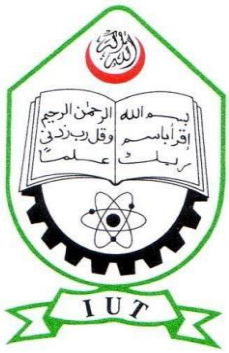
Multi-class Classification of Brain Tumors from MRI Image using Hybrid CNN

by

Fariha Mahjabin Islam (200021304)
Sahat Abrar Rafij (200021333)
Feehad Kamal (200021218)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING



Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

October 2025

DECLARATION

I do hereby declare that the work presented in this thesis, titled **"Multi-class Classification of Brain Tumors from MRI Image using Hybrid CNN"**, is the result of our own research carried out under the guidance of **Mr. Ashraful Islam Mridha**, Lecturer, Department of Electrical and Electronic Engineering (EEE), **Islamic University of Technology (IUT)**. This thesis was completed as part of the requirements for the degree of **Bachelor of Science (B.Sc.) in Electrical and Electronic Engineering** at this university. No part of this work has been submitted elsewhere, in full or in part, for the award of any other academic qualification, certificate, diploma, or degree.
All sources of information have been duly acknowledged, and wherever the work of others has been cited, it has been properly referenced.

Fariha Mahjabin Islam (200021304)

Sahat Abrar Rafij (200021333)

Feehad Kamal (200021218)

Multi-class Classification of Brain Tumors from MRI Image using Hybrid CNN

Approved by:

Ashraful Islam Mridha

Supervisor and Lecturer,
Department of Electrical and Electronic Engineering,
Islamic University of Technology (IUT),
Boardbazar, Gazipur-1704.

Date:

ACKNOWLEDGEMENTS

First and foremost, we express our heartfelt gratitude to Almighty Allah for granting us the strength, patience, and perseverance to successfully complete our thesis work. His blessings have guided us through every stage of this journey.

We would like to extend our deepest appreciation to our honorable supervisor, Mr. Ashraful Islam Mridha, Lecturer, Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT), for his continuous guidance, valuable suggestions, and constant encouragement throughout this research. His insightful advice and constructive feedback have been instrumental in shaping this work and enriching our understanding of deep learning and medical image analysis.

We are also grateful to Prof. Dr. Syed Iftekhar Ali, Head of the Department of Electrical and Electronic Engineering, IUT, and to all the faculty members for their support, motivation, and for providing an excellent academic environment conducive to research.

Finally, we would like to express our sincere appreciation to our families and friends for their unwavering encouragement, patience, and moral support during this endeavor. Their belief in us has been a constant source of inspiration and strength.

ABSTRACT

Brain tumors are a significant health concern, requiring timely and accurate diagnosis to improve treatment outcomes. Magnetic Resonance Imaging (MRI) is a widely used approach for detecting brain tumors, and the traditional method of MRI interpretation is manual, time-consuming and liable to errors. In this study, a hybrid deep neural network is proposed utilizing the architectural features of VGG16, ResNet50 and EfficientNetB3 for multi-class brain tumors classification. The aim is to develop a model that is not only efficient and high performing, but also capable of classifying four tumorous and non-tumorous brain MRI scans: glioma, meningioma, and pituitary tumors. The proposed model was trained and evaluated on a publicly available dataset from Kaggle, containing 7,023 MRI images. A robust preprocessing pipeline was used which consisted of normalization, resizing, and augmentation to adapt the images to be more consistent and to reinforce generalization for the training. The model was trained using a 5-fold cross-validation strategy to ensure unbiased performance and robustness across different subsets of the data. The hybrid model achieved an accuracy of 98.77%, with weighted precision, recall, and F1-scores all also recorded at 98.77%. Additionally, the model's compact size of just 2.20 MB ensures computational efficiency and adaptability, making it suitable for resource-constrained environments. Compared to existing classification methods, which require higher storage and computational resources, the proposed model performs effectively while requiring fewer resources. This research demonstrates the potential of hybrid deep learning models for medical image classification, providing a promising solution for faster and more accurate brain tumor diagnosis. The findings suggest that the model's compact size and high accuracy could significantly enhance clinical workflows and decision-making processes in medical imaging.

Table of Contents

List of Table

List of Figures

List of Acronyms

MRI	Medical Resonance Imaging
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
SVMs	Support Vector Machines
RFs	Random Forests
DWT	Discrete Wavelet Transform
FNN	Feedforward Neural Network
VGG	Visual Geometry Group
ResNet	Residual Neural Network
SE	Squeeze-and-Excitation

CHAPTER 1

INTRODUCTION

The brain is the central organ of the human body and is responsible for sensory perception, regulation of body processes, and higher order cognitive actions including reasoning, memory, and decision making. Like other necessary tissues and organs, the brain can be affected by diseases and disorders that can affect function. Among the major health concerns in terms of the brain, the development of brain tumors is a central issue that can greatly affect health, quality of life, and survival. A brain tumor is defined as the uncontrolled growth of abnormal cells in the brain or central spinal canal. Depending on the type and location of a tumor, co-morbid conditions or other factors, a tumor may exert pressure on adjacent healthy brain tissues, interfere with normal brain function, and result in devastating functional impairments. Early and accurate identification and classification of a brain tumor are therefore important for treatment and management of brain tumors.

Magnetic Resonance Imaging (MRI) is the most widely used non-invasive imaging method used to detect brain tumors. It helps to analyze soft tissue formations enabling the identification and characterization of tumors. However, the interpretation of the scans is time consuming and requires a highly specialized skill set. Also, the more sophisticated the data set becomes, the more pronounced the inconsistencies and errors of human judgment become. This is particularly true in the case of different tumor types. Inconsistencies in the diagnosis become increasingly pronounced in under-resourced settings, where the availability of radiologists is a key constraint.

To address the challenges and limitations, the automation of the interpretation of medical images is increasingly incorporating Artificial Intelligence (AI) technologies. In particular, deep-learning AI systems automate mundane tasks, improve accuracy, and provide consistent evaluations of different cases by streamlining the diagnostic process. They provide radiologists with advanced decision support systems by helping identify patterns that are often missed in the manual assessment, thus speeding the process dramatically. In the specific domain of brain tumor analysis, AI-based models have demonstrated the ability

to achieve diagnostic accuracies that rival, and in some cases surpass, human experts. This makes AI a transformative technology in the field of medical diagnostics.

1.1 Basic Functionalities of Brain Tumor Classification Systems

Every brain tumor classification system is based on three fundamental components: image acquisition, feature extraction, and classification. The process begins with the MRI brain image acquisition. Magnetic Resonance Imaging is the most common and preferred method of imaging due to the non-invasive approach, no exposure to ionizing radiation, and the ability to create high-resolution images of soft tissues. MRI also has the multiple imaging sequences, such as T1 weighted, T2 weighted, FLAIR, and contrast enhanced scans, which provide additional complementary information pertinent to overall brain anatomy and pathology. These images are utilized as the key input and foundational the subsequent classification pipeline stages.

The second stage, feature extraction, involves identifying and isolating meaningful characteristics from MRI scans that can distinguish tumor tissues from healthy brain structures. Traditionally, this process was based on handcrafted features, where domain experts manually designed descriptors to capture low level characteristics such as edges, intensity changes, and textural variations, as well as high-level features such as shapes, symmetry, and spatial arrangements of tissues. While these approaches could provide clinically interpretable features, they were heavily dependent on the expertise of radiologists and researchers, making the process subjective and prone to inconsistencies. Furthermore, handcrafted features often lacked the ability to fully represent the wide variability in tumor appearance caused by differences in tumor grade, size, location, or imaging conditions. This limitation reduced both the efficiency and the generalizability of traditional machine learning (ML) based classification systems.

The final stage is classification, where extracted features are analyzed to assign the MRI scan to a specific category, such as glioma, meningioma, or pituitary tumor. In classical ML methods, classification relied on algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (KNN), or Random Forests, which required carefully engineered input

features to function effectively. However, the overall performance of these systems was restricted by the quality of the handcrafted features provided during the extraction stage.

Deep learning (DL), and more specifically Convolutional Neural Networks (CNNs), has changed feature extraction completely. There is no more need for manual feature engineering, as CNNs learn straight from unprocessed data. CNNs extract data features automatically through multiple convolutional layers. The first few layers recognize primitive features like edges and gradients. The deeper layers capture more complex and abstract representations specific to the tumor. This design structure lets CNNs learn all levels of contextual data in the MRI scans and improves classification accuracy. The use of CNNs in brain tumor classification systems has streamlined the classification process and improved the CNNs classification method automated decision support systems in clinical practice.

1.2 State-of-the-Art Models and Challenges

Over the past decade, there has been a significant increase in the use of various CNN architectures for tasks specific to medical image classification. CNNs such as VGG16, ResNet50 and EfficientNetB3 have been noted for their success in the classification of brain tumors.

VGG16 is a deep CNN architecture that has become popular because of the success of its DL architectures for image recognition. With 16 layers and more than 65 million parameters to analyze, it has the capability to identify features of images in greater detail. However, its large model size and computational requirements are quite challenging for deployment in real time or low resource clinical settings.

ResNet50 resolves the problem of vanishing gradients in significantly deep networks by adding residual connections and permits the training of extremely deep models with great ease. It demonstrates strong performance in tumor classification while reducing computational complexity compared to VGG16. Nevertheless, it still requires significant memory and processing power.

EfficientNetB3 proposes a new method of compound scaling that concurrently improves depth, width and resolution of the network. It achieves similar levels of accuracy as VGG16 and ResNet50 while requiring fewer parameters, making it a more efficient use of compute resources. However, in the context of brain tumor classification, EfficientNetB3 sometimes

lacks the ability to discriminate among different subtypes of the tumor, and accordingly it has lower accuracy for some use cases.

Despite the progress achieved by these state-of-the-art models, one of the ongoing challenges in model development is to maximize diagnostic accuracy, while still maintaining low compute costs. Large networks, while they have performance advantages, may not be deployable in areas where resources are limited for example rural hospitals, onsite mobile MRI, or device deployments where embedded diagnostics may be located. Such challenges create an obvious gap in research that needs to be addressed. Developing models which are lightweight and accurate is essential so that they can be accessible and useful across varying healthcare delivery settings.

1.3 Objectives and Scope of the Study

This study aims to fill the gap by evaluating the performance of VGG16, ResNet50, and EfficientNetB3 on brain MRI datasets, and also developing a new hybrid deep learning model. This hybrid model combines parts of all three architectures, taking advantage of their strengths while lessening their weaknesses. In particular, the model aims for high classification accuracy while also minimizing the parameters, and storage size, allowing for real world clinical implementation.

This research goes beyond just accuracy assessment by also measuring other evaluation metrics such as precision, recall, F1-score, and the confusion matrices, all of which aid in understanding the model's reliability for the different tumor classes. These metrics offer a comprehensive analysis of the state-of-the-art models and the proposed hybrid model.

This study addresses the need for practical and efficient solutions in brain tumor classification, contributing to AI-based medical diagnostics. The outcomes of this research are particularly important for advancing diagnostic capabilities in low resource settings, which improves access to timely and accurate brain tumor diagnosis.

1.4 Background and Motivation

Brain tumors are abnormal cell growths in the brain, or surrounding the brain. Tumors are generally classified as either primary, which arise from brain tissue, or secondary, which metastasize from an additional tumor in another part of the body such as the lungs, breasts, or kidneys [1]. Some examples of primary brain tumors include gliomas, meningiomas, and pituitary tumors [2]. For example, gliomas arise from glial cells and, thus, are the most common primary malignant brain tumor in adults. Gliomas are classified into subtypes like astrocytomas, oligodendrogliomas, and glioblastomas. Meningiomas arise from the meningeal layers, which are the protective layers surrounding the brain and spinal cord, and often are benign but can create pressure against the brain and spinal cord [2][5]. Pituitary tumors arise from the pituitary gland and often disrupt endocrine function [2].

It was estimated that there would be approximately 2,001,140 new cases of cancer and 611,720 cancer-related deaths in the United States in 2024[3]. Approximately 18,330 individuals (10,170 men and 8,160 women) are anticipated to die from brain and spinal cord tumors in the United States in 2025 [4]. The global incidence of brain tumors is heterogeneous by tumor type, age, and geographical region. It is estimated that there are approximately 308,000 new cases of brain and central nervous system tumors each year throughout the world, with the highest incidence found in the 45–65 age range [6]. Mortality rates are variable based on the tumor's malignancy, location, and the treatment options available including surgical resection, radiation therapy, and chemotherapy [7].

