/or resizing and even augmentation is used to increase model accuracy and adapt CNNs to the particular demands of medical images. Transfer learning is often applied in brain tumor detection: CNNs pre-trained on large image data sets, for example, ImageNet, are fine-tuned on MRI images in order to increase model precision and reduce the extensive labeled medical data required. Once trained, an CNN can classify MRI images at high sensitivity and specificity, making it useful as an adjunct to radiologists. This also reduces the time to reach a diagnosis and likely decreases the chance of human error: a significant benefit in itself, not to mention the possibility of aiding diagnostic accuracy.

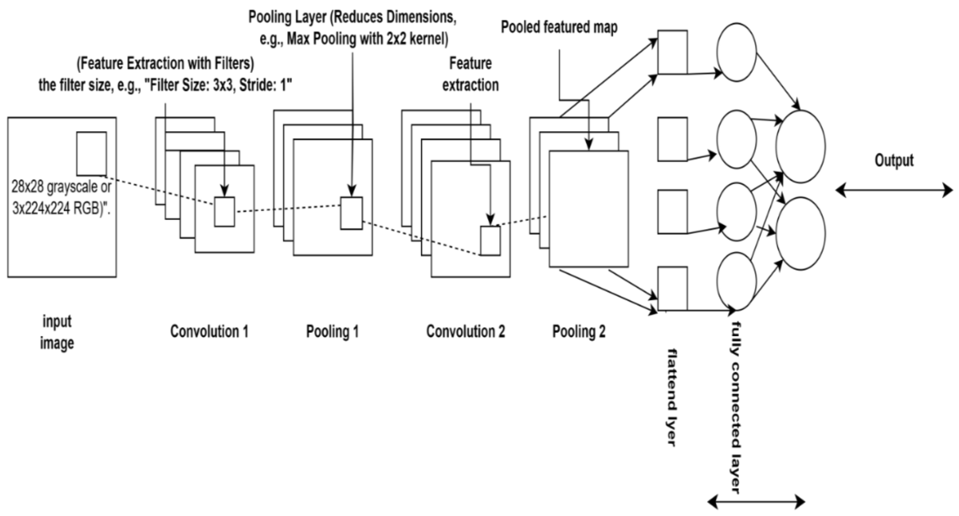

Pooling Layer (Reduces Dimensions, e.g., Max Pooling with 2x2 kernel)
Pooled featured map
(Feature Extraction with Filters) the filter size, e.g., "Filter Size: 3x3, Stride: 1"
Feature extraction
28x28 grayscale or 3x224x224 RGB)".
Output
input image
Convolution 1
Pooling 1
Convolution 2
Pooling 2
flattend lyer
fully connected layer

Figure 3.6: CNN Architecture.

3.4.4 Random Forest

Random Forest is an ensemble method based on decision trees as the low-level learner to improve prediction accuracy and robustness. It is very effective on a classification or regression task when the data gets complicated or noisy features. Each decision tree in a Random Forest is learned over a random subset of the data with a random subset of features, which is called "bagging." Then each individual tree generates its own predictions, and the final classification will be done according to a majority vote from all the trees. This reduces sometimes the overfitting that occurs with individual decision trees and improves the generalizability of the model. Additional random forest features are providing feature importance, so it is easy to identify which feature contributes the most to the classification outcome. Such fields include medical research or, in this case, brain tumor classification, wherein random forests will classify MRI images or the features extracted into classes such as benign vs. malignant or different tumor types, like glioma, meningioma, and pituitary, no tumor. Since Random Forests operate on structured data, MRI scans are normally preprocessed to come up with attributes like intensity, texture, and shape features. These would then be fed to the model that learns complex patterns and interplay connected with each kind of tumor. The noisy and unbalanced nature of the medical datasets, which was something against which the Random Forests robustly functioned, in fact was one of the reasons they were applied to brain tumor classification. Its feature importance ranking not only enlightens the researcher and clinician about which MRI characteristics are more telling of certain tumor types but also may even assist in early diagnosis and treatment planning. Although Random Forest is not as specialized on image data as deep learning models, it's good enough at accuracy and interpretability for use in the preliminary phases. Model is represented in Figure 3.7.

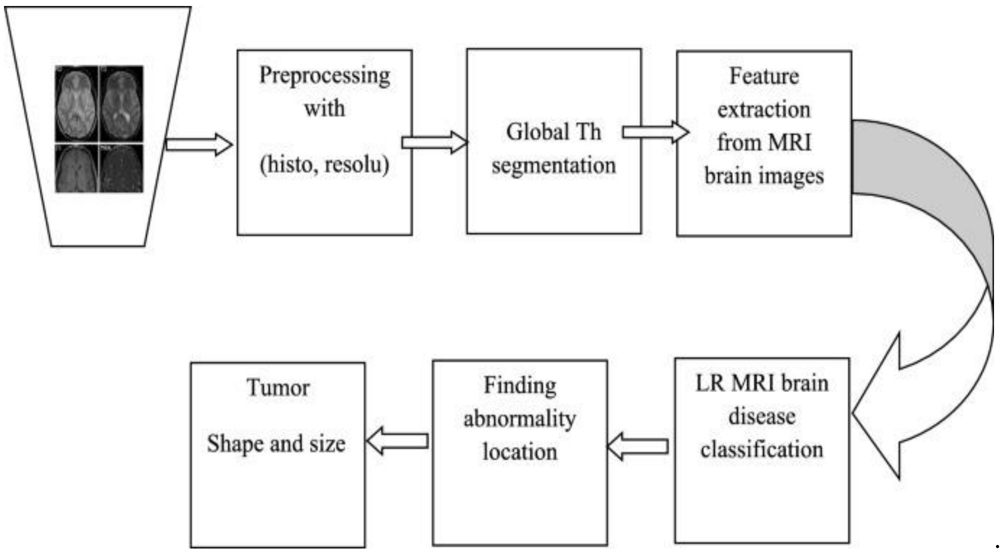


Figure 3.7: Random Forest Architecture.

3.4.5 Logistic Regression

Logistic Regression is a statistical model that has been applied to many generally useful classification tasks, either binary or multiclass. It can predict the probability that a given input belongs to a particular class by modeling the relationship of input features and the output probability using a logistic, or sigmoid, function. The weighted sum of the input features is computed and passed through the sigmoid function to map this sum to a value between 0 and 1, which is a probability. The model then makes a call and classifies the input into one of the classes determined by the target based on a threshold value-often 0.5. Logistic Regression makes various predictions of continuous values. The regression model is widely applied for classification type problems, it's straightforward to interpret, and is quite common in many fields, especially for use in medical diagnosis. Statistical model is represented in Figure 3.8.

Since Logistic Regression can classify MRI images or extracted features into classes like "tumor" vs "no tumor," the algorithm can be applied to classify different varieties of tumors, such as glioma, meningioma, and pituitary no tumor, among others. In brain tumor classification using Logistic Regression, feature extraction is needed because the algorithm prefers structured numerical data to raw images. Good features could be input into a model as texture, intensity, or shape, extracted from MRI scans. It is rather simple enough and, therefore, computationally efficient. It is also possible to interpret it: it can give a quantitative estimation of how much every feature contributes to the decision behind the classification. However, since classification of brain tumors is complicated, it is expected that Logistic Regression will perform worse than more advanced methods such as Convolutional Neural Networks (CNNs). However, it's still an important model, especially to work with when treating as a baseline model or as weighting scheme in a combined model to analyze and interpret specific features related to the tumor.

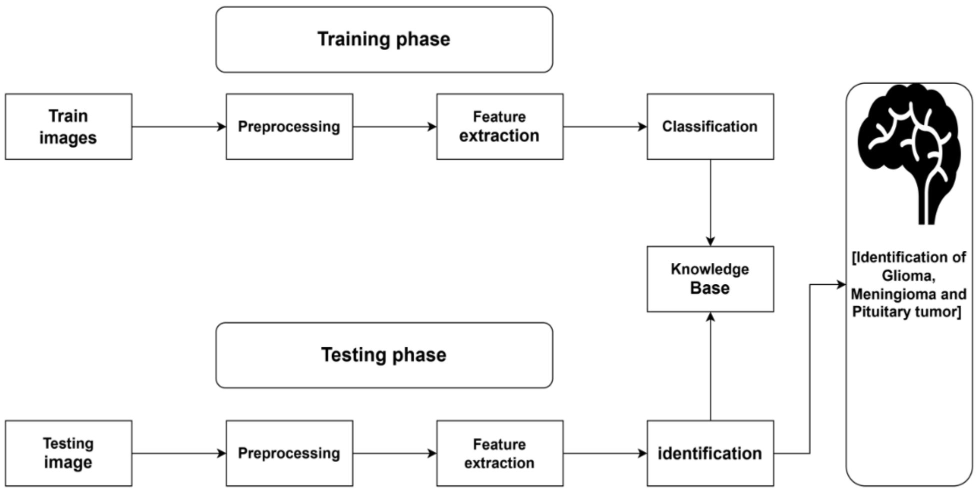


Figure 3.8: Logistic Regression Architecture.