MRI is the most widely used non-invasive imaging modality for brain tumor detection, diagnosis, and treatment planning [8]. Different MRI sequences such as T1-weighted, T2-weighted, FLAIR, and contrast-enhanced scans provide complementary anatomical and pathological information, allowing clinicians to assess tumor types, grades, and related regions including necrosis, edema, and active tumor tissue. Traditional diagnostic approaches rely heavily on the expertise of radiologists, who manually analyze multiple sequences to identify abnormal tissue characteristics and classify tumors based on morphology, signal intensity, and anatomical location [9]. Although this manual evaluation is clinically effective, it requires significant expertise, is time consuming, and is prone to inter observer variability. Moreover, measuring tumor volume or delineating precise boundaries can be challenging with purely manual methods [9]. These limitations have motivated the development of computer assisted and automated techniques. Advances in

image processing, ML, and more recently deep DL, have significantly enhanced the potential for automated tumor detection, segmentation, and classification. ML and DL based methods improve diagnostic accuracy, efficiency, and consistency by learning discriminative features directly from imaging data, reducing subjectivity, and supporting clinical decision-making [10][11]. Recent surveys highlight that DL architectures, particularly CNNs, have achieved state-of-the-art performance in brain tumor classification and segmentation tasks, demonstrating their ability to outperform traditional handcrafted feature based approaches and providing robust generalization across datasets [12].

ML approaches, such as SVMs [13], are widely applied in brain tumor analysis due to their effectiveness in handling high-dimensional feature spaces and robustness in binary classification problems. Similarly, Random Forests (RFs) [14], which combine multiple decision trees through bagging and random feature selection, are valued for their stability, interpretability, and ability to reduce overfitting in medical classification tasks. However, both SVMs and RFs rely heavily on manually engineered features, which require domain expertise for extraction and selection. While this allows clear interpretability for clinical applications, it limits scalability and generalization when imaging data become more complex.

In contrast, DL methods, particularly CNNs [15], have transformed medical image analysis by enabling automatic feature extraction from raw imaging data without manual intervention. CNNs hierarchically learn local and global features, capturing spatial, structural, and intensity-based variations in MRI scans, which significantly enhances tumor classification and segmentation performance. State-of-the-art CNN architectures, such as VGG16 [16], EfficientNet [17], ResNet50 [18], InceptionNet [19], and DenseNet [20], have been widely employed in brain tumor studies. VGG16 introduced very deep stacked convolutional layers that improved feature representation, though at the cost of high computational requirements [16]. EfficientNet optimized network scaling in depth, width, and resolution, providing strong accuracy with reduced computational demand [17]. ResNet50 addressed the problem of vanishing gradients through residual connections, enabling effective training of deeper networks [18]. InceptionNet introduced multi-scale feature extraction within the same layer using parallel convolutional filters [19], while DenseNet connected each layer to all subsequent layers, facilitating better feature reuse and improved gradient flow [20]. Collectively, these architectures automate feature learning and

improve classification accuracy, but their performance often comes at the expense of large memory requirements and computational complexity.

To mitigate these challenges, transfer learning [21] has emerged as an effective strategy, where pre-trained models on large-scale datasets (e.g., ImageNet) are fine-tuned for brain tumor classification. This approach reduces training time, requires fewer annotated medical images, and often achieves strong performance even on relatively small MRI datasets. Furthermore, hybrid learning models [22] have recently gained attention by combining CNN based local feature extraction with additional mechanisms, such as global context modeling or ensemble learning, to improve segmentation accuracy and classification robustness. These hybrid methods address the limitations of individual models and demonstrate promising performance in balancing accuracy, computational efficiency, and generalization ability across diverse MRI datasets. Nevertheless, most existing architectures rely on increasingly deeper or more complex structures, which can be computationally demanding, underscoring the need for optimized approaches that balance accuracy with resource efficiency.

El-Dahshan et al. [23] introduced a hybrid ML approach combining discrete wavelet transform (DWT) to extract features. The refined features were classified using a feedforward neural network (FNN), achieving an accuracy of 98.6% on brain MRI images. In recent years, DL has surpassed traditional ML techniques in terms of performance and applicability. Rehman et al. [24] studied AlexNet, GoogLeNet, and VGGNet in comparison with a smaller MRI dataset, and concluded that VGG16 yields the highest accuracy of 98.69% when fine-tuned. Although these models offered better prediction, many of them introduced additional complexity and resource requirements, limiting their practicality in real-time or resource-constrained settings. Liu et al. [25] developed a custom model called MobileNet-BT, derived from the MobileNetV2 architecture, where all layers were tuned to capture more specific features of brain tumor MRI images. The MobileNet-BT model surpassed many other pre-trained models including VGG16, ResNet-18, and EfficientNet-B0, achieving an accuracy of 99.24%. But the model's size is about 14MB, with 2.2 million trainable parameters,

Özkaraca et al. [26] designed a dense CNN model for the multiclass classification of brain tumors using MRI images. It accomplished an accuracy of 98.13% in identifying gliomas, meningiomas, and pituitary tumors. The research highlighted the complex feature extraction potential of densely connected convolutional layers for MR images and demonstrated the

potential for deep learning's applicability due to its flexibility to various types of tumors and generalization capabilities. In the same way, Irmak [27] developed an optimized deep CNN model devoted to brain tumor classification. Their approach attained a classification accuracy of 96.56% for the diverse tumor types by tumor type through the optimization of the layers' order and the elimination of excess design elements. This work showed that fine-tuned, redundant architecture designs could still deliver high results while optimized for resource usage, a crucial requirement for real-time diagnosis systems.

Other researchers have also sought to improve classification performance by modifying or combining existing architectures. A hybrid algorithm for brain tumor detection that utilizes both DL and ML was presented by Bang et al.[28]. Their model employs feature extraction using VGG16, and for classification, a SVM classifier is used. Using a dataset of over 7000 MRI images, their model achieved 91% accuracy, showcasing the power of ML and DL integration in medical imaging. Acclaimed by Çinar and Yildirim [29] state that they created a modified architecture of ResNet50 by removing the last five layers and placing eight additional layers to enhance feature extraction. This hybrid CNN model outperformed others achieving 97.2% classification accuracy on brain MRI images. Their studies along with others propagate the idea of tuning frameworks for improving classification of brain tumors in patients. The study conducted by Zulfiqar et al. [30] analyzed the implementation of EfficientNet (B0 - B4) models for the multi-class categorization of tumors into gliomas, meningiomas, and pituitary tumors. They showed that EfficientNetB2 gave the highest accuracy of 98.86% when leveraging a transfer learning approach with fine-tuning on the CE-MRI Figshare Brain Tumor Dataset. However, this model's size was approximately 90 MB, indicating a trade-off between accuracy and computational efficiency.

These studies collectively showcase progress of sophisticated DL brain tumor classifiers while noting the need for model flexibility alongside accuracy parameters in computing resources.The integration of DL and ML techniques, as well as the modification of existing architectures, continue to play a pivotal role in enhancing diagnostic accuracy and efficiency in medical imaging.

However, most of the existing models that are widely used for classification tasks suffer from a fundamental trade-off between accuracy and efficiency. In general, models that achieve higher accuracy tend to come with a much larger number of parameters, leading to significantly bigger model sizes and higher computational costs. On the other hand, models

with fewer parameters are often more efficient but may compromise performance in terms of accuracy.

This inherent trade-off highlights the need for approaches that can strike a balance between these two aspects, ensuring that models are not only accurate but also lightweight and scalable. Motivated by this challenge, we have developed a model architecture that addresses these limitations by reducing the number of parameters and compressing the overall model size, while at the same time achieving improved accuracy compared to existing alternatives. This
makes our approach both efficient and effective, offering a practical solution for real-world applications where computational resources are limited but high performance is still required.

CHAPTER 2

OVERVIEW OF BRAIN TUMOR CLASSIFICATION

Given the rising global incidence of brain tumors and their significant impact on public health, the importance of timely and accurate diagnosis cannot be overstated. Traditional diagnostic procedures rely heavily on radiologists, who must analyze large volumes of MRI scans with careful attention to subtle textural and structural differences. This manual process, while clinically effective, is both time-consuming and susceptible to inter-observer variability, which can compromise diagnostic consistency. Consequently, there is an urgent need for robust automated classification systems that can provide reliable support to clinical decision-making. Addressing this need, our work proposes the development of an efficient yet highly accurate deep learning-based model, which specifically aims to overcome the common trade-off between performance and computational cost. By striking this balance, the model aspires to deliver practical applicability in real-world clinical environments where computational resources may be limited.

2.1 Introduction

This study centers on the design and implementation of a hybrid deep learning model that integrates complementary strengths from three widely used convolutional neural network (CNN) architectures: VGG16, ResNet50, and EfficientNetB3. Each of these models contributes unique advantages—VGG16 is effective at capturing fine-grained spatial features through its simple but deep layered structure, ResNet50 introduces residual connections that enable stable training of deeper networks without degradation, and EfficientNetB3 achieves state-of-the-art performance by scaling network depth, width, and resolution in a balanced and computationally efficient manner. By combining these salient features, the proposed hybrid model is designed to maintain a compact architecture that avoids unnecessary computational complexity while still achieving high classification accuracy. The ultimate goal of this work is to present a model that achieves a strong balance between diagnostic accuracy, computational efficiency, and scalability, making it both academically significant and practically valuable for deployment in medical imaging applications.

2.2 Dataset and Preprocessing

2.2.1 Dataset

For this study, a publicly available dataset of brain MRI images was sourced from Kaggle [31], consisting of a total of 7,023 images. This dataset was originally compiled from three different sources: Figshare, the SARTAJ dataset, and the Br35H collection, each of which contributes to the diversity and reliability of the combined dataset. By integrating images from multiple sources, the dataset ensures greater variability in imaging conditions, patient demographics, and tumor characteristics, which is particularly important for building a robust and generalizable classification system. The dataset is organized into four distinct classes: glioma, meningioma, pituitary tumors, and no tumor. These categories represent the most common primary brain tumors as well as healthy cases, thereby providing a comprehensive basis for multi-class classification tasks in medical imaging.

To prepare the dataset for model training and evaluation, the images were reorganized into three non-overlapping subsets: training, validation, and testing, following a 70:15:15 ratio. This partitioning strategy ensures that the majority of the data is used for training, while adequate portions are reserved for validation and testing to assess model generalization and prevent overfitting. The training set was used for model parameter optimization, the validation set was employed for fine-tuning hyperparameters and monitoring learning progress, and the testing set was exclusively reserved for final performance evaluation on previously unseen data. This systematic division enhances the credibility of the experimental results by providing a fair and unbiased measure of the model's predictive capability.

Furthermore, the dataset was carefully examined to ensure that each subset preserved the class balance across the four categories, thereby avoiding skewed training that might bias the model toward majority classes. Such balanced distribution is crucial in medical imaging studies, where misclassification of minority classes can lead to serious diagnostic implications. The detailed distribution of images across categories and partitions is presented in Table 2.1, which provides a clear overview of how the dataset was structured for this research.

Table **2.1** MRI IMAGE NUMBERS IN THE DATASET

Data	Glioma	Meningioma	No tumor	Pituitary	Total
Training data	1135	1151	1400	1230	4916
Testing data	243	247	300	264	1054
Validation data	243	247	300	263	1053

2.2.2 Preprocessing and Augmentation

For model training, all images were first resized to 128 × 128 pixels to ensure a consistent input format across the dataset. CNN architectures require fixed input dimensions for stable convolutional and pooling operations, and without resizing, feature maps would become irregular. The chosen resolution of 128 × 128 provided a practical balance between computational efficiency and information retention, allowing the model to preserve essential tumor features while avoiding excessive memory and training time requirements.

Since the dataset originated from multiple repositories with varying formats, all images were also converted into RGB color space. This step ensured compatibility with CNN architectures, particularly pre-trained models built for three-channel inputs such as those on ImageNet. While MRI scans are inherently grayscale, replicating the single channel across three dimensions preserved structural information and enabled seamless integration with transfer learning frameworks. This standardization improved dataset uniformity and ensured smooth operation of the hybrid deep learning pipeline.

In addition, a custom preprocessing routine was applied to normalize pixel intensity values and correct inconsistencies introduced by the heterogeneous nature of the original sources (Figshare, SARTAJ, and Br35H). This included rescaling pixel values to a standardized range between 0 and 1, thereby reducing sensitivity to illumination variations and improving the convergence of the learning process. These steps not only improved the quality of the dataset but also minimized noise, ensuring that the models focused on clinically relevant features rather than imaging artifacts.

To further strengthen the generalization capability of the model, data augmentation techniques were employed during the training phase. Augmentation artificially increases dataset diversity by applying controlled transformations to the original images, thereby helping the model learn robust features that are invariant to orientation, scale, and lighting conditions. The transformations included slight random rotations (±15 degrees), horizontal and vertical shifts (up to 10%), brightness adjustments, and zooming (up to 20%). These operations mimic the variability commonly encountered in real-world MRI scans, such as differences in patient positioning, machine calibration, or acquisition protocols. As a result, the augmented dataset better represented the heterogeneity of clinical imaging scenarios, reducing the risk of overfitting and enhancing the model's robustness.

For the testing phase, only rescaling was performed, ensuring that no artificial transformations altered the original diagnostic features of the images. This step was critical to maintaining objectivity and fairness in model evaluation, since augmented test data could bias the assessment of classification performance.

By adopting this preprocessing and augmentation pipeline, both the consistency and diversity of the data were improved. Standardization guaranteed that all images followed a uniform structure suitable for the hybrid CNN framework, while augmentation enriched the training data, allowing the model to effectively learn complex and subtle tumor-specific patterns. Figure 2.1 illustrates examples of MRI brain tumor images after augmentation, showcasing how these transformations expand the variability of the dataset while preserving essential structural details needed for accurate tumor classification.