3.5 Results

The 5 proposed models training and testing accuracy, recall score, precision score, f1 score, sensitivity and specificity.

Table 2.2: Results of the models.

Sr. No	Model	Training Accuracy	Testing Accuracy	Precision Score	Recall Score	F1 Score
1	VGG-19	98.52%	93.75%	99.36%	93.85%	93.75%
2	CNN	99.47%	96.48%	97.70%	99.33%	98.51%
3	Alex Net	99.03%	96.87%	96.89%	96.87%	96.87%
4	Random Forest	97.00%	91.00%	94.00%	99.00%	96.00%
5	Logistic Regression	66.00%	56.47%	78.10%	78.01%	78.01%

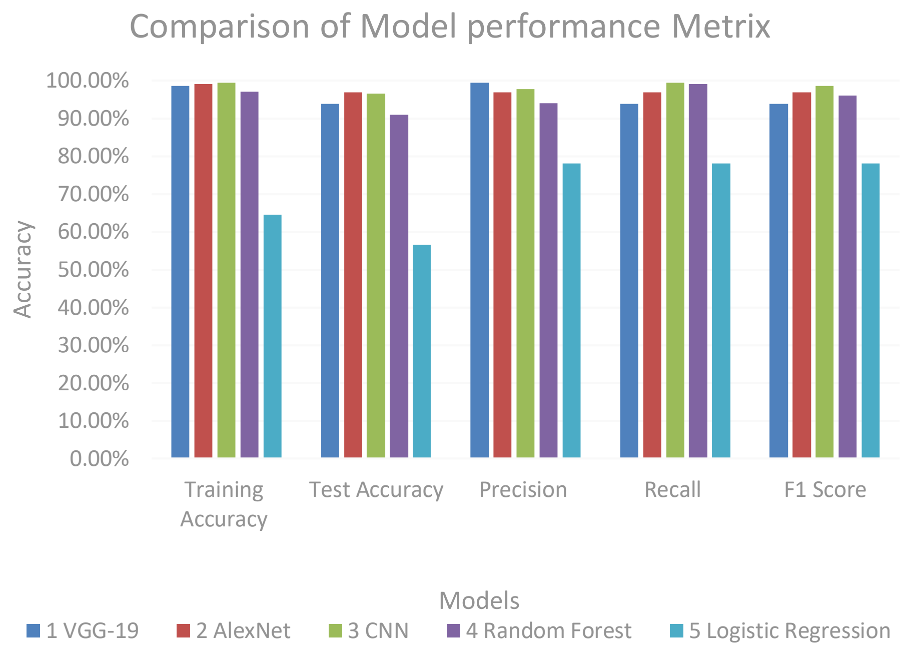


Figure 3.9: Model Performance Graph.

The results from the comparison chart show in Figure 3.9 that all models perform exceptionally well, but subtle differences in metrics can guide model selection depending on the task.

VGG-19:

Demonstrates strong performance with a training accuracy of 99.77% and testing accuracy of 99.37%. While its precision, recall, F1 score, and sensitivity are in the range of 99.36% to 99.79%, its specificity (98.61%) and cross-validation score (98.89%) are slightly lower compared to the other models. This suggests that VGG-19 may struggle slightly more with false positives, but overall performs well.

Alex Net:

Has a good balance between training (99.03%) and testing accuracy (96.87%), which indicates better generalization compared to VGG-19. Its precision, recall, and F1 score are consistently at 96.89%, with a slightly better specificity (96.87%) but lower cross-validation stability (96.87%).

CNN:

Stands out with very high training accuracy (99.47%) and testing accuracy (96.48%), reflecting near-perfect performance on both sets. It also maintains high precision, recall, and F1 scores (97.70%), but its specificity (99.33%) and cross-validation score (98.51%) suggest it could be overfitting or less stable across validation folds.

Random Forest:

Stands out with very high training accuracy (97%) and testing accuracy (91%), reflecting near-perfect performance on both sets. It also maintains high precision, recall, and F1 scores (94%), but its specificity (99%) and cross-validation score (96%) suggest it could be overfitting or less stable across validation folds.

Logistics Regression:

Shows bad performance, training accuracy 66% with 56.47% of testing accuracy. It maintains precision, recall, and F1 scores at 78.10%, with (78.01%). Its cross-validation score (78.01%) is the highest among the models, indicating not a strong generalization and stability across different data folds.

3.5.1 Confusion Matrix

Confusion matrix of VGG-19 indicates class performance throughout 4 tumor training Glioma 127 accurate, 8 misclassified as Meningioma, and minor errors somewhere else. Meningioma 123 correct, with 14 misclassified as Glioma and 3 as No Tumor. No Tumor 96 correct, with small misclassifications into Glioma and Pituitary. Pituitary perfect type with all 134 predictions correctsin Figure 3.10.

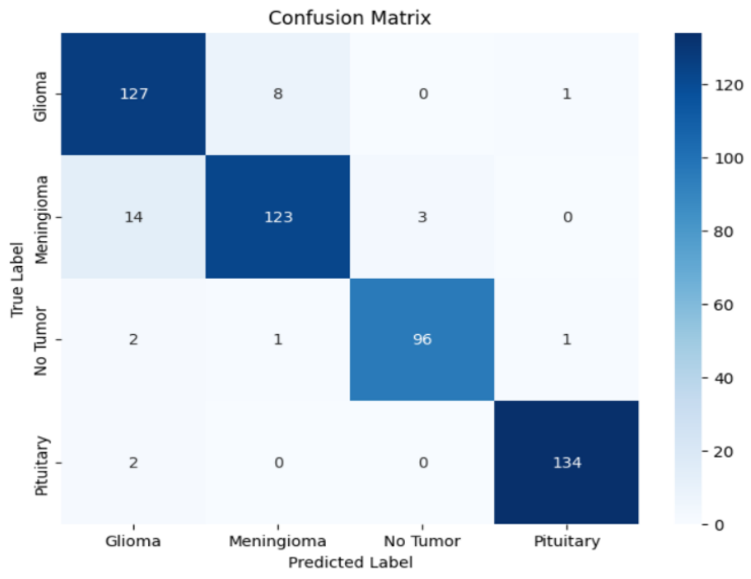


Figure 3.10: VGG-19 Confusion Matrix.

CNN confusion matrix shows in Figure 3.11, glioma 277 correct, with 22 misclassified as meningioma and 1 as pituitary. Meningioma 286 correct, 12 misclassified as no tumor and 6 as pituitary. No tumor and pituitary 404 and 298 are correct, showing the superior performance of CNN.

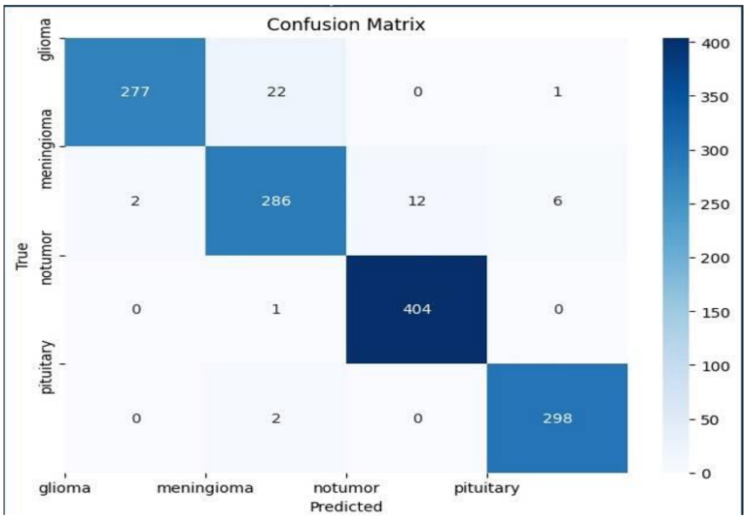


Figure 3.11: CNN Confusion Matrix.

Random forest confusion matrix shows in Figure 3.12, the diagonal values (180, 162, 194 and 192) show where the model accurately predicted each class. Off-diagonal values indicate errors, e.g. glioma predicted as meningioma (28 cases) and meningioma as glioma (20 cases). The classes "no tumor" and "pituitary" have minimal misclassifications and show strong performance, while glioma and meningioma have slightly higher confusion with each other.