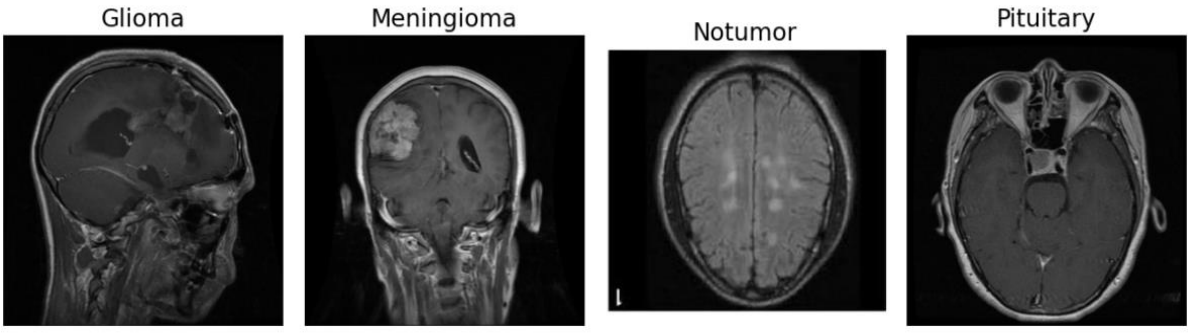


Figure 2.1: Augmented MRI images

2.3 Model Architecture

VGG16, ResNet50, and EfficientNetB3 are the state-of-the-art models which are widely used as base models and also independently for deep learning frameworks. VGG16 is widely used for its simple structure and stacked convolutional pooling layers which can help a lot in feature extraction. ResNet50, though has a deep network, introduces the residual or skip connection structure which helps to mitigate the vanishing gradient problem for deeper architectures. EfficientNetB3 uses the compound scaling method to balance both factors: accuracy and efficiency. By combining the complementary architectural strengths of the selected three models, the proposed hybrid model aims to deliver improved feature representation, compactness, and competitive classification accuracy

The architectures of the three base models: VGG16, ResNet50 and EfficientNetB3 and reasons for choosing them for architectural inspiration are explained below:

2.3.1 Architecture of VGG16

VGG16 has been of the most widely used convolutional neural network (CNN) architectures. The architecture is characterized by its simplicity, uniform design philosophy, and strong performance on large-scale visual recognition tasks such as ImageNet. Unlike earlier models that employed large convolutional filters or complex multi-branch structures, VGG16 demonstrated that stacking small convolutional filters in depth could yield significant representational power and performance improvements.

The VGG16 network consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers. It employs 3×3 convolutional filters throughout, which are the smallest kernels capable of capturing spatial context, applied with a stride of 1 and appropriate padding to maintain spatial resolution. This consistent use of small filters allows the model to build complex hierarchical feature representations by combining multiple layers, effectively expanding the receptive field while maintaining computational efficiency.

The network accepts an input image of size 224×224×3 and processes it through a sequence of convolutional blocks. Each block comprises several convolutional layers followed by a max pooling layer with a 2×2 window and a stride of 2. The pooling layers

progressively reduce the spatial resolution while preserving the most salient information, enabling the network to learn spatial hierarchies that evolve from low-level edge and texture detection in early layers to high-level semantic feature extraction in deeper layers.

The overall configuration of VGG16 can be summarized as follows:

- Block 1: Two 3×3 convolutional layers with 64 filters, followed by a 2×2 max pooling layer.
- Block 2: Two 3×3 convolutional layers with 128 filters, followed by max pooling.
- Block 3: Three 3×3 convolutional layers with 256 filters, followed by max pooling.
- Block 4: Three 3×3 convolutional layers with 512 filters, followed by max pooling.
- Block 5: Three 3×3 convolutional layers with 512 filters, followed by max pooling.

Following the convolutional and pooling blocks, the network transitions into three fully connected layers: two dense layers containing 4096 neurons each, and a final softmax layer that outputs class probabilities. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity and improves convergence by mitigating the vanishing gradient problem.

Unlike more recent architectures such as ResNet or DenseNet, VGG16 does not include skip or residual connections; instead, information propagates sequentially through the network. Despite this, its depth and regular structure enable it to learn highly discriminative features effectively. The uniform design approach—using the same small convolutional filter size throughout—makes VGG16 both elegant and modular, allowing it to serve as a foundational model for various applications, including object detection, image segmentation, and transfer learning.

The stacked 3×3 convolutional layers enable the extraction of fine spatial details, while successive pooling operations help reduce the feature map dimensions without losing crucial information. The fully connected layers subsequently interpret the learned hierarchical features and map them to specific output categories. Although VGG16 is computationally demanding due to its large number of parameters (approximately 138 million), it remains a benchmark model for visual recognition tasks because of its balance between architectural simplicity and strong feature extraction capabilities.

Overall, VGG16 exemplifies how network depth, consistent architectural design, and the use of small convolutional filters can together achieve high classification accuracy and robust visual representation learning. The sequential arrangement of convolutional and pooling layers, as illustrated in Figure 2.2, demonstrates the progressive transition from local spatial details to abstract global representations, establishing VGG16 as a cornerstone in the evolution of deep learning-based computer vision models.

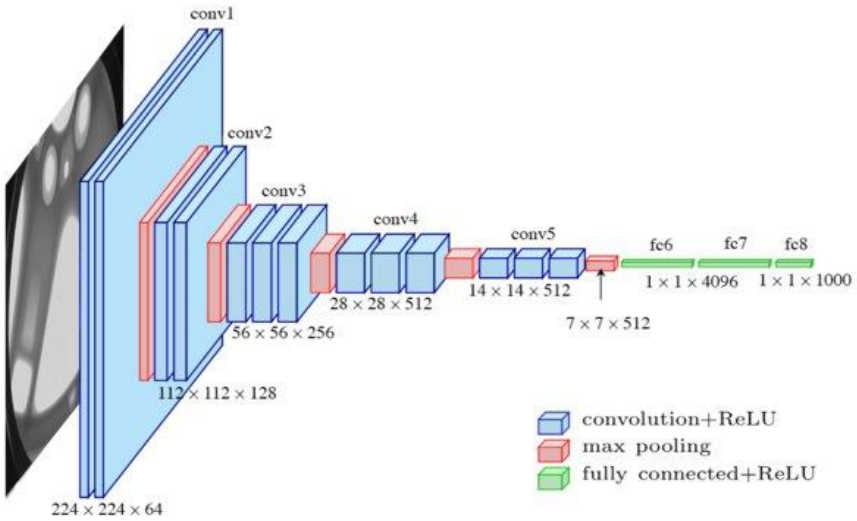


Figure 2.2: The architecture of VGG16 [32]

2.3.2 Architecture of ResNet50

ResNet50, short for Residual Network with 50 layers, is a deep convolutional neural network proposed by He et al. [18] in 2016. It introduced the concept of residual learning, which addressed one of the major challenges in training very deep networks—the vanishing and exploding gradient problems. Traditional deep networks often suffer performance degradation as their depth increases, making optimization increasingly difficult. ResNet50 overcomes this limitation by incorporating skip, or residual, connections that allow the flow of information directly across layers, thus enabling the effective training of extremely deep architectures.

The fundamental building block of ResNet50 is the residual block, which introduces a shortcut connection that bypasses one or more layers. Mathematically, instead of learning an unreferenced mapping H(x), the residual block learns a residual mapping F(x) = H(x) −

x, allowing the original function to be expressed as $H(x) = F(x) + x$. This reformulation significantly simplifies optimization, as it becomes easier for the model to learn the residual corrections rather than the complete transformation. These shortcut connections ensure that gradients can propagate backward without attenuation, thereby preventing vanishing gradients and enabling stable training even in networks with hundreds of layers.

ResNet50 follows a deep architecture consisting of 50 layers organized into an initial convolutional and pooling stage, followed by a series of residual blocks. The model begins with a 7×7 convolutional layer with 64 filters and a stride of 2, followed by a 3×3 max pooling layer. This is succeeded by four stages of residual blocks, each containing a series of bottleneck structures that reduce computational cost while maintaining representational power. The structure of these stages can be summarized as follows:

- Conv1: 7×7 convolution, 64 filters, stride 2, followed by max pooling.
- Conv2_x: Three residual blocks with 1×1, 3×3, and 1×1 convolutions (64, 64, 256 filters respectively).
- Conv3_x: Four residual blocks with (128, 128, 512) filters.
- Conv4_x: Six residual blocks with (256, 256, 1024) filters.
- Conv5_x: Three residual blocks with (512, 512, 2048) filters.

After these convolutional stages, a global average pooling layer aggregates spatial information, and a fully connected layer with a softmax activation function produces the final classification output. Batch normalization is applied after each convolutional layer, and Rectified Linear Unit (ReLU) activation functions are used to introduce non-linearity and accelerate convergence.

The use of bottleneck residual blocks in ResNet50 ensures computational efficiency. Each block compresses the input feature dimensions through a 1×1 convolution before applying the 3×3 convolution and then restores the original dimensions through another 1×1 convolution. This not only reduces the number of parameters but also allows deeper networks to be trained effectively without performance degradation.

The residual connections act as direct pathways for gradients and information, enabling the network to maintain high accuracy even as the number of layers increases. Consequently, ResNet50 achieves both depth and stability, offering excellent feature extraction

capabilities and generalization performance across a wide range of computer vision tasks, including image classification, object detection, and segmentation.

Overall, ResNet50 exemplifies the success of residual learning in overcoming the limitations of traditional deep CNNs. Its innovative architecture allows effective gradient flow, robust feature propagation, and ease of optimization, all while maintaining remarkable depth. Figure 2.3 illustrates the layer-wise structure of ResNet50, highlighting how the integration of residual blocks supports deep learning without degradation in accuracy or convergence.

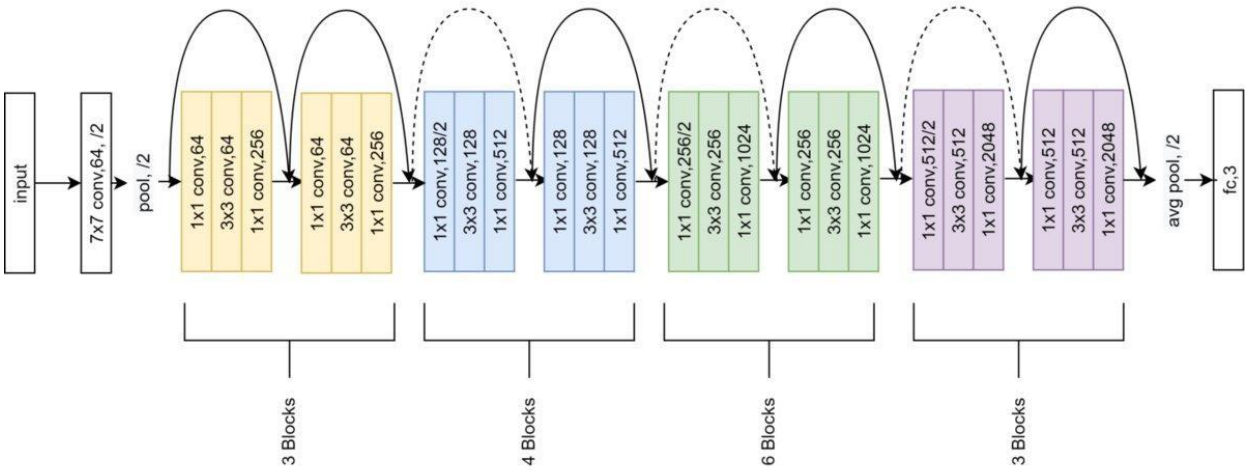


Figure 2.3: An illustration of ResNet-50 layers architecture [33]

2.3.3 Architecture of EfficientNetB3

EfficientNet-B3 is a part of the EfficientNet family of convolutional neural networks developed by Tan and Le in 2019 [17], which introduced a novel approach to model scaling through a technique known as compound scaling. Traditional deep learning models typically improve performance by increasing either depth, width, or input resolution independently, which often leads to diminishing returns or inefficiencies in computation. EfficientNet addresses this limitation by uniformly and systematically scaling all three dimensions depth, width, and resolution using a set of fixed scaling coefficients. This balanced scaling strategy allows the model to achieve significantly better accuracy and efficiency compared to previous architectures of similar complexity.

The backbone of EfficientNet-B3 is constructed upon the Mobile Inverted Bottleneck Convolution (MBConv) blocks, which employ depth wise separable convolutions to drastically reduce computational cost while maintaining strong representational power. In a depthwise convolution, a single convolutional filter is applied to each input channel separately, thereby extracting intra-channel spatial features. This is followed by a pointwise (1×1) convolution that combines these features across channels, capturing inter-channel dependencies. Together, these operations enable EfficientNet-B3 to maintain a high level of expressiveness with far fewer parameters compared to conventional convolutional layers.

A distinctive component integrated within each MBConv block is the Squeeze-and-Excitation (SE) mechanism, which introduces an adaptive channel attention mechanism to recalibrate feature responses. In the squeeze phase, global average pooling is applied to each channel, summarizing its spatial information into a single representative value that reflects its overall importance. The excitation phase then passes these aggregated values through fully connected layers followed by a sigmoid activation function, generating channel-wise weights. These weights are used to selectively emphasize informative channels while suppressing less relevant ones, thereby improving the network's ability to focus on critical features.

The EfficientNet-B3 architecture begins with a standard 3×3 convolutional stem, followed by a series of MBConv blocks with varying expansion ratios, filter sizes, and numbers of layers. These are carefully scaled versions of the baseline EfficientNet-B0 model, adjusted according to the compound scaling coefficients derived through a grid search optimization. The final layers include a 1×1 convolution for feature aggregation, a global average pooling layer, dropout for regularization, and a fully connected layer with softmax activation for classification. Batch normalization and Swish (or SiLU) activation functions are applied throughout the network to enhance non-linearity and improve training stability.

EfficientNet-B3 achieves an optimal balance between accuracy and computational efficiency by leveraging its compound scaling and lightweight building blocks. Despite having significantly fewer parameters and FLOPs compared to earlier deep architectures, EfficientNet-B3 consistently delivers superior performance across a wide range of computer vision tasks.

In summary, EfficientNet-B3 exemplifies the effectiveness of compound scaling and channel attention mechanisms in achieving high accuracy with minimal computational resources. Figure 2.4 provides an overview of the EfficientNet-B3 architecture, illustrating how compound scaling and squeeze-and-excitation modules are integrated to enhance model efficiency and feature representation.

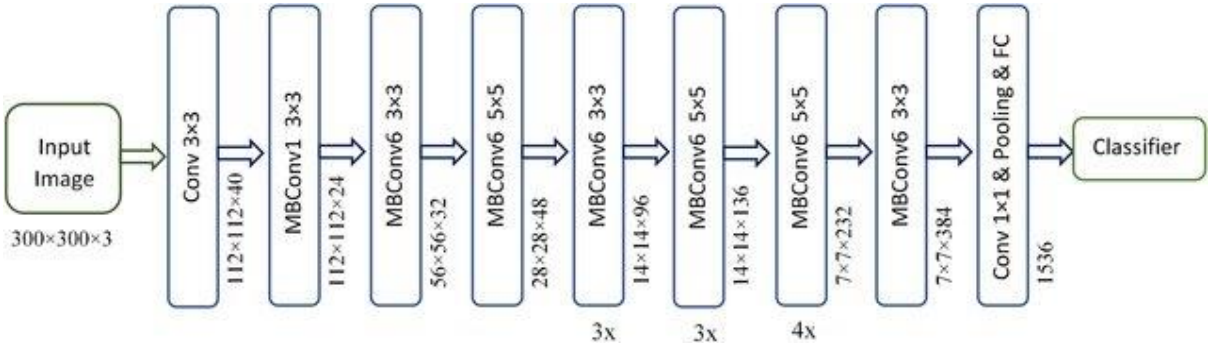


Figure 2.4: Schematic representation of EfficientNet-B3 [34]

Figure 2.5 displays the implementation layout of the VGG16, ResNet50, and EfficientNetB3 models, forming the foundation for comparison and hybridization in the proposed classification framework.

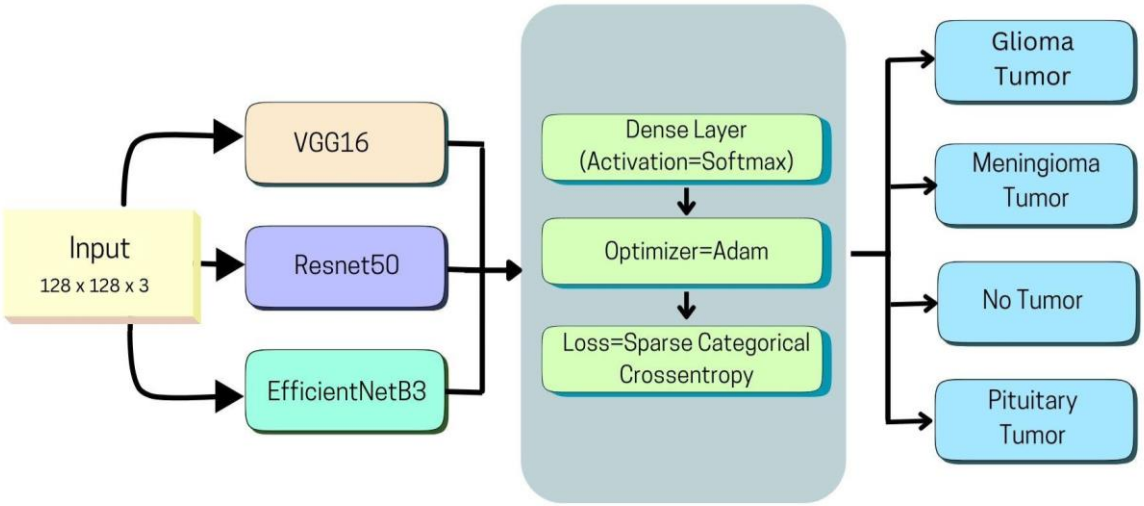


Figure 2.5: Implementation of VGG16, ResNet50 and EfficientNet models

VGG16 contributes to our design by offering a structured and hierarchical approach to feature extraction. Its stacked convolutional and pooling layers allow the network to capture low-level features such as edges and corners in the early layers, and progressively

build more complex textures and patterns in deeper layers. This provides a solid foundation for extracting meaningful representations from the images.

ResNet50 enhances the learning capability by introducing residual connections, which address the vanishing gradient problem and allow the network to be trained much deeper. These skip connections enable the model to retain important low-level information while learning higher-level abstract features, improving both convergence speed and overall performance.

EfficientNetB3 contributes compactness and efficiency to the architecture. It combines depthwise and pointwise convolutions with advanced attention mechanisms, including squeeze-and-excitation blocks, which allow the network to focus selectively on the most informative features. This ensures that the model remains lightweight while maintaining high representational power.

By integrating these three architectures VGG16's structured feature extraction, ResNet50's depth and stability, and EfficientNetB3's efficiency and attention mechanisms we form the backbone of our hybrid model. Leveraging the strengths of each, the proposed design aims to create a lightweight network that delivers high classification accuracy with low computational cost, making it both practical and effective for real-world MRI image analysis.

2.3.4 Architecture of Proposed Model

The proposed hybrid model integrates key architectural elements inspired by VGG16, ResNet50, and EfficientNetB3, aiming to achieve effective feature extraction and accurate classification for brain tumor analysis. The model begins with a stem layer that utilizes convolutional kernels to capture low-level features from MRI inputs of size (128, 128, 3). This initial layer sets the foundation by preserving spatial details and reducing dimensionality. Figure 2.6 illustrates the architecture of hybrid model.

The model architecture is organized into key functional components: repeated convolutional blocks forming the head, residual structures that maintain information flow across layers while enhancing feature learning; and a bottleneck incorporating depthwise and pointwise convolutions to capture complex spatial and inter-channel relationships

efficiently. The inclusion of squeeze-and-excitation (SE) blocks further refines feature maps by adaptively focusing on the most relevant information.

The architecture concludes with a global average pooling layer, dense layers, and batch normalization to aggregate features and stabilize activations.

Stem Layer

The hybrid model begins with a stem layer that applies a 3×3 convolution with 64 filters, followed by batch normalization and max pooling. This initial layer extracts basic visual features such as edges, corners, and gradients, which form the foundation for higher-level feature extraction. Batch normalization stabilizes the activations and accelerates convergence, while max pooling reduces the spatial dimensions of the feature maps, making the subsequent layers computationally efficient and less prone to overfitting.

Stacked Convolution Layers

The next stage consists of stacked convolutional layers with 128 filters each, interleaved with batch normalization and followed by max pooling. Each convolutional layer progressively captures more complex patterns by combining the lower-level features from previous layers. The network first identifies simple textures and shapes, and as the layers deepen, it forms richer, more abstract representations. Max pooling continues to reduce spatial size while retaining the most important information, allowing the network to focus on relevant features.

Residual Block

The residual block introduces a skip connection to improve training stability and gradient flow. A 1×1 convolution on the input creates a skip pathway that aligns dimensions for element-wise addition. The main path applies a 3×3 convolution followed by batch normalization, and its output is added to the skip connection. A ReLU activation then introduces non-linearity. This structure allows the model to preserve low-level details while learning higher-level features and mitigates the vanishing gradient problem, enabling deeper networks to be trained more effectively.

Depth wise Convolution

Depth wise convolution applies a spatial filter independently to each input channel. By treating each channel separately, the network can capture fine-grained local patterns without mixing information across channels. This approach reduces the number of

parameters and computational cost compared to standard convolutions, while preserving the ability to learn meaningful spatial features. Depth wise convolution is particularly effective for building lightweight networks that require efficient processing.

Suppose we have an input feature map of size H×W×C where H is height, W is width, and C is the number of channels. For each channel, a separate filter of size K×K is applied. The mathematical formulation of depthwise convolution can be expressed as follows:

$$Y_{h,w,c} = \sum_{i=o}^{K-1} \sum_{j=o}^{K-1} X_{h+i,w+i,c} . F_{i,j,c}$$

Where:

- $X_{h+i,w+i,c}$ is the input pixel at position (h+i,w+j) for channel c
- $F_{i,j,c}$ is the corresponding depthwise filter value
- $Y_{h,w,c}$ is the resulting output pixel

Pointwise Convolution

Pointwise convolution, implemented as a 1×1 convolution, follows depthwise convolution to merge information across channels. Each output channel is computed as a linear combination of all input channels, allowing the network to capture cross-channel correlations. This operation integrates the independently filtered features from the depthwise step into a unified representation, enhancing the richness of the learned features while keeping computational complexity low.

The mathematical formulation of pointwise convolution is given by:

$$Y_{h,w,c\prime} = \sum_{c=o}^{C-1} X_{h,w,c} . F_{c,c\prime}$$

Where:

- $X_{h,w,c}$ is the input feature map at position (h,w) for channel c.
- $F_{c,c\prime}$ is the filter weight connecting input channel c to output channel c′.
- $Y_{h,w,c\prime}$ is the final output feature map value.

Squeeze-and-Excitation (SE) Block

The squeeze-and-excitation (SE) block is a channel attention mechanism that enhances the representational capacity of convolutional neural networks by learning to emphasize informative feature channels and suppress less relevant ones. It allows the network to model inter-channel dependencies explicitly, improving its ability to focus on important features.

The operation of the SE block can be described in three conceptual stages: squeeze, excitation, and recalibration.

In the squeeze stage, the global spatial information of each channel is compressed into a single representative value using global average pooling. This operation summarizes the overall importance of each channel by aggregating spatial information across the entire feature map.

In the excitation stage, the block learns how much importance should be assigned to each channel. This is achieved by passing the squeezed vector through two fully connected layers separated by a ReLU activation, followed by a sigmoid activation to generate normalized channel-wise weights. This mechanism captures nonlinear dependencies between channels and learns how to reweight them based on their relative contribution to the final prediction.

In the recalibration stage, the obtained channel weights are used to scale the original feature maps through channel-wise multiplication. Each channel of the input feature map is multiplied by its corresponding attention coefficient. This operation enhances the most informative channels and suppresses less relevant ones, allowing the network to adaptively refine its feature representations based on contextual importance.

The SE block is lightweight and adds only a small number of extra parameters, making it suitable for integration into various deep architectures such as ResNet, DenseNet, and EfficientNet. By enabling dynamic channel reweighting, it improves the discriminative power of convolutional networks, leading to better feature utilization, enhanced robustness, and improved classification accuracy.

Global Average Pooling and Dense Layers

Global Average Pooling (GAP) is a technique used in convolutional neural networks to reduce the spatial dimensions of feature maps while preserving depth information. Instead of flattening a feature map into a long vector, GAP computes the average value of each feature channel across all spatial locations. This produces a single value per channel, resulting in a vector whose length equals the number of channels. GAP captures the global presence of each feature rather than local activations, providing spatial invariance and reducing the number of parameters compared to fully connected layers. It is particularly useful in attention mechanisms, such as the squeeze-and-excitation block, where it generates channel descriptors that summarize the overall importance of each channel before applying adaptive scaling. By compressing spatial information efficiently, GAP helps the network maintain essential context while minimizing computational complexity.

Dense layers, also known as fully connected layers, are used to learn complex feature combinations and perform high-level reasoning within neural networks. Each neuron in a dense layer is connected to every element of the input vector, allowing the network to model arbitrary relationships between features. Dense layers transform the input vector through a linear combination of all input values followed by a non-linear activation function, enabling the learning of non-linear mappings between inputs and outputs. In classification tasks, dense layers are usually placed near the end of the network to combine high-level features extracted by convolutional layers into decision scores for each class. Within attention mechanisms like the SE block, dense layers first reduce the dimensionality of channel descriptors to capture a compact representation and then restore it while applying a non-linear activation to generate attention weights. The number of trainable parameters in a dense layer depends on the number of input features, the number of neurons, and bias terms, all of which adapt during training to capture complex feature interactions.