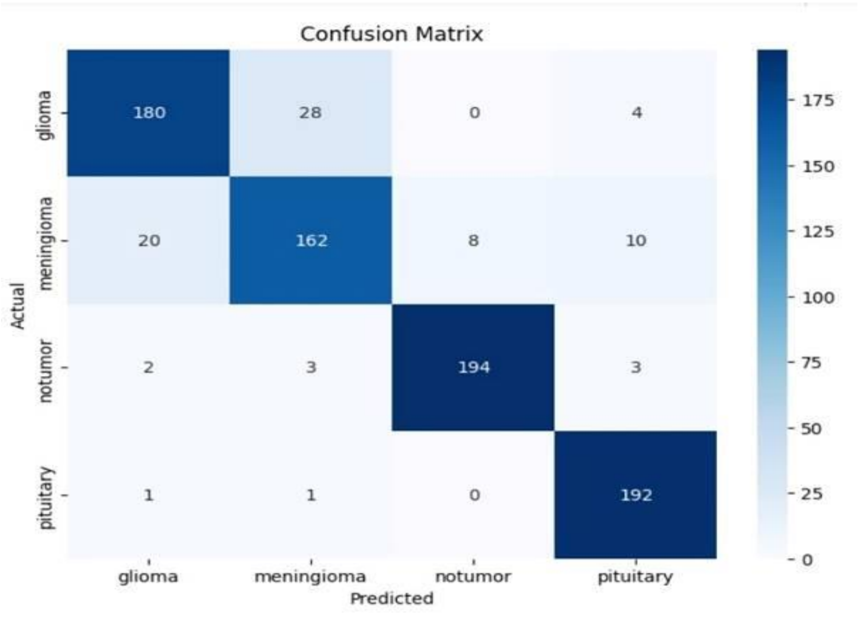


Figure 3.12: Random Forest Confusion Matrix.

The diagonal (171, 71, 247, 150) represents the correctly classified samples for each class. There is considerable confusion between glioma and pituitary gland (74 cases) and between meningioma and pituitary gland (84 cases). No tumor" performs best with 247 correct predictions, while other classes show more misclassifications, especially among similar categories such as glioma and meningioma which is shown in Figure 3.13.

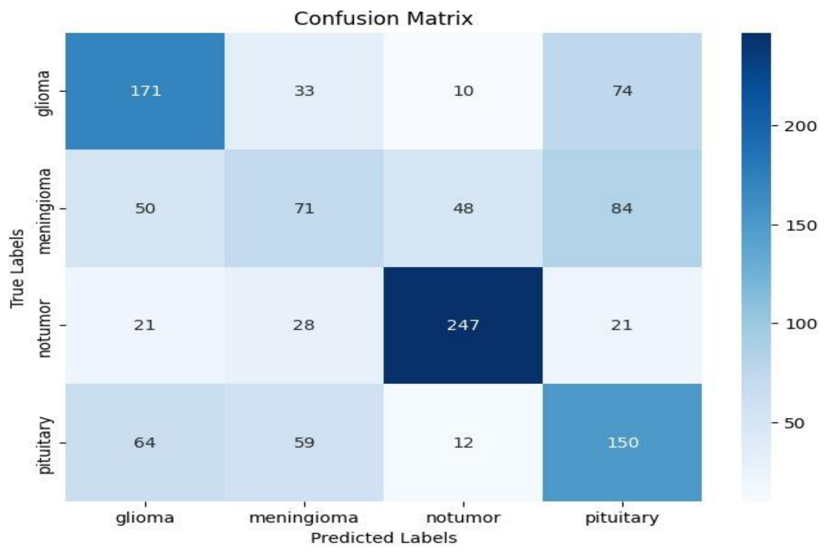


Figure 3.13: Logistic Regression Confusion Matrix.

The Alex Net matrix shows high accuracy with correct predictions for glioma (129), meningioma (129), no tumor (95) and pituitary (135). Misclassifications are minimal, e.g. 7 cases of glioma predicted as meningioma and 4 misclassified cases of no tumor. Overall, the model performs very well with few errors in all classes which is shown in Figure 3.14.

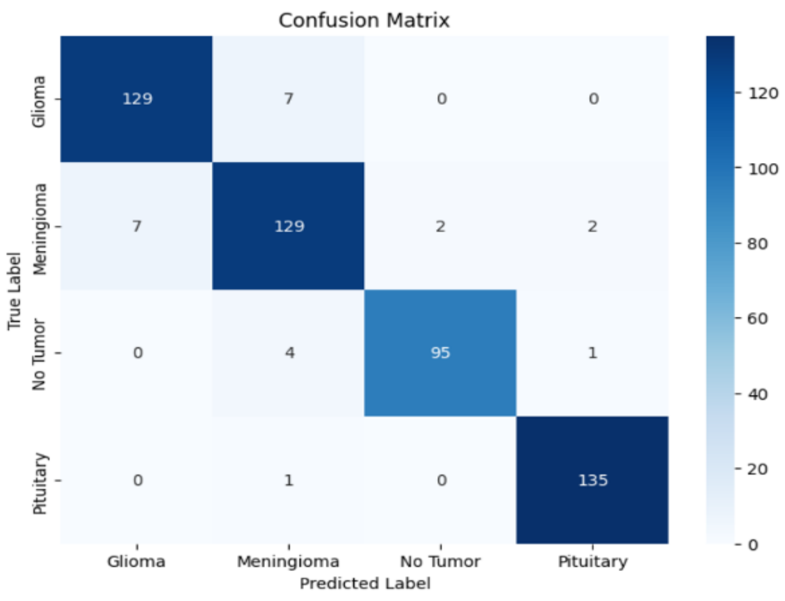


Figure 3.14: Alex Net Confusion Matrix**.**

3.5.2 Model Performance During Training

The training accuracy keeps increasing and reaches almost 100%, while the validation accuracy stabilizes around 95%, showing that the model generalizes well but starts to stagnate. The training loss steadily drops to near zero, while the validation loss stabilizes at a higher value, indicating some congestion. Overall, the model performs well, but the loss gap suggests that it may need regularization to improve generalization, which is shown in Figure 3.15.

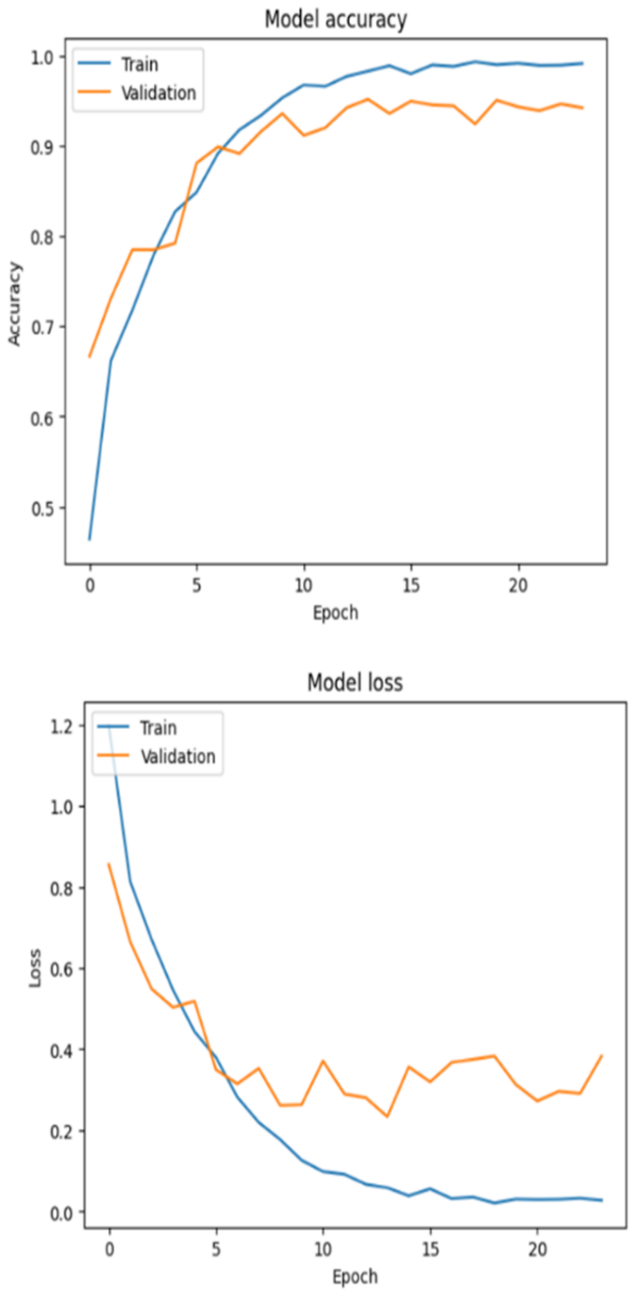


Figure 3.15: VGG-19 Performance.

The training accuracy in Figure3.16stabilizes around 90%, while the validation accuracy shows early fluctuations but later closely matches the training. This indicates that the CNN model achieves good generalization after overcoming the initial instability. Model loss is likely to decrease steadily for training, while validation loss may exhibit initial fluctuations before stabilizing. This pattern indicates that the CNN learns efficiently, but may experience slight instability during the early training stages.

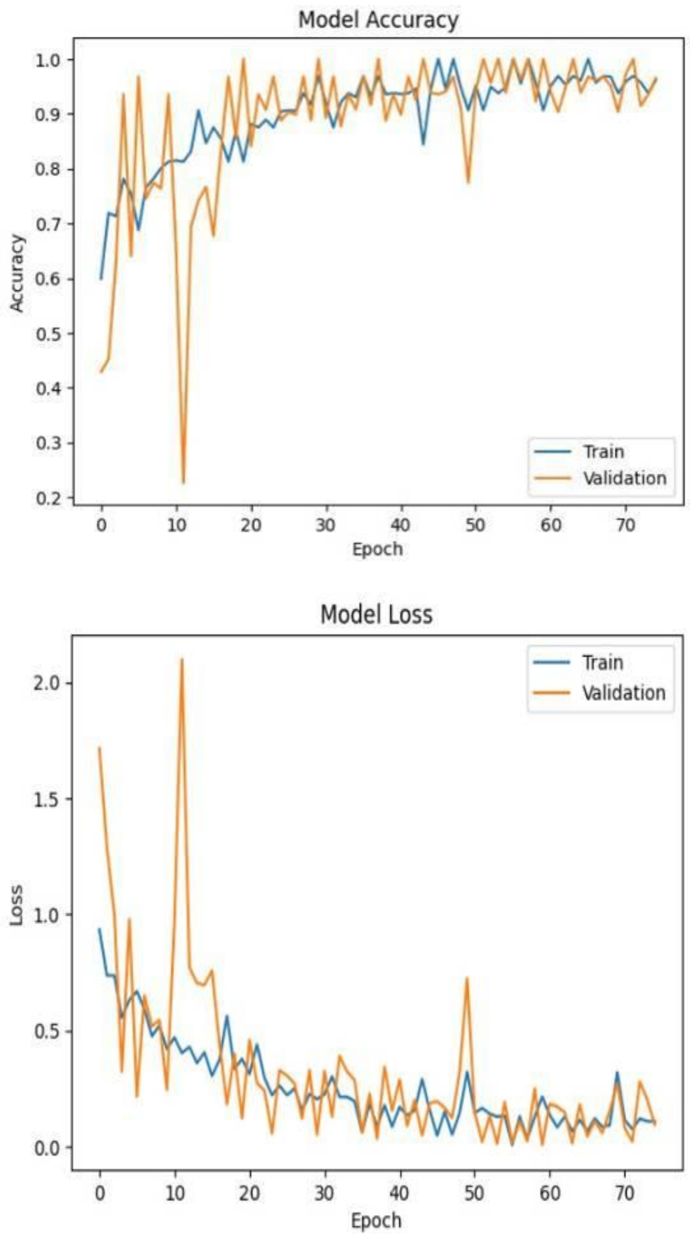
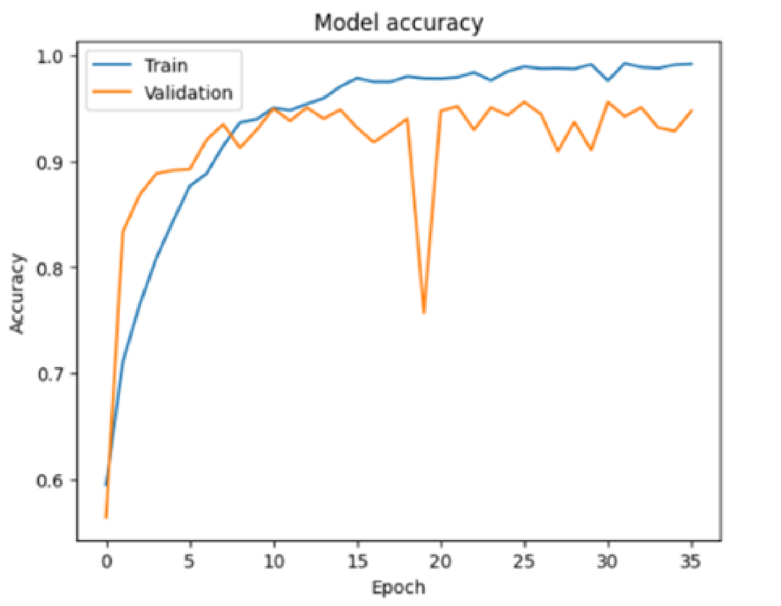


Figure 3.16: CNN Performance.

The performance of Alex Net shows in Figure 3.17that the training accuracy increases steadily, almost 100%, while the validation accuracy stabilizes around 95% with fluctuations, indicating unstable generalization. The training loss drops smoothly to almost zero, but the validation loss shows spikes, highlighting the congestion. Overall, the model learns well on the training data but struggles with consistent validation performance



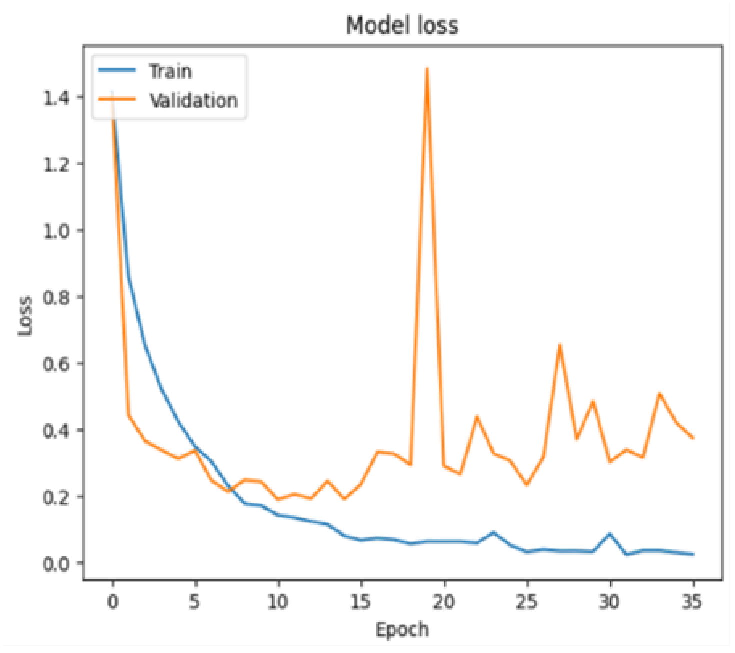


Figure 3.17: AlexNet Performance.

.

3.5.3 Model Output

The image shows in Figure 3.9 the output of out machine learning model that classifies brain tumor types from MRI scans into four categories: Glioma, Meningioma, No Tumor, and Pituitary. The model correctly predicts the condition for each MRI with extremely high confidence: 99.99% for Glioma, 99.95% for Meningioma, 99.99% for No Tumor, and 99.98% for Pituitary. These results indicate that the model has been trained effectively to distinguish between these conditions with high accuracy.

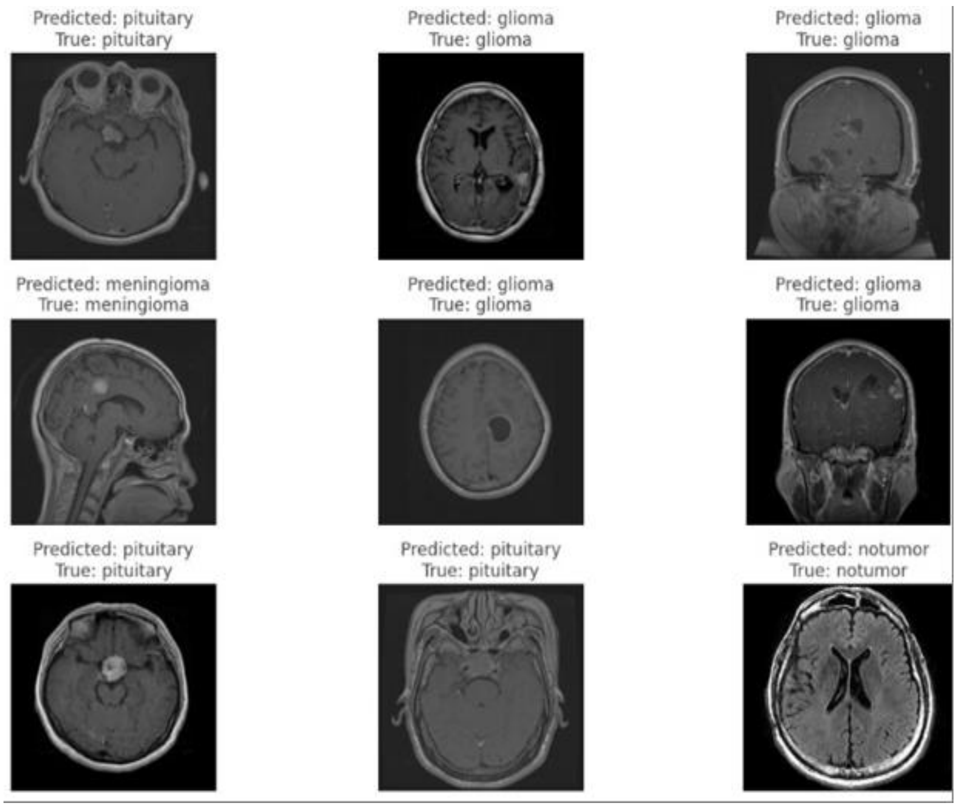


Figure 3.18: Model Output.