Together, Global Average Pooling and dense layers form a powerful combination in modern CNNs. GAP efficiently summarizes spatial information across feature channels, while dense layers process these summaries to learn feature interactions, produce adaptive weights, or make final predictions. In attention mechanisms, they work hand in hand to recalibrate channel importance, enhancing the representational capability and discriminative power of the network.

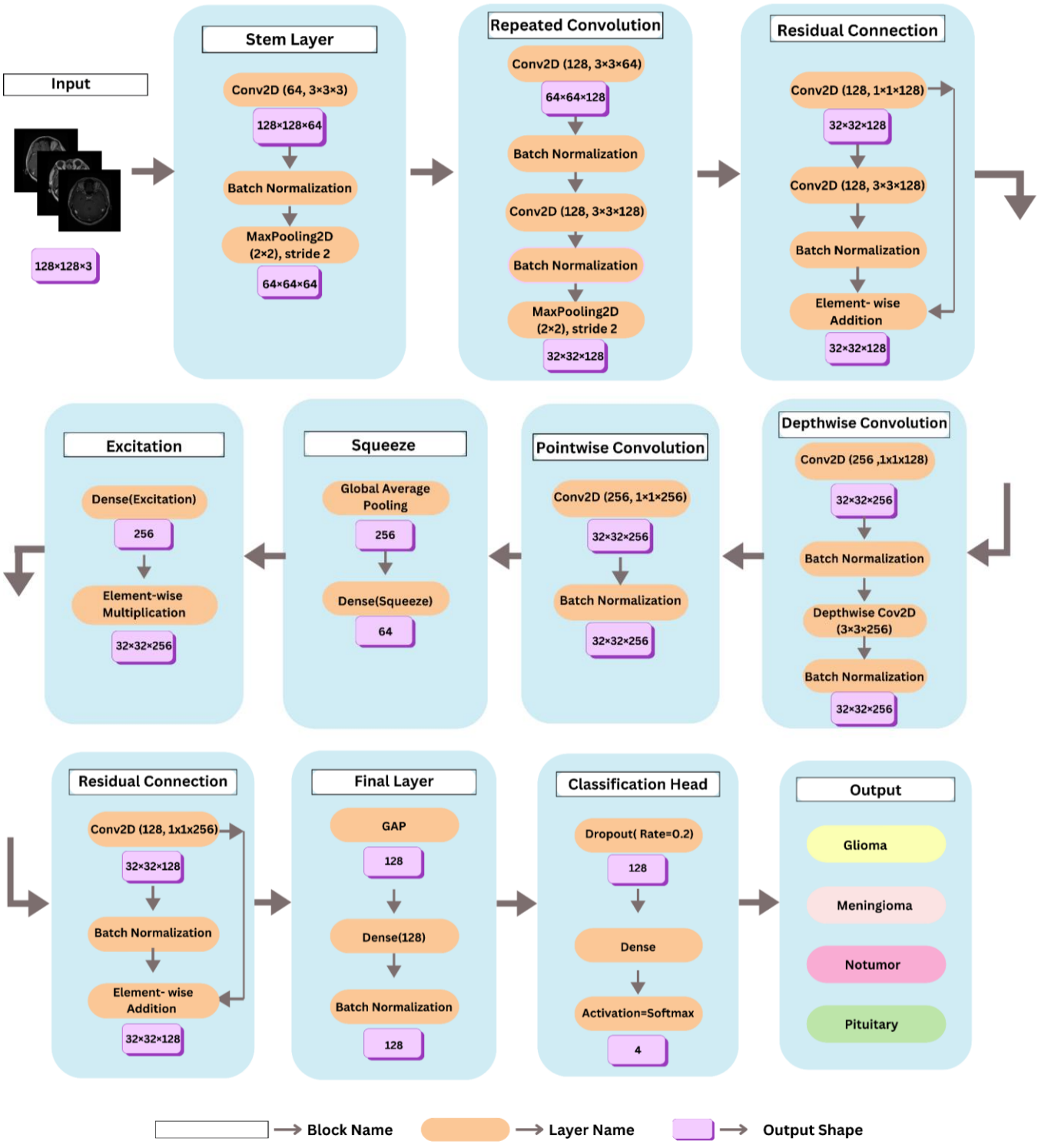


Figure 2.6: Architecture of Hybrid Model

2.3.5 Model Structure and Parameters

Table 2.2 presents the layer-by-layer configuration of the proposed hybrid CNN model, including layer type, output shape, number of parameters, and trainable components. The model consists of multiple convolutional; normalization, pooling, residual, and attention-based layers designed to efficiently extract and refine hierarchical features from the input data.

The architecture begins with an input layer that accepts images of shape (128, 128, 3). This is followed by the first convolutional block (Conv2D_1), which applies 64 filters of size 3×3×3 with a ReLU activation function. The total number of trainable parameters in this layer is calculated as (3×3×3)×64 + 64 = 1,792, where the additional 64 corresponds to the bias terms. A batch normalization layer (BatchNorm_1) follows, normalizing the feature maps across the batch dimension to stabilize training and accelerate convergence, adding 256 trainable parameters ((64×2) + 128). A max-pooling operation (MaxPooling_1) with a 2×2 kernel and stride 2 is then used to down sample the spatial resolution, reducing it from (128, 128, 64) to (64, 64, 64) without introducing trainable parameters.

The second convolutional block (Conv2D_2) applies 128 filters of size 3×3×64 with ReLU activation, significantly increasing the feature depth. The parameters are computed as (3×3×64×128) + 128 = 73,856. This is followed by BatchNorm_2 with 512 parameters to ensure feature normalization. The third convolutional block (Conv2D_3) further refines these representations with 128 filters of size 3×3×128 and ReLU activation, totaling (3×3×128×128) + 128 = 147,584 parameters. BatchNorm_3 again provides feature normalization (512 parameters), followed by MaxPooling_2, which reduces the feature map to (32, 32, 128) without any trainable parameters.

To preserve information flow and support gradient propagation, a residual connection (Residual Conv) with a 1×1 convolution is introduced, having (1×1×128×128) + 128 = 16,512 parameters. This forms the basis for the first residual block, where Conv2D_4 applies another 3×3 convolution (147,584 parameters), followed by BatchNorm_4 (512 parameters). The Add_1 operation merges the input and output of the block through a skip connection, enabling the model to bypass certain transformations and thus mitigate the vanishing gradient problem.

The network then proceeds to the expansion phase, beginning with an Expand Conv layer that projects feature depth from 128 to 256 using 1×1 convolutions with ReLU activation ((1×1×128×256) + 256 = 33,024 parameters). BatchNorm_5 provides normalization (1,024 parameters). The DepthwiseConv layer performs depth wise convolution with a 3×3 kernel, where each filter operates on a single input channel ((3×3×256) + 256 = 2,560 parameters), followed by BatchNorm_6 (1,024 parameters). A Pointwise Conv (1×1×256) combines these outputs, adding (1×1×256×256) + 256 = 65,792 parameters, and BatchNorm_7 again normalizes the activations (1,024 parameters).

The architecture incorporates a Squeeze-and-Excitation (SE) block to adaptively recalibrate channel-wise feature responses. The SE mechanism begins with global average pooling (SE Pooling) to produce a 256-dimensional vector representing global channel statistics. A dense layer (SE Dense_1) reduces this vector to 64 dimensions with ReLU activation, contributing (256×64) + 64 = 16,448 parameters. Another dense layer (SE Dense_2) expands it back to 256 dimensions using a sigmoid activation function ((64×256) + 256 = 16,640 parameters). The output from this layer is used to rescale the original feature maps via a multiplication operation (Multiply), allowing the network to emphasize informative features and suppress less relevant ones.

A subsequent convolutional layer (Conv2D_5) compresses the channel depth from 256 to 128 using a 1×1 convolution with (1×1×256×128) + 128 = 32,896 parameters, followed by BatchNorm_8 (512 parameters). The Add_2 operation forms another skip connection to preserve learned representations, followed by a ReLU activation (ReLU_2) to maintain non-linearity.

The final section of the network transitions from feature extraction to classification. A global average pooling (GAP) layer aggregates spatial features into a 128-dimensional vector, followed by a dense layer (Dense_3) with ReLU activation containing (128×128) + 128 = 16,512 parameters. Finally, a softmax dense layer (Dense_4) outputs class probabilities across four categories, with (128×4) + 4 = 516 trainable parameters.

Overall, the proposed Hybrid CNN model efficiently integrates convolutional, normalization, residual, depth wise-separable, and attention mechanisms to achieve deep feature extraction and robust classification while maintaining computational efficiency.

Table 2.2 LAYER WISE ARCHITECTURE AND PARAMETER DETAILS OF THE PROPOSED MODEL

Layer Name	Type	Output Shape	Parameters (Trainable)	Total Trainable Parameters
Input	Input Layer	(128, 128, 3)	0	0
Conv2D_1	Convolutional (3×3×3), stride 1, **ReLU**	(128, 128, 64)	(3×3×3)×64 + 64	1,792
BatchNorm_1	Batch Normalization	(128, 128, 64)	(64×2) + 128	256
MaxPooling_1	MaxPooling (2×2), stride 2	(64, 64, 64)	0	0
Conv2D_2	Convolutional (3×3×64), stride 1, **ReLU**	(64, 64, 128)	(3×3×64×128) + (128)	73,856
BatchNorm_2	Batch Normalization	(64, 64, 128)	(128×2) + 256	512
Conv2D_3	Convolutional (3×3×128), stride 1, **ReLU**	(64, 64, 128)	(3×3×128×128) + (128)	147,584
BatchNorm_3	Batch Normalization	(64, 64, 128)	(128×2) + 256	512
MaxPooling_2	MaxPooling (2×2), stride 2	(32, 32, 128)	0	0
Residual Conv	Convolutional (1×1×128), stride 1	(32, 32, 128)	(1×1×128×128) + (128)	16,512
Conv2D_4	Convolutional (3×3×128), stride 1, **ReLU**	(32, 32, 128)	(3×3×128×128) + (128)	147,584
BatchNorm_4	Batch Normalization	(32, 32, 128)	(128×2) + 256	512
Add_1	Skip Connection	(32, 32, 128)	0	0
ReLU_1	Activation Layer, **ReLU**	(32, 32, 128)	0	0
Expand Conv	Convolutional (1×1×128), stride 1, **ReLU**	(32, 32, 256)	(1×1×128×256) + (256)	33,024
BatchNorm_5	Batch Normalization	(32, 32, 256)	(256×2) + 512	1,024
DepthwiseConv	Depthwise Conv (3×3×256), stride 1, **ReLU**	(32, 32, 256)	(3×3×256) + (256)	2,560
BatchNorm_6	Batch Normalization	(32, 32, 256)	(256×2) + 512	1,024
Pointwise Conv	Convolutional (1×1×256), stride 1, **ReLU**	(32, 32, 256)	(1×1×256×256) + (256)	65,792
BatchNorm_7	Batch Normalization	(32, 32, 256)	(256×2) + 512	1,024
SE Pooling	Global Average Pooling	(256)	0	0
SE Dense_1	Dense, **ReLU**	(64)	(256×64) + (64)	16,448
SE Dense_2	Dense, **Sigmoid**	(256)	(64×256) + (256)	16,640

Multiply	SE Scaling	(32, 32, 256)	0	0
Conv2D_5	Convolutional (1×1×256), stride 1	(32, 32, 128)	(1×1×256×128) + (128)	32,896
BatchNorm_8	Batch Normalization	(32, 32, 128)	(128×2) + 256	512
Add_2	Skip Connection	(32, 32, 128)	0	0
ReLU_2	Activation Layer, **ReLU**	(32, 32, 128)	0	0
GAP	Global Average Pooling	(128)	0	0
Dense_3	Dense, **ReLU**	(128)	(128×128) + (128)	16,512
BatchNorm_9	Batch Normalization	(128)	(128×2) + 256	512
Dropout	Dropout (Rate = 0.2)	(128)	0	0
Dense_4	Dense, **Softmax**	(4)	(128×4) + (4)	516

2.3.6 Hyperparameter Tuning

The training configuration presented includes multiple components that work together to improve the model performance and stability. Each parameter and function has a specific role in optimizing the hybrid model through an adaptive learning process.

Stratified K-Fold Cross Validation:

This evaluation technique is used to assess how a model performs across different subsets of data. Here, the entire dataset is divided into k equally sized folds, ensuring that the proportion of samples from each class remains consistent across all folds. This preservation of class distribution is important for imbalanced datasets, where certain classes may have fewer samples than others. In each iteration, one fold acts as the validation set, while the remaining folds are used for training. The process repeats k times so that each of the folds serves as the validation set at least once. The final performance metric is obtained by averaging the results from all folds, providing a more stable and unbiased estimate of the model's true performance. The parameters: shuffle=True and random_state=42 ensure that data samples are randomized before splitting, ensuring reproducibility of the results.