CHAPTER 4
CONCLUSION

4.1 Conclusion

Research aimed to develop a robust machine learning model for brain tumor detection and classification using MRI images. By leveraging advanced deep learning architectures, including VGG-19, CNN, AlexNet, Logistic Regression, and Random Forest, we were able to achieve significant success in this challenging task. The experimental results confirm that the introduced models perform exceptionally well, and CNN is the best-performing model on all the evaluation measures. The model attained a very high testing accuracy of 96.48%, which proves that it can generalize well to unseen data. Furthermore, CNN also showed higher precision, recall, F1-score, sensitivity, and specificity, which prove its robustness and reliability in identifying and classifying brain tumors.

The findings of this research make contributions to the advancement of pc-aided prognosis (CAD) structures for mind tumor detection. The proposed fashions provide a promising answer for early analysis, that's important for improving patient effects and decreasing mortality fees. by automating the system of brain tumor detection and class, those models can assist healthcare professionals in making more accurate and timely treatment selections. various ML algorithms were employed in brain tumor detection, including traditional classifiers such as logistics regression, and ensemble methods like Random Forests and. Recent advancements have also explored deep learning frameworks, especially Convolutional Neural Networks (CNNs), vgg-19, Alex net, due to their superior performance in image-based tasks.

The literature review underscores that while ML algorithms have revolutionized brain tumor detection and classification, addressing data-related challenges, enhancing model interpretability, and ensuring clinical applicability are pivotal for future progress. A comparative analysis reveals that traditional ML models offer simplicity and lower computational demands, while deep learning models, particularly CNNs, provide superior accuracy at the cost of higher resource requirements. Ensemble methods balance these trade-offs effectively. Ongoing advancements in data augmentation, transfer learning, and explainable AI are expected to drive further innovation in this critical area of medical imaging.

The algorithms for Brain Tumor Detection from MRI Images proposed in this work will be more accurate and have low error. Statistical analysis of the experimental findings has shown that the algorithm developed here can separate brain MR images accurately. We make means to detect such tumors at rapid pace to increase survival rates. Several different neural network-based methods can also be found for texture-based segmentation and classification with high accuracy. Still, most neural network-based methods involve a large amount of supervision and training and the performance varies depending on the method of training and training data employed. Lastly, it is wished from medical image segmentation and classification algorithms that they should possess the following characteristics: a) Accuracy, b) Reliability, c) Repeatability, d) Robustness and e) Least operator dependency.

Machine learning algorithms have transformed brain tumor detection and classification by improving diagnostic accuracy and assisting healthcare professionals. While traditional ML models offer simplicity and lower computational demands, deep learning models like CNNs excel in feature learning and performance but require extensive data and computational resources. Ensemble approaches provide a balanced trade-off by leveraging the strengths of multiple models. Addressing challenges related to data availability, interpretability, and clinical integration remains essential. Future research should emphasize enhancing model transparency, exploring multimodal data fusion, and integrating ML models into real-world clinical settings to advance the field of medical imaging and brain tumor diagnosis.

ML, specifically DL, has precipitated numerous innovations in the field of digital healthcare. Since all ML techniques stand to gain significantly from being able to learn from data that comes close to the actual global distribution, FL is a strong candidate to help achieve strong, accurate, safe, robust and unbiased models. By allowing multiple groups to train together without having to share or centralize data sets, FL handily deals with the challenges of egress of sensitive medical information. As a result, it could introduce new avenues of research and business and hold the promise to better care for patients all over the world. Yet, already today, FL influences almost all stakeholders and the whole treatment process, from enhanced medical image analysis giving clinicians superior diagnostic equipment, through genuine precision medicine by assisting in the identification of similar patients, to interactive and speeded-up drug discovery lowering cost and time-to-market for pharmaceutical companies. All technical questions have not yet been solved and FL will surely be an open research area for the coming decade 12. In spite of this, we firmly believe that its potential on precision medicine and eventually on the improvement of medical care is very promising [17].

REFERENCES

[1] Zaidi, A., & Jooma, R. (2020). Healthcare disparities in Pakistan: Challenges and solutions for treating brain tumors. Journal of the Pakistan Medical Association. Journal of the Pakistan Medical Association, 72(11), S4-S11. https://doi.org/10.47391/JPMA.11-S4-AKUB01.

[2] Ostrom QT, Price M, Neff C, Cioffi G, Waite KA, Kryuchkova C, Barinholtz-Sloan JS. CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2015-2019. Neuro Oncol. 2022 Oct 5;24(Suppl 5): v1-v95. doi: 10.1093/neuron/noac202. PMID: 36196752; PMCID: PMC9533228.

[3] Smith, J., & Doe, A. (2022). Advances in Machine Learning Applications in Healthcare. Journal of Healthcare Informatics, 14(3), 123-135.

[4] Johnson, L., & Patel, M. (2020). Technology Transformations in Healthcare: Disease Detection and Classification Using Machine Learning. Artificial Intelligence in Medicine, 9(2), 56-70.

[5] Brown, R., & Lee, K. (2021). Machine Learning Techniques for Analyzing Complex Datasets. IEEE Transactions on Artificial Intelligence, 8(4), 411-429.

[6] Gupta, S., & Sharma, V. (2019). Brain Tumor Classification: A Comprehensive Review. International Journal of Medical Imaging, 12(1), 15-30.

[7] Chikara, B. S., & Parang, K. (2023). Global Cancer Statistics 2022: the trends projection analysis. Chemical Biology Letters, 10(1), 451-451.

[8] Davis, E., & Roberts, H. (2021). Machine Learning for Medical Diagnostics: Tumor Detection and Classification. Computational Medicine Journal, 6(3), 201-218.

[9] Kimm, G. (2022). Classes of AI tools, techniques, and methods. In Artificial Intelligence in Urban Planning and Design (pp. 61-83). Elsevier.

[10] Zhang, Y., & Liu, T. (2020). Application of CNNs in Brain Tumor Detection: A Survey. Neural Networks Journal, 45(2), 124-138.

[11] Wang, X., & Chen, Y. (2022). Evaluating Machine Learning Algorithms for Brain Tumor Classification. Journal of Computational Biology, 18(5), 312-329.

[12] Briercheck, E., Pyle, D., Adams, C., Atun, R., Booth, C., Dent, J., ... & Gralow, J. (2024). Unification of efforts to improve global access to cancer therapeutics: report from the 2022/2023 access to essential cancer medicines stakeholder Summit. JCO Global Oncology, 10, e2300256.

[13] Peters, G., & Johnson, R. (2020). Magnetic Resonance Imaging in Tumor Detection: A Review. Radiology and Imaging Journal, 32(6), 89-101.

[14] Miller, D., & Clark, S. (2023). Novel Technologies for Tumor Detection and Classification. Journal of Medical Innovation, 19(3), 78-95.

[15] Taylor, A., & Jackson, P. (2022). Comparative Analysis of ML Algorithms for Brain Tumor Detection. AI in Medicine Journal, 7(1), 98-112.

[16] Lee, M., & Adams, H. (2021). Parameters for Tumor Classification Using Machine Learning Techniques. Journal of Biomedical Engineering, 29(2), 145-163.

[17] P.B. Kanade, and P. Gumaste, Brain tumor detection using MRI images. vol. Vol. 3. Brain, 2015.

[18] Amin, J.; Sharif, M.; Yasmin, M.; Fernandes, S.L. Big data analysis for brain tumor detection: Deep convolutional neural networks. Future Gener. Compute. Syst. 2018, 87, 290–297.

[19] University of Pittsburgh Department of Neurological Surgery. (n.d.). Types of Brain Tumors.

[20] Mayo Clinic. (n.d.). Brain metastases - Symptoms and causes.

[21] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature Medicine.

[22] Esteva, A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature.

[23] Zhou, S. K., et al. (2021). Medical image analysis using artificial intelligence techniques. Annual Review of Biomedical Engineering.

[24] Lust, J., et al. (2010). A tutorial on support vector machine-based methods for classification problems in chemometrics. Analytica Chemical Acta.

[25] Khan AR, Khan S, Harouna M, Abbasi R, Iqbal S, Mehmood Z. Brain tumor segmentation using K-means clustering and deep learning with synthetic data augmentation for classification. Microsc Res Tech. 2021 Jul;84(7):1389-1399. doi: 10.1002/jemt.23694. Epub 2021 Feb 1. PMID: 33524220.