The Adam Optimizer:

The short form for Adaptive Moment Estimation, is a gradient-based optimization algorithm that combines the advantages of two earlier methods, AdaGrad and RMSProp. Adam calculates the adaptive learning rates for each parameter using estimates of the first and second moments of the gradients. The first moment (mean) shows the average of past gradients, providing momentum to the new ones, while the second moment (uncentered variance) measures the magnitude of recent gradients, helping to control the step sizes. By maintaining these moving averages, Adam dynamically adjusts the learning rate for each of the parameters. This makes it effective for problems like sparse gradients or noisy data. The default learning rate used here is 0.001, which provides a balance between convergence speed and stability. Adam's adaptive mechanism ensures the efficient training and faster convergence compared to the traditional stochastic gradient descent mechanisms.

Sparse Categorical Cross Entropy:

The loss function used is Sparse Categorical Cross Entropy, designed for multi-class classification tasks where target labels are provided as integers rather than one-hot vectors. The loss measures how well the predicted probability distribution aligns with the true class label. For each sample, it computes the negative logarithm of the predicted probability corresponding to the correct class. Lower loss values indicate that the model's predictions are closer to the actual labels. Sparse Categorical Cross Entropy is computationally efficient, as it does not require one-hot encoding of labels, and it ensures stable gradients during backpropagation, which is vital for convergence in deep learning models.

Accuracy:

It is an important metric used for training. It shows the ratio of correct prediction to the total number of samples. While it provides a straightforward measure of performance, only accuracy may not be sufficient for imbalanced datasets because in such datasets, the majority class will dominate. In such cases, additional metrics such as precision, recall, and F1-score become more informative. However, it is a quick indicator of the model's classification ability across all the training epochs.

EarlyStopping Callback:

To prevent overfitting and improve training efficiency, the EarlyStopping callback is used. This mechanism monitors a chosen metric—in this case, the validation loss—and halts training when the metric stops improving for a specified number of epochs, defined by the patience parameter. Here, a patience value of 10 allows the model to continue training for several epochs after the last improvement before stopping. Once training ends, the parameter restore_best_weights=True ensures that the model reverts to the weights corresponding to the best validation performance, rather than the final epoch's weights. This approach prevents the model from overfitting to noise in the later stages of training and ensures optimal generalization on unseen data.

ReduceLROnPlateau Callback:

Another essential component is the ReduceLROnPlateau callback, which adaptively modifies the learning rate when progress stagnates. This technique monitors the validation loss and reduces the learning rate by a specified factor—in this case, 0.5—after a certain number of epochs (patience=5) without improvement. The lower bound for the learning rate is set to 1e-6 to prevent excessively small updates that could stall learning. This adaptive adjustment allows the model to make larger parameter updates during the early training phase and finer updates as it approaches convergence. By gradually lowering the learning rate, the model can escape local minima and refine its learning trajectory toward an optimal solution.

Batch Size:

The batch size defines how many samples are processed before updating the model's parameters. A batch size of 32 provides a good balance between computational efficiency and gradient stability. Smaller batch sizes introduce more noise into gradient updates, which can enhance generalization but slow down convergence, while larger batches lead to smoother but potentially less diverse updates.

Training Epochs:

Epochs show the number of times the training is done. In this study, the model is trained for 50 epochs, each representing one full pass through the training dataset. All the four models have been trained under the same conditions for 50 epoch. However, if the validation loss stops improving, due to early stopping, the training may terminate earlier preventing unnecessary computation and overfitting.

Class Weights:

In case of imbalanced datasets, class weights are calculated and used in training to make the training balanced and unbiased. In datasets where certain classes are underrepresented, the model tends to become biased toward the dominant classes. By assigning higher weights to minority classes, the loss function penalizes misclassifications of these classes more heavily and thus mitigates the effect of the dominant class. This forces the model to learn features in a balanced way across classes and improves fairness in classification performance across all categories.

The confusion matrix:

The confusion matrix provides a detailed insight into model performance of each class. Each of the row of the matrix represents the true class, while each column corresponds to the predicted class. The diagonal shows the correct predictions, whereas the off-diagonal entries indicate the misclassifications. The confusion matrix analysis shows which classes are being confused with the other and helps in further tuning.

Classification Report and Performance Metrics:

The classification report provides key performance metrics for each class, including precision, recall, F1-score, and support. Precision measures how many of the predicted positives are actually correct, recall measures how many actual positives are correctly detected, and the F1-score combines both into a single mean. Support indicates the number of true samples belonging to each of the classes. Together, these metrics provide a detailed explanation of how the model performs on both majority and minority classes.

Visualization of Training and Validation Curves:

The training process also involves visualization of learning curves including accuracy and loss graphs across epochs. These plots provide shows the model's learning behavior over time. A steady decrease in training and validation loss and an increase in accuracy, indicates a proper convergence. In contrast, a widening gap between training and validation performance shows overfitting. It means the model performs well on training data but poorly on unseen data. These visualizations help in diagnosing learning issues and guide hyperparameter tuning.

Finally, once all k folds of cross-validation are completed, the average validation loss and accuracy across folds are computed. This aggregation provides a more reliable measure of the model's true generalization ability, as it mitigates bias from any single train-validation split. By training and evaluating the model multiple times on different subsets of data, the results become statistically more robust and representative of real-world performance.

Overall, these hyperparameters and techniques collectively ensure that the training process is efficient, stable, and generalizable. Each component—from the optimizer and callbacks to evaluation metrics and validation strategy—contributes to refining the model's learning dynamics and ensuring that it performs effectively across diverse and potentially imbalanced datasets.

2.3.7 Experimental Setup

All experiments were conducted using Kaggle's cloud-based GPU environment, which provided a stable and powerful computational infrastructure for deep learning model training. The hardware setup consisted of an Intel Xeon processor (4 cores, 2.00 GHz), a Tesla P100 GPU with 16 GB VRAM powered by NVIDIA, and 31 GB of RAM. This configuration allowed for efficient handling of high-resolution MRI data and the training of complex deep learning architectures. Training was performed for 50 epochs with a batch size of 32 samples, a setting chosen to balance computational efficiency and convergence stability.

Cross-validation is a model evaluation technique used to test how well a ML model generalizes to unseen data. Instead of splitting the dataset just once into training and testing

sets, cross-validation divides the dataset into multiple parts and evaluates the model multiple times. To ensure robust evaluation and reduce the risk of overfitting, a 5-Fold Stratified Cross-Validation approach was employed. In this method, the dataset is divided into five equal parts (folds). During each iteration, the model is trained on four folds and validated on the remaining fold. This process is repeated five times, ensuring that every fold is used once as a validation set. The final performance of the model is then calculated as the average of the five validation scores, providing a more reliable and unbiased estimate of the model's true performance.

Unlike simple k-Fold Cross-Validation, which may result in uneven class distributions, the stratified variant maintains the same proportion of tumor classes (glioma, meningioma, pituitary, and no tumor) across both training and validation sets. This prevents biased learning that could occur if one fold had disproportionately more samples from a particular class. Compared to Leave-One-Out Cross-Validation (LOOCV), which trains and validates on nearly every single data point but is highly computationally expensive, 5-fold cross-validation offers a practical balance between computational efficiency and evaluation robustness.

In medical imaging tasks such as brain tumor classification, this method is especially important because datasets often exhibit class imbalance. The stratified approach ensures that each class is fairly represented during both training and validation, thereby improving the generalization ability of the model. The choice of five folds strikes an optimal trade-off: using more folds (e.g., 10) increases training cost, while fewer folds (e.g., 2 or 3) reduce reliability. Thus, the 5-Fold Stratified Cross-Validation framework not only strengthens
model stability but also provides a comprehensive and fair assessment of performance across all tumor classes.

CHAPTER 3

RESULT ANALYSIS

3.1 Performance Metrics

To evaluate the performance of the models, standard performance metrics, including accuracy, precision, recall, F1-score and the confusion matrix were used. Together, these metrics offer a well-rounded view of the model's effectiveness by measuring not only overall prediction correctness but also class-specific performance and error distribution.

The confusion matrix is a tabular representation that summarizes a model's classification performance by comparing actual class labels with predicted labels. It consists of four key components:

True Positive (TP) is the instance when the model correctly recognizes a tumor (or a specified class) as present when it is actually present. True Negative (TN) is when the model accurately determines a tumor (or a specified class) as absent when it is indeed absent. False Positive (FP) is an instance when a model predicts a tumor (or a specified class) as present when it is absent, leading to a false alarm. False Negative (FN) is when the model predicts a tumor (or a specified class) as absent when it is actually present, representing a missed diagnosis. Table 3.1 outlines the parameters used to compute the evaluation metrics, which guide the interpretation of model performance throughout this study.

Table 3.1 Performance Metrics

Actual/Predicted	***Positive***	***Negative***
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

The confusion matrix is particularly valuable in medical applications because it reveals not just the overall success of the model, but also the distribution of specific errors, such as misclassifying one tumor type as another. In this study, confusion matrices were generated

for all five validation folds to assess model consistency across different subsets of data, allowing for a comprehensive understanding of classification reliability. Furthermore, these matrices visually demonstrate how effectively the hybrid model distinguishes between glioma, meningioma, pituitary, and non-tumor classes.

Accuracy measures the proportion of positive and negative cases that the model classified correctly out of all of the cases. In medical imaging, accuracy gives a global perspective about model performance. However, it may be misleading at times, especially if the dataset is imbalanced (for example, if one type of tumor appears much more often than other types).

Precision, or positive predictive value (PPV), evaluates how many of the cases the model predicted were positive and were truly positive. High precision indicates that when the model predicts the presence of a tumor, it is usually correct. This is important to consider in a clinical sense, as false positives can evoke unnecessary anxiety in patients, lead to unwanted tests, and even an unnecessary treatment process.

Recall, or sensitivity or true positive rate, measures the capacity of the model to identify all actual positive cases. High recall indicates that the model rarely misses tumor cases. Recall is crucial in health applications, because if a tumor is not detected (false negative), treatment may be delayed and could negatively affect the patient outcome.

The F1-score provides a means to combine precision and recall, taking a balanced approach to the trade-off between the two. It is especially valuable in situations where both false positives and false negatives have important implications, as seen in brain tumor diagnoses. A high F1-score indicates that the model maintains both reliable detection and accurate classification.

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

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision+Recall}$$

3.2 Comparison of Model Parameters and Storage Requirements

Table 3.2 presents the total parameters and model size of the three base models and the proposed hybrid model. Evaluating these parameters are essential in determining a model's clinical practicality that includes its computational and storage requirements. Among the baseline architectures, VGG16 has the most parameters with a total of 65,070,916 and a total storage of 248.23 MB. With 23,595,908 parameters and a storage of 90.01 MB, ResNet50 has significantly reduced the parameters. ResNet50 improves training efficiency and accuracy over VGG16 and its training parameters. Despite this improvement, the system as a whole has still not dramatically improved its training parameters. EfficientNetB3 has a more efficient design and contains 10,789,683 parameters while occupying 41.16 MB of storage. EfficientNetB3 achieves improved efficiency versus VGG16 and ResNet50 by employing compound scaling of network depth, width, and resolution. However, storage and number of parameters are still significant amounts when evaluating for deployment in low-resource clinical scenarios. The comparative analysis of these architectures highlights the trade-off between model complexity and deployment feasibility in real-world healthcare environments. Reducing the parameter count without compromising accuracy is vital for enabling faster inference, lower memory consumption, and integration into portable diagnostic systems. In comparison, the proposed hybrid model is compressed to only 577,604 parameters and requires a storage size of 2.20 MB, or a nearly 99% decrease in parameter count from VGG16 and over 95% decrease from EfficientNetB3.

Table 3.2 COMPARISON OF MODEL PARAMETERS AND STORAGE REQUIREMENTS OF THE MODELS

Model	***Total Parameters***	***Model Size***
VGG16	65,070,916	248.23 MB
ResNet50	23,595,908	90.01 MB
EfficientNetB3	10,789,683	41.16 MB
Hybrid	577,604	2.20 MB

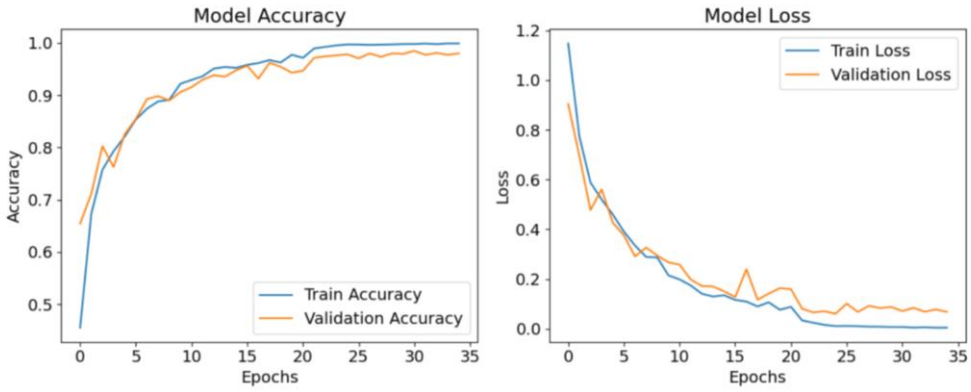
3.3 Accuracy and Loss Curves Over Epochs

Figure 3.1 illustrates the Accuracy vs. Epochs and Loss vs. Epochs curves for VGG16, ResNet50, EfficientNetB3, and the Proposed Hybrid Model. The curves highlight the comparative performance of the baseline architectures against the developed hybrid approach across 50 training epochs.