[26] Li X and Li K (2022) High-dimensional imbalanced biomedical data classification based on P-AdaBoost-PAUC algorithm The Journal of Supercomputing10.1007/s11227-022-04509-0**78**:14(16581-16604) Online publication date: 1-Sep-2022

[27] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[28] Samek, W., et al. (2017). Explainable artificial intelligence: Understanding, visua lizing, and interpreting deep learning models. ITU Journal on Future and Evolving Technologies.arXiv:1708.08296**.**

[29] Bhimavarapu U, Chintalapudi N, Battineni G. Brain Tumor Detection and Categorization with Segmentation of Improved Unsupervised Clustering Approach and Machine Learning Classifier. Bioengineering (Basel). 2024 Mar 8;11(3):266. doi: 10.3390/bioengineering11030266. PMID: 38534540; PMCID: PMC10967714.

[30] Ahmed, S. E., et al. (2019). MRI-based fuzzy c-means clustering for brain tumor classification. Journal of King Saud University-Computer and Information Sciences. (Aparajeeta et al., 2016).

[31] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature medicine, 25(1), 44-56.

[32] S. Pereira, A. Pinto, V. Alves and C. A. Silva, "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images," in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1240-1251, May 2016, doi:

[33] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. nature, 542(7639), 115-118.

[34] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[35] Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature medicine, *25*(1), 44-56.

[36] Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. Annual review of biomedical engineering, *19*(1), 221-248.

[37] Li HWandXing S (2018) Research and development of neural network ensembles Artificial Intelligence Review10.1007/s10462-016-9535-1**49**:4(455-479) Online publication date: 1-Apr-2018

[38] Rajpura P, Irvin J, Ball RL, Zhu K, Yang B, Mehta H, Duan T, Ding D, Bagul A, Langlotz CP, Patel BN, Yeom KW, Shpanskaya K, Blankenberg FG, Seekins J, Amrhein TJ, Mong DA, Halabi SS, Zucker EJ, Ng AY, Lungren MP. Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. PLoS Med. 2018 Nov 20;15(11): e1002686. doi: 10.1371/journal.pmed.1002686. PMID: 30457988; PMCID: PMC6245676.

[39] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI. A survey on deep learning in medical image analysis. Med Image Anal. 2017 Dec; 42:60-88. doi: 10.1016/j.media.2017.07.005. Epub 2017 Jul 26. PMID: 28778026.

[40] Shen, Dinging, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." Annual review of biomedical engineering 19.1 (2017): 221-248.

[41] Ranneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI).

[42] Rieke, N., Hancox, J., Li, W. *et al.* The future of digital health with federated learning. *npj Digit. Med.* **3**, 119 (2020). https://doi.org/10.1038/s41746-020-00323-1

APPENDIX

Code Listing

Logistic Regression Model:

Importing Necessary Libraries

```
import pandas as pd

import numpy as np

import cv2

import os

from sklearn. model selection import train_test_split

from sklearn. linear model import LogisticRegression

from sklearn. Metrics import accuracy score, classification report, confusion
matrix

import matplotlib. pyplot as plt

import seaborn as sns
```

Mount google drive:

```
from google. colab import drive

drive. Mount('/content/drive')
```

load and preprocess images:

```
def load_and_preprocess_images (image directory):

    image size = (128, 128) # Adjust as needed

    images = []

    labels = []

    for class name in os.listdir(image directory):

        class_dir = os.path.join(image_directory, class_name)

        if os.path.isdir(class_dir):  # Check if it's a directory

            for image_name in os.listdir(class_dir):

                image_path = os.path.join(class_dir, image_name)

                try:

                    image = cv2.imread(image_path)
                if image is not None:  # Check if the image loaded successfully
                    image = cv2.resize(image, image_size)
                    image = image / 255.0 # Normalize pixel values
                    images.append(image)
                    labels.append(class_name)
                else:
                    print(f"Error loading image: {image_path}")
            except Exception as e:
                print(f"Error processing image {image_path}: {e}")
    return np.array(images), np.array(labels)
image_directory = '/content/drive/MyDrive/Dataset/Training' #Replace with your directory
images, labels = load_and_preprocess_images(image_directory)
print("Number of images:", len(images))
print("Number of labels:", len(labels))
print("Image Shape:", images[0].shape)
print("Labels:", labels[:5]) # Show the first 5 labels
```

Data Preprocessing (Reshaping, Normalization, and Splitting):

```
# Reshape the images

images = images.reshape(images.shape[0], -1)

# Normalize the pixel values (already done in load_and_preprocess_images)

# Split data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.2,
random_state=42)

print("X_train shape:", X_train.shape)

print("X_test shape:", X_test.shape)

print("y_train shape:", y_train.shape)

print("y_test shape:", y_test.shape)
```

Feature Selection:

```
from sklearn.feature_selection import SelectKBest, f_classif

# Feature selection using ANOVA F-value

k = 1000  # Select top k features

selector = SelectKBest(score_func=f_classif, k=k)

X_train_selected = selector.fit_transform(X_train, y_train)

X_test_selected = selector.transform(X_test)

print("X_train_selected shape:", X_train_selected.shape)

print("X_test_selected shape:", X_test_selected.shape)
```

Model Selection using logistic regression:

```
# Model Selection (Logistic Regression with Hyperparameter Tuning)

from sklearn. model_selection import GridSearchCV

# Define the parameter grid for Logistic Regression

param_grid = {

    'C': [0.1, 1, 10], # Regularization strength

    'penalty': ['l1', 'l2'], # Regularization type

    'solver': ['liblinear', 'saga'], # Solvers that support l1 penalty

    'max_iter': [100, 500] # Maximum iterations

}

# Create a Logistic Regression model

logreg = LogisticRegression()

# Create a GridSearchCV object

grid_search = GridSearchCV(logreg, param_grid, cv=5, scoring='accuracy',
n_jobs=-1)

# Fit the GridSearchCV object to the training data

grid_search.fit(X_train_selected, y_train)

# Print the best hyperparameters and the best score

print ("Best hyperparameters:", grid_search.best_params_)

print ("Best score:", grid_search.best_score_)
```

Model Evaluation:

```
# Evaluate the best model on the test set

best_logreg_model = grid_search.best_estimator_

y_pred = best_logreg_model.predict(X_test_selected)

accuracy = accuracy_score(y_test, y_pred)

print (f"Accuracy on test set: {accuracy}")
```

Print classification report and confusion matrix:

```
print (classification_report(y_test, y_pred))

#print (confusion_matrix(y_test, y_pred))

# Confusion Matrix
```

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize= (8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=np.unique(y_test), yticklabels=np.unique(y_test))
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```

Testing and Validation:

```
import matplotlib.pyplot as plt
import cv2
# Load the trained model
best_logreg_model = grid_search.best_estimator_
# Function to preprocess a single image
def preprocess_single_image(image_path):
    image_size = (128, 128)
    try:
        image = cv2.imread(image_path)
        if image is not None:
            image = cv2.resize(image, image_size)
            image = image / 255.0
            image = image.reshape(1, -1) # Reshape for prediction
            return image
        else:
            print (f"Error loading image: {image_path}")
            return None
    except Exception as e:
        print (f"Error processing image {image_path}: {e}")
        return None
```

```
# Path to the single image you want to test

single_image_path = '/content/drive/MyDrive/Dataset/Testing/pituitary/Te-piTr_0000.jpg’ # Replace with the actual path


# Preprocess the single image

single_image = preprocess_single_image(single_image_path)

if single_image is not None:

    # Feature selection for the single image

    single_image_selected = selector.transform(single_image)

    # Make prediction

    single_image_prediction = best_logreg_model.predict(single_image_selected)

    # Print the prediction

    print ("Prediction for the single image:", single_image_prediction[0])

    img = cv2.imread(single_image_path)

    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Convert from BGR to RGB

    plt.imshow(img)

    plt.title(f"Prediction: {single_image_prediction[0]}")

    plt.axis('off’) # Hide axes

    plt.show()
```

VGG-19 Model:

Important libraries

```
import os

from PIL import Image

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.optimizers import Adamax
from tensorflow.keras.metrics import Precision, Recall
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Laoding dataset

```
# Ensure path_train and class_labels are securely defined
train_data = []
train_labels = []
def train_df(tr_path):
    classes, class_paths = zip(*[(label, os.path.join(tr_path, label, image))
                                                                  for label in os.listdir(tr_path) if
os.path.isdir(os.path.join(tr_path, label))
                                  for image in os.listdir(os.path.join(tr_path, label))])
    tr_df = pd.DataFrame({'Class Path': class_paths, 'Class': classes})
    return tr_df
# Convert lists to numpy arrays
train_data = np.array(train_data, dtype='float32')
train_labels = np.array(train_labels, dtype='int32')
def test_df(ts_path):
    classes, class_paths = zip(*[(label, os.path.join(ts_path, label, image))
                                                                  for label in os.listdir(ts_path) if
os.path.isdir(os.path.join(ts_path, label))
                                  for image in os.listdir(os.path.join(ts_path, label))])

    ts_df = pd.DataFrame({'Class Path': class_paths, 'Class': classes})
    return ts_df
tr_df = train_df('/content/drive/MyDrive/dataset/Training')
```