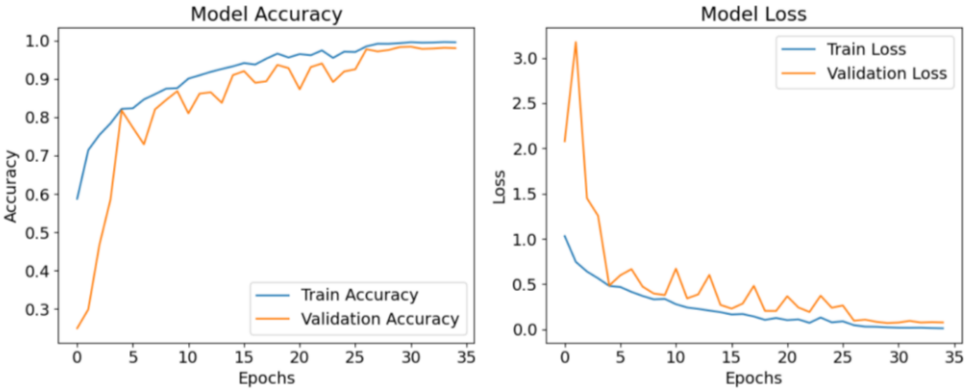
In terms of accuracy, all baseline models demonstrate gradual improvement in training accuracy over the epochs, but their validation accuracy tends to plateau early, reflecting limited generalization. ResNet50 performs better than VGG16 owing to its residual connections, yet its validation accuracy fluctuates across epochs. EfficientNetB3 achieves relatively higher accuracy compared to both VGG16 and ResNet50, benefiting from compound scaling in depth, width, and resolution, but its performance plateaus after the mid epochs without further significant gains. By contrast, the Proposed Hybrid Model consistently achieves higher accuracy throughout the training process, with both training and validation accuracy stabilizing at a higher level than all three baselines. This demonstrates the improved capability of the hybrid framework to learn discriminative features and generalize effectively.

The loss curves further validate this observation. While all baseline models show a decline in training loss, their validation loss exhibits minor fluctuations, suggesting instability and potential overfitting. VGG16 and ResNet50 end with comparatively higher final loss values, reflecting their limited learning efficiency. EfficientNetB3 achieves lower loss values than VGG16 and ResNet50, but the convergence is less smooth. The Proposed Hybrid Model, however, achieves the lowest training and validation loss among all models. Both curves steadily decrease and stabilize at lower points, indicating efficient error minimization with stronger generalization.

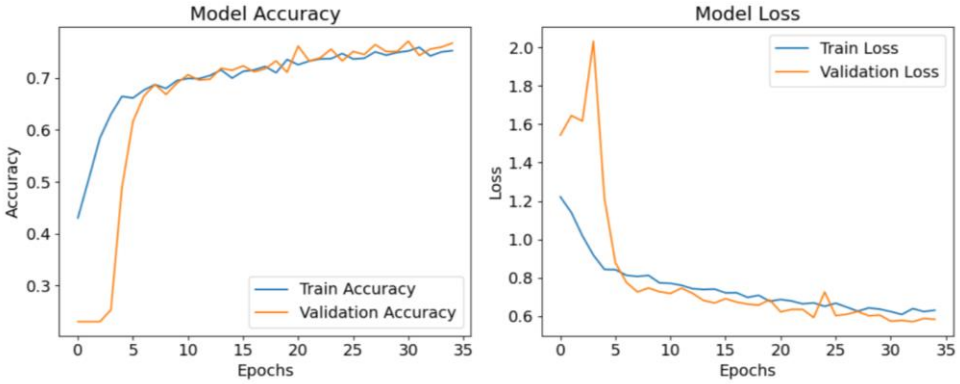
Overall, the trends clearly confirm the superiority of the Proposed Hybrid Model. By integrating the complementary strengths of VGG16, ResNet50, and EfficientNetB3 namely hierarchical feature extraction, residual learning, and compound scaling the hybrid architecture achieves higher accuracy with reduced loss, while maintaining stability across epochs. This highlights its robustness and effectiveness in brain tumor classification compared to individual state-of-the-art CNN models.



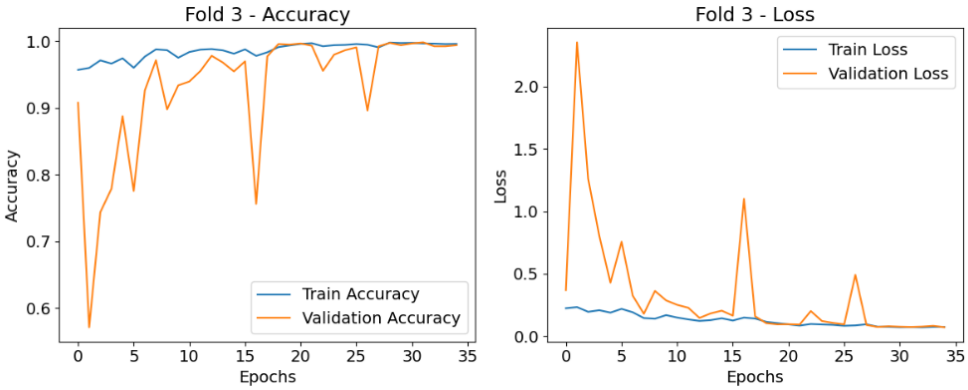
VGG16



ResNet50



EfficientNetB3

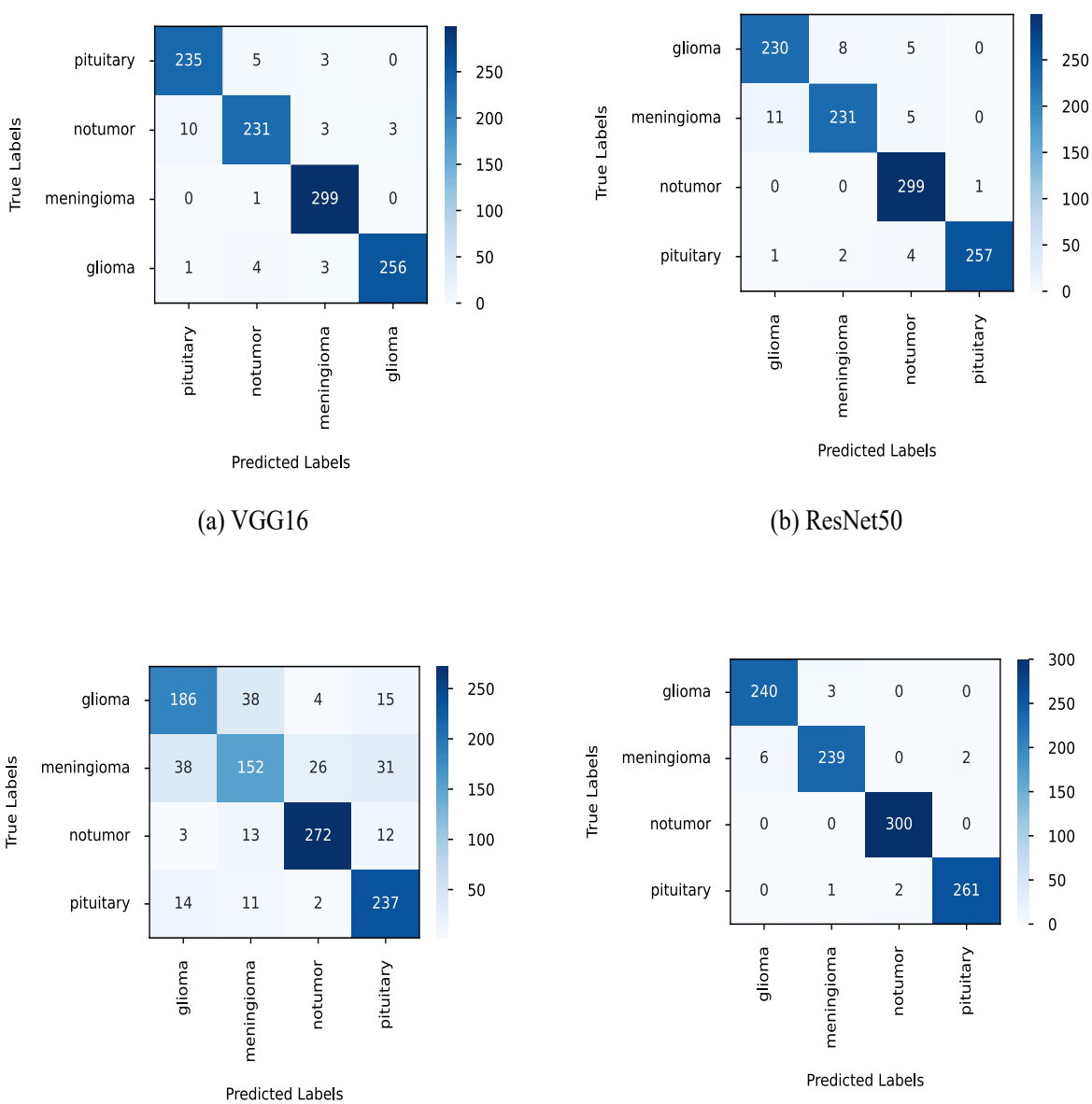


Proposed Model

Figure 3.1: Accuracy vs Epochs and Loss vs Epochs of the models

3.4 Confusion Matrix Analysis of the Models

A confusion matrix is also utilized to visualize and quantify the prediction outcomes across all classes. This analytical tool provides a detailed breakdown of how each model distinguishes between tumor types, helping to identify specific patterns of misclassification that may not be evident through overall accuracy scores alone. In medical image analysis, such visualization is crucial for understanding class-wise diagnostic strengths and weaknesses, which directly reflect the model's potential clinical reliability. The confusion matrices of VGG16, ResNet50, Efficient-NetB3 and proposed model across four tumor classes are shown in Figure 3.2. For VGG16, the model demonstrated excellent performance overall, particularly for meningioma and glioma. It achieved an accuracy of 99.67% for meningioma and 98.48% for glioma, showing very limited misclassification between these categories. ResNet50 also showed strong classification ability, especially for the No Tumor class, where it produced highly accurate predictions. However, the model struggled to fully separate meningioma from glioma, misclassifying 8.10% of meningioma cases as glioma. EfficientNetB3 did not perform as well as VGG16 and ResNet50 and was rather weak in classifying meningioma cases, achieving only 54.25% accuracy. A large portion of the cases were misclassified as glioma (22.27%) and pituitary (13.76%), indicating that the compound scaling of EfficientNetB3 failed to the necessary fine differences to be made in the dataset for EffectiveNetB3 to classify the brain tumors correctly. The proposed hybrid model achieved balanced and highly accurate performance across all the tumor classes, obtaining 97.94% for glioma, 98.38% for meningioma, 99.67% for no tumor, and 98.86% for pituitary. There were very few misclassifications, the largest being 1.65% confusion between glioma and meningioma. Relative to VGG16, ResNet50, and EfficientNetB3, the hybrid model demonstrated sustained performance across all classes, indicating its reliability and applicability across datasets.



(a) VGG16

(b) ResNet50

(c) EfficientNetB3

(d) Proposed Model

Figure 3.2: Confusion Matrix of the models

3.5 5-Fold Cross-Validation Performance

The proposed hybrid model was trained and evaluated using a 5-fold cross-validation strategy to ensure robustness and reliability of the results. This validation approach is particularly critical in medical imaging tasks, where datasets are often limited and imbalanced, as it minimizes the dependency on any single train-test split and provides a comprehensive evaluation of model stability. Each fold thus acts as an independent experiment, ensuring that the reported performance reflects true generalization rather than dataset-specific tuning. Figure 3.3 illustrates the training and validation accuracy and loss curves across the five folds. The training accuracy remained consistently high throughout all folds, indicating that the model was able to effectively learn discriminative features from the training data. Although the validation accuracy and loss exhibited minor fluctuations during the early epochs, both metrics stabilized in the later stages of training, suggesting that the model successfully avoided overfitting and maintained stable convergence across multiple training cycles. This demonstrates good generalization capability across different partitions of the dataset.

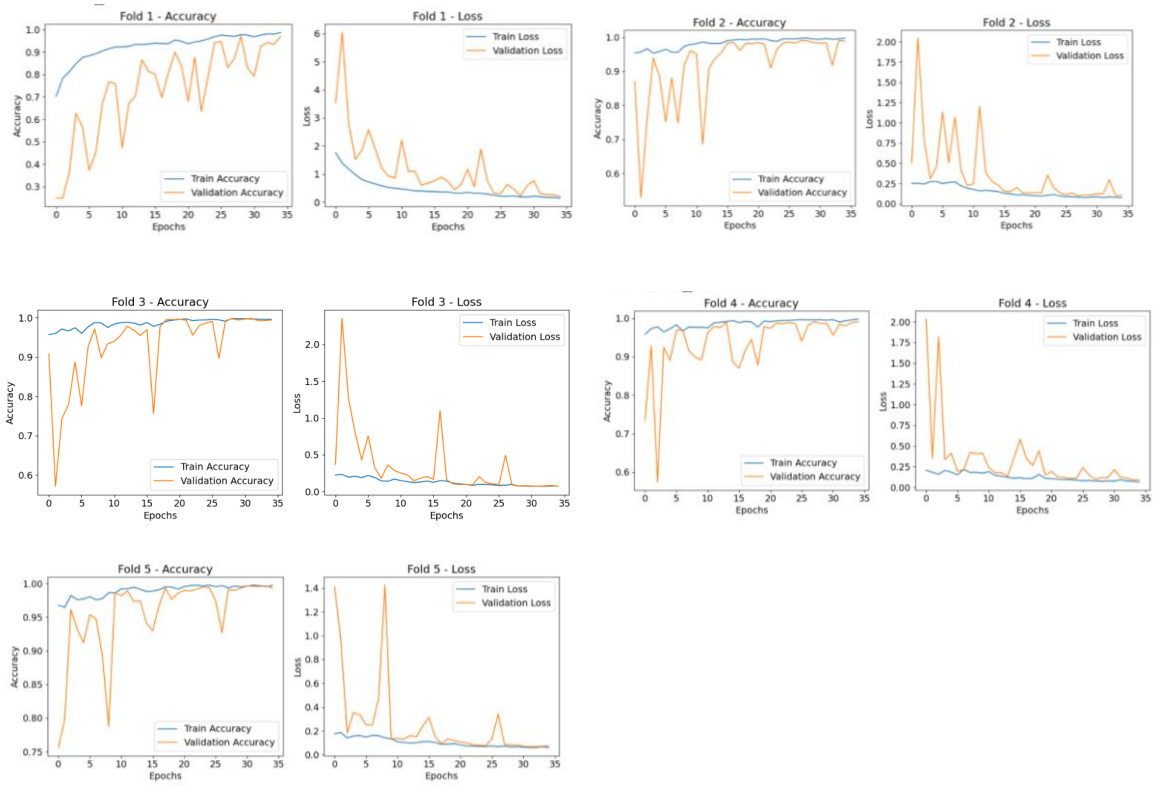


Figure 3.3: Accuracy vs Epochs and Loss vs Epochs of the 5-Fold

The performance of the model is further validated by the confusion matrices for each fold, as shown in Figure 3.4. In all five folds, the highest values are concentrated along the diagonals, which represent correctly classified samples. This consistent diagonal dominance highlights the model's ability to accurately classify glioma, meningioma, pituitary tumors, and no-tumor cases across all splits. Misclassifications were minimal and scattered, reflecting those errors were not biased toward a particular class. Overall, the confusion matrices confirm that the proposed hybrid model achieved strong and stable classification performance across all folds, supporting its robustness and reliability in handling multi-class brain tumor classification tasks.

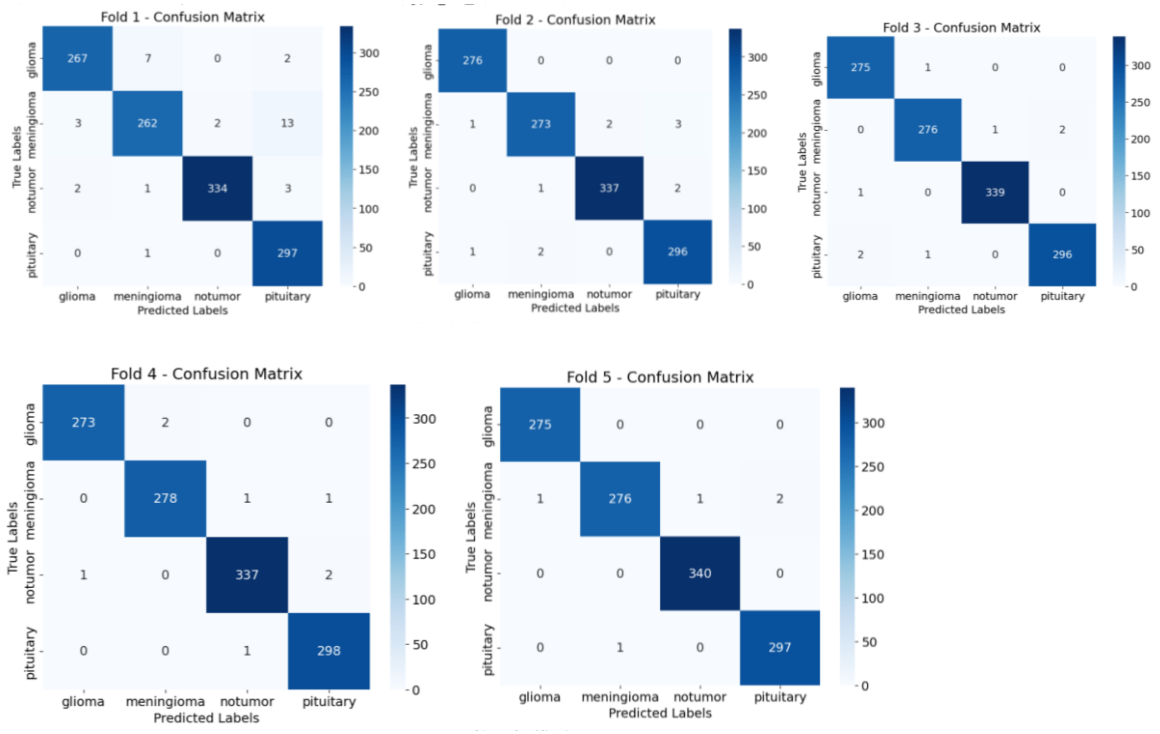


Figure 3.4: Confusion Matrix of the 5-Fold

3.6 Performance Comparison of the Models

Table 3.3 summarizes the accuracy, precision, recall, and F1-score performance metrics for the VGG16, ResNet50, EfficientNetB3, and the proposed hybrid model on the four classes of brain tumors: glioma, meningioma, no tumor, and pituitary tumor. Evaluating these metrics provides a detailed understanding of how each model performs across different pathological conditions, reflecting not only their general accuracy but also their diagnostic reliability for each tumor type.

The results achieved by VGG16 demonstrate that the model performs strongly and consistently across all classes. For glioma, VGG16 obtained an accuracy of 97.44%, precision of 98.86%, and recall of 98.48%, showing that VGG16 correctly identified most gliomas while avoiding false positives. In the meningioma case, VGG16 performance was even more outstanding, attaining a precision and recall of over 98% and an F1-score of 99.01%, the highest of all models. For the no-tumor category, performance remained strong with a recall of 93.93% and an F1-score of 95.47%. Pituitary tumors were also classified well, with an accuracy of 97.12%, providing further evidence of the robustness of VGG16 across multiple categories. However, despite its high performance, VGG16 is computationally heavy, requiring large memory and storage, which limits its real-time applicability in clinical environments.

ResNet50 performs exceptionally well for no-tumor cases, achieving 99.67% recall and an F1-score of 97.71%, surpassing VGG16 in this category. This demonstrates ResNet50's strength in differentiating healthy patients from tumor-bearing cases. ResNet50 also achieves the highest performance for pituitary tumors across all models, with 99.22% precision and 98.08% recall, confirming its robustness in identifying pituitary-related anomalies. However, the model struggles with recalling meningioma, where recall drops to 88.66%, yielding an F1-score of 91.44%. For glioma, ResNet50 also shows slightly lower performance compared to VGG16, with an F1-score of 92.93%. These results indicate that while ResNet50 benefits from its residual learning mechanism for deeper network training, it occasionally overgeneralizes features between visually similar tumor types, particularly glioma and meningioma. Nevertheless, its stability across multiple classes makes it a strong baseline for clinical diagnostic systems.

EfficientNetB3, despite its reputation for balancing accuracy and efficiency, performs noticeably weaker in this study, particularly in glioma and meningioma detection. For gliomas, it achieves only 74.33% precision and 79.84% recall, resulting in an F1-score of 76.98%, which is significantly lower than both VGG16 and ResNet50. The weakest performance was for meningioma, with recall at 54.25% and F1-score at 62.18%, indicating an inability to reliably distinguish meningioma features from other tumor types. However, EfficientNetB3 achieves relatively strong results in the no-tumor and pituitary categories, scoring 92.00% and 91.67% respectively. This suggests that while EfficientNetB3's compound scaling provides efficiency advantages, it may not sufficiently capture subtle spatial and intensity variations in complex MRI datasets, leading to reduced accuracy for tumor-specific differentiation.

The proposed hybrid model demonstrates more robust and balanced performance across all tumor classes, overcoming the weaknesses observed in individual architectures. Specifically, glioma detection scored 99.17% precision, 97.94% recall, and an F1-score of 98.55%, outperforming all three baseline models. Similarly, meningioma diagnosis achieved 97.59% precision, 98.38% recall, and 97.98% F1-score, demonstrating the hybrid model's strong capacity to handle inter-class variability. In particular, the hybrid model excels in detecting the absence of tumors, with each metric exceeding 99.67%, and maintains equally high performance for pituitary tumor classification (98.86% recall and 98.68% F1-score). This consistency across all tumor categories indicates that the hybrid model successfully combines the spatial precision of VGG16, the depth stability of ResNet50, and the scaling efficiency of EfficientNetB3. In addition, the model achieves this superior performance with only 577,604 parameters and a total storage requirement of 2.20 MB, a fraction of the size of conventional architectures. This compact design ensures faster inference, reduced memory overhead, and high suitability for real-world deployment in low-resource clinical setups or portable diagnostic systems.

Overall, the proposed hybrid model demonstrates exceptional adaptability, achieving near-perfect class discrimination with minimal trade-offs, establishing itself as a reliable and computationally efficient tool for automated brain tumor diagnosis.

Table 3.3 PERFORMANCE COMPARISON OF THE MODELS

Model	Diseases	Precision	Recall	F1-score	Accuracy
VGG16	Glioma	98.86	98.48	98.67	97.44
	Meningioma	98.36	99.67	99.01	
	No Tumor	97.07	93.93	95.47	
	Pituitary	95.16	97.12	96.13	
ResNet50	Glioma	91.27	94.65	92.93	95.26
	Meningioma	94.40	88.66	91.44	
	No Tumor	95.83	99.67	97.71	
	Pituitary	99.22	98.08	98.08	
EfficientNetB3	Glioma	74.33	79.84	76.98	80.27
	Meningioma	72.83	54.25	62.18	
	No Tumor	90.49	92.00	91.24	
	Pituitary	79.61	91.67	85.21	
Hybrid	Glioma	99.17	97.94	98.55	98.77
	Meningioma	97.59	98.38	97.98	
	No Tumor	99.67	99.67	99.67	
	Pituitary	98.49	98.86	98.68	

3.7 Analysis of the 5-Fold Performance of the Hybrid Model

Table 3.4 summarizes the quantitative performance of the proposed hybrid deep learning model across all five folds using key metrics: precision, recall, F1-score, and accuracy. These metrics collectively assess the model's classification reliability, sensitivity, and generalization capability for the four target categories: Glioma, Meningioma, Pituitary, and No Tumor. The results indicate that the model achieved remarkably stable and near-perfect performance across all folds, with overall accuracy consistently at or above 0.99. This consistency across multiple folds highlights the absence of data-dependent bias and confirms that the model's learning process generalizes effectively to unseen samples.

The Glioma class maintained exceptionally high precision and recall across folds, with values ranging from 0.98 to 1.00, demonstrating the model's strong ability to distinguish glioma regions from other tumor types. The F1-score of 0.99–1.00 confirms that false positives and false negatives were almost negligible. This high reliability suggests that the hybrid model effectively captured the irregular textures and non-uniform boundaries typically associated with gliomas in MRI scans. Since gliomas often exhibit heterogeneous tissue composition and infiltrative growth patterns, achieving nearly perfect precision and recall in this class underscores the hybrid model's deep representation power and its capability to learn complex spatial-intensity relationships.

For the Meningioma class, performance also remained outstanding, with precision and recall ranging between 0.98 and 1.00, though slight variations were observed in recall (0.98–0.99) across a few folds. These small deviations can be attributed to the morphological similarity of meningioma tissues to healthy meninges or neighboring tumor types, making this class slightly more challenging to classify. However, with F1-scores consistently between 0.98 and 0.99, the impact of these variations is minimal, confirming the model's robustness in handling inter-class overlap. The model's consistent identification of meningioma despite such structural similarities demonstrates its ability to capture discriminative high-level semantic features while minimizing overfitting to low-level noise.

The No Tumor class achieved perfect precision, recall, and F1-scores of 1.00 across almost all folds, illustrating that the model reliably identified non-pathological brain scans without confusion. This stability indicates the hybrid model's efficiency in distinguishing tumor-free cases, which is particularly important for minimizing unnecessary false alarms in diagnostic

settings. Similarly, the Pituitary Tumor class achieved exceptional performance, with all evaluation metrics equal to or very close to 1.00 in every fold. This result confirms the model's capacity to learn and generalize well to the distinct visual patterns of pituitary tumors, which