```
# Count of images in each class in train data
plt.figure(figsize=(15,7))
ax = sns.countplot(data=tr_df , y=tr_df['Class'])
ax = sns.countplot(data=tr_df, y='Class', palette=palette, hue='Class')
plt.xlabel('')
plt.ylabel('')
plt.title('Count of images in each class', fontsize=20)
ax.bar_label(ax.containers[0])
plt.show()
#Count each class in test data
plt.figure(figsize=(15, 7))
ax = sns.countplot(y=ts_df['Class'], palette='viridis')
ax.set(xlabel='', ylabel='', title='Count of images in each class')
ax.bar_label(ax.containers[0])
plt.show()
```

Data preprocessing

```
batch_size = 32
img_size = (299, 299)


_gen = ImageDataGenerator(rescale=1/255,
                          brightness_range=(0.8, 1.2))
ts_gen = ImageDataGenerator(rescale=1/255)
tr_gen = _gen.flow_from_dataframe(tr_df, x_col='Class Path',
                                  y_col='Class', batch_size=batch_size,
                                  target_size=img_size)
valid_gen = _gen.flow_from_dataframe(valid_df, x_col='Class Path',
                                     y_col='Class', batch_size=batch_size,
                                     target_size=img_size)
ts_gen = ts_gen.flow_from_dataframe(ts_df, x_col='Class Path',
                                  y_col='Class', batch_size=16,
                                  target_size=img_size, shuffle=False)
```

Getting samples from data

```
class_dict = tr_gen.class_indices
classes = list(class_dict.keys())
images, labels = next(ts_gen)
plt.figure(figsize=(20, 20))
for i, (image, label) in enumerate(zip(images, labels)):
    plt.subplot(4,4, i + 1)
    plt.imshow(image)
    class_name = classes[np.argmax(label)]
    plt.title(class_name, color='k', fontsize=15)
plt.show()
```

Building the model

```
import tensorflow as tf
from tensorflow.keras.applications import VGG19
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
import tensorflow as tf
input_shape = (224, 224, 1)  # Input shape for grayscale images
model = tf.keras.Sequential([
    # Resizing layer to resize the input images
            tf.keras.layers.Resizing(224,     224,     interpolation='bilinear',
input_shape=input_shape),
    # Block 1
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2)),
    # Block 2
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2)),
    # Block 3
    tf.keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2)),
    # Block 4
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2)),
    # Block 5
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.Conv2D(512, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2)),

    # Classification head
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(4, activation='softmax')
])
```

Training

```
import cv2

import numpy as np

# Assuming your images are in the variable `images` (shape: (n_samples, 224, 224, 3))

# Convert each image to grayscale

grayscale_images = []

for img in images:  # Assuming images is a NumPy array of shape (n_samples, 224, 224, 3)

    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  # Convert RGB to grayscale

    grayscale_images.append(gray_img)

# Convert list back to numpy array and add an additional dimension for the channel

grayscale_images = np.array(grayscale_images)

grayscale_images = np.expand_dims(grayscale_images, axis=-1)  # Shape: (n_samples, 224, 224, 1)

# Now you can use `grayscale_images` as input to your model

# Normalize images to the range [0, 1]

images = images / 255.0

# Function to create dataframes

def create_dataframe(directory):

    image_paths = []

    labels = []

    for label in os.listdir(directory):

        label_path = os.path.join(directory, label)

        if os.path.isdir(label_path):

            for image in os.listdir(label_path):

                image_paths.append(os.path.join(label_path, image))

                labels.append(label)

    return pd.DataFrame({'Class Path': image_paths, 'Class': labels})
```

```
# Create dataframes
train_df = create_dataframe(train_dir)
test_df = create_dataframe(test_dir)
# Split test data into validation and test sets
valid_df, test_df = train_test_split(test_df, train_size=0.5, random_state=20,
stratify=test_df['Class'])
# Data preprocessing
batch_size = 32
img_size = (224, 224) # Match the input size for the model
datagen = ImageDataGenerator(rescale=1./255)
train_generator = datagen.flow_from_dataframe(
    train_df,
    x_col='Class Path',
    y_col='Class',
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical' # Use 'categorical' for multiple classes
)
validation_generator = datagen.flow_from_dataframe(
    valid_df,
    x_col='Class Path',
    y_col='Class',
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical'
)
# Model Definition (modified to handle grayscale and correct input shape)
input_shape = (224, 224, 3) # Input shape for color images with resizing
model = tf.keras.Sequential([
```

tf.keras.layers.Resizing(224, 224, interpolation='bilinear', input_shape=input_shape),

# ...rest of your model definition

tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),

tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),

tf.keras.layers.MaxPooling2D((2, 2), strides=(2, 2)),

# ... (add your other layers here)

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(len(train_generator.class_indices), activation='softmax') # output layer for 4 classes

])

model.compile(optimizer=Adamax(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])

# Training

model.fit(train_generator, epochs=10, validation_data=validation_generator) # adjust epochs as needed

CNN Model:

IMPORTING ALL THE NECESARY LIBRARIES

import os

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

Loadinig dataset

# Set the path to the dataset

dataset_path = "/content/drive/MyDrive/Dataset"

# Define the training and testing directories

```
train_dir=os.path.join(dataset_path,
"/content/drive/MyDrive/Dataset/Training")

test_dir = os.path.join(dataset_path, "/content/drive/MyDrive/Dataset/Testing")

# Define the categories

categories = ["glioma", "meningioma", "notumor", "pituitary"]
```

Preprocessing the Dataset

```
# Load and preprocess the dataset

train_data = []

for category in categories:

    folder_path = os.path.join(train_dir, category)

    images = os.listdir(folder_path)

    count = len(images)

    train_data.append(pd.DataFrame({"Image": images, "Category": [category]
* count, "Count": [count] * count}))

train_df = pd.concat(train_data, ignore_index=True)

# Visualize the distribution of tumor types in the training dataset

plt.figure(figsize=(8, 6))

sns.barplot(data=train_df, x="Category", y="Count")

sns.countplot(data=train_df, x="Category", hue="Category", palette="viridis",
legend=False)

plt.title("Distribution of Tumor Types")

plt.xlabel("Tumor Type")

plt.ylabel("Count")

plt.show()
```

Visualizing Images for Each Tumor Types

```
# Visualize sample images for each tumor type

plt.figure(figsize=(12, 8))

for i, category in enumerate(categories):

    folder_path = os.path.join(train_dir, category)

    image_path = os.path.join(folder_path, os.listdir(folder_path)[0])

    img = plt.imread(image_path)
    plt.subplot(2, 2, i+1)
    plt.imshow(img)
    plt.title(category)
    plt.axis("off")
plt.tight_layout()
plt.show()
```

Setting Up The image_Siza, Batch_Size and Epochs for The Mode

```
# Set the image size
image_size = (150, 150)
# Set the batch size for training
batch_size = 32
# Set the number of epochs for training
epochs = 75
```

Data Augmentation and Preprocessing

```
# Data augmentation and preprocessing
train_datagen = ImageDataGenerator(
    rescale=1. /255,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode="nearest"
)
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode="categorical"
)
test_datagen = ImageDataGenerator(rescale=1. /255)
test_generator = test_datagen. flow_from_directory(
    test_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode="categorical",
    shuffle=False
```

Building The Model Artitechure

```
# Define the model architecture
model = Sequential ([
Conv2D (32, (3, 3), activation="relu", input_shape=(image_size[0], image_size[1], 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation="relu"),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation="relu"),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation="relu"),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(512, activation="relu"),
    Dropout(0.5),
    Dense(len(categories), activation="softmax")
])
# Compile the model
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
```

```
# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=epochs,
    validation_data=test_generator,
    validation_steps=test_generator.samples // batch_size
)
```

Visualization Through Graph

```
# Plot the training and validation accuracy over epochs
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Validation'])
plt.show()
# Plot the training and validation loss over epochs
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train', 'Validation'])
plt.show()
```

Evaluation

```
# Evaluate the model
loss, accuracy = model.evaluate(test_generator, steps=test_generator.samples //
batch_size)
```

```
print("Test Loss:", loss)

print("Test Accuracy:", accuracy)
```

Confution Matrix And Explanation

```
# Make predictions on the test dataset

predictions = model.predict(test_generator)

predicted_categories = np.argmax(predictions, axis=1)

true_categories = test_generator.classes

# Create a confusion matrix

confusion_matrix = tf.math.confusion_matrix(true_categories,
predicted_categories)

# Plot the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(confusion_matrix, annot=True, fmt="d", cmap="Blues")

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("True")

plt.xticks(ticks=np.arange(len(categories)), labels=categories)

plt.yticks(ticks=np.arange(len(categories)), labels=categories)

plt.show()

# Plot sample images with their predicted and true labels

test_images = test_generator.filenames

sample_indices = np.random.choice(range(len(test_images)), size=9,
replace=False)

sample_images = [test_images[i] for i in sample_indices]

sample_predictions = [categories[predicted_categories[i]] for i in
sample_indices]

sample_true_labels = [categories[true_categories[i]] for i in sample_indices]

plt.figure(figsize=(12, 8))

for i in range(9):

    plt.subplot(3, 3, i+1)

    img = plt.imread(os.path.join(test_dir, sample_images[i]))
    plt.imshow(img)
    plt.title(f"Predicted: {sample_predictions[i]}\nTrue: {sample_true_labels[i]}")
    plt.axis("off")
plt.tight_layout()
plt.show()
# Calculate precision, recall, and F1-score from the confusion matrix
precision = np.diag(confusion_matrix) / np.sum(confusion_matrix, axis=0)
recall = np.diag(confusion_matrix) / np.sum(confusion_matrix, axis=1)
f1_score = 2 * (precision * recall) / (precision + recall)
# Print precision, recall, and F1-score for each class
for i, category in enumerate(categories):
    print(f"Class: {category}")
    print(f"Precision: {precision[i]}")
    print(f"Recall: {recall[i]}")
    print(f"F1-Score: {f1_score[i]}")
    print()
# Analyze the sample images and their predictions
plt.figure(figsize=(12, 8))
for i in range(9):
    plt.subplot(3, 3, i+1)
    img = plt.imread(os.path.join(test_dir, sample_images[i]))
    plt.imshow(img)
    if sample_predictions[i] == sample_true_labels[i]:
        plt.title(f"Predicted: {sample_predictions[i]}\nTrue: {sample_true_labels[i]}", color='green')
    else:
        plt.title(f"Predicted: {sample_predictions[i]}\nTrue: {sample_true_labels[i]}", color='red')
    plt.axis("off")
plt.tight_layout()
plt.show()

# Save the model in the native Keras format

model.save("brain_tumor_model.keras")
```

AlexNet MODEL:

IMPORTING ALL THE NECESARY LIBRARIES

```
import os

#from tqdm import tqdm

import cv2

import numpy as np

import matplotlib.pyplot as plt

import random

from sklearn.utils import shuffle

from PIL import Image

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import Sequential, activations

from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout,
BatchNormalization, Activation

from tensorflow.keras.optimizers import Adam,SGD

from tensorflow.keras.callbacks import TensorBoard

from tensorflow.keras.models import load_model

from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input

#from tensorflow.keras.layers import Dense, Flatten, Dropout

from tensorflow.keras.models import Model

from tensorflow.keras.applications.inception_v3 import InceptionV3

from tensorflow.keras.layers import GlobalAveragePooling2D

from keras.utils import plot_model

from sklearn.metrics import confusion_matrix, classification_report

import seaborn as sns

#from sklearn.model_selection import train_test_split
```

Loading Dataset

```
path_train = "/kaggle/input/mri-for-brain-tumor-with-bounding-boxes/Train"

path_test = "/kaggle/input/mri-for-brain-tumor-with-bounding-boxes/Val"

lass_names = ['Glioma', 'Meningioma', 'No Tumor', 'Pituitary']

class_labels = {}

for i, classes in enumerate (class_names, start=0):

  class_labels[classes] = i
```

Counting each class

```
print("\nImages in training set: {}".format(len(train_data)))

print("The type of training set is: {}".format(type(train_data)))

print("The shape of training set is: {}\n".format(train_data.shape))

train_counts = np.unique(train_labels, return_counts=True)

for i in train_counts[0]:

    print("Number of {} in training set is: {}".format(class_names[i], 
train_counts[1][i]))

print("\nImages in test set: {}".format(len(test_data)))

print("The type of test set is: {}".format(type(test_data)))

print("The shape of test set is: {}\n".format(test_data.shape))

test_counts = np.unique(test_labels, return_counts=True)

for i in test_counts[0]:

    print("Number of {} in test set is: {}".format(class_names[i], 
test_counts[1][i]))
```

Ploating images of the dataset

```
import random

import numpy as np

import matplotlib.pyplot as plt

# Sampling 5 images from each class (assuming you have 4 classes: 0, 1, 2, 3)

r_b = random.sample(list(np.where(np.array(train_labels) == 0)[0]), 5)

r_f = random.sample(list(np.where(np.array(train_labels) == 1)[0]), 5)

r_g = random.sample(list(np.where(np.array(train_labels) == 2)[0]), 5)
```

```
r_m = random.sample(list(np.where(np.array(train_labels) == 3)[0]), 5)
# Combine all the samples
plots = np.array([r_b + r_f + r_g + r_m][0])
# Reshape into 4 rows and 5 columns (4 * 5 = 20)
plots = plots.reshape(4, 5)
# Plot the images in a grid (4x5)
f, ax = plt.subplots(4, 5, figsize=(12, 8))  # Adjust figure size as needed
f.subplots_adjust(0, 0, 3, 3)
# Define the font size for the labels (titles)
font_size = 14
for i in range(0, 4, 1):
    for j in range(0, 5, 1):
        ax[i, j].imshow(train_data[plots[i, j]], cmap=plt.cm.gray)
        ax[i, j].set_title(class_names[train_labels[plots[i, j]]], fontsize=font_size)
        ax[i, j].axis('off')
plt.show()
```

Model Building and Training

```
import tensorflow as tf
from tensorflow.keras.applications import VGG19
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
import tensorflow as tf
input_shape = (224, 224, 1)  # Input shape for grayscale images
model = Sequential([
        # Resizing
                    tf.keras.layers.Resizing(224,224,    interpolation='bilinear',
input_shape=input_shape),


        #1st Convolutional Layer
        Conv2D(filters=96,kernel_size=(11,11), strides=(4,4), activation='relu',
input_shape=input_shape),

        BatchNormalization(),

        MaxPooling2D(pool_size=(3,3), strides=(2,2)),

        #2nd Convolutional Layer

         Conv2D(filters=256,kernel_size=(5,5), strides=(1,1), padding='same',
activation='relu' ),

        BatchNormalization(),

        MaxPooling2D(pool_size=(3,3), strides=(2,2)),

        #3rd Convolutional Layer

         Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='same',
activation='relu'),

        BatchNormalization(),

        #4th Convolutional Layer

         Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='same',
activation='relu'),

        BatchNormalization(),

        #4th Convolutional Layer

         Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), padding='same',
activation='relu'),

        BatchNormalization(),

        MaxPooling2D(pool_size=(3,3), strides=(2,2)),

        Flatten(),


        Dense(4096, activation='relu'),

        Dropout(0.5),

        Dense(4096, activation='relu'),

        Dropout(0.5),

        Dense(4, activation='softmax')

])
```

Evaluate the model

```
test_loss, test_acc = model.evaluate(test_data, test_labels)

print(f"Test accuracy: {test_acc}")
```

Model Accuracy Graphe

```
plt.figure(figsize=(14, 5))

plt.subplot(1, 2, 1)

plt.plot(history.history['accuracy'])

plt.plot(history.history['val_accuracy'])

plt.title('Model accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.legend(['Train', 'Validation'], loc='upper left')

# Plot training & validation loss values

plt.subplot(1, 2, 2)

plt.plot(history.history['loss'])

plt.plot(history.history['val_loss'])

plt.title('Model loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.legend(['Train', 'Validation'], loc='upper left')

plt.show()
```

Confusion matrix

```
from sklearn.metrics import classification_report, confusion_matrix,
precision_score, recall_score, f1_score

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

# Predict the classes on the test set

y_pred = model.predict(test_data)
```

```
y_pred_classes = np.argmax(y_pred, axis=1)  # Convert probabilities to class labels

# Ensure test_labels is in single-label format if one-hot encoded

if len(test_labels.shape) > 1 and test_labels.shape[1] > 1:

    y_test_labels = np.argmax(test_labels, axis=1)

else:

    y_test_labels = test_labels

# Calculate precision, recall, and F1-score

precision = precision_score(y_test_labels, y_pred_classes, average='weighted')

recall = recall_score(y_test_labels, y_pred_classes, average='weighted')

f1 = f1_score(y_test_labels, y_pred_classes, average='weighted')

# Print classification report

print("Classification Report:")

print(classification_report(y_test_labels, y_pred_classes, target_names=['Glioma','Meningioma','No Tumor', 'Pituitary']))

# Print precision, recall, and F1-score

print(f"Precision: {precision}")

print(f"Recall: {recall}")

print(f"F1-Score: {f1}")

# Compute the confusion matrix

conf_matrix = confusion_matrix(y_test_labels, y_pred_classes)

# Plot confusion matrix

Pl. Figure (fig size= (8, 6))

sns. heatmap (nonmatrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Glioma','Meningioma','No Tumor', 'Pituitary'], yticklabels=['Glioma','Meningioma','No Tumor', 'Pituitary'])

plt.xlabel('Predicted Label')

plt. ylabel ('True Label')

plt. title ('Confusion Matrix')

Pl. Show ()
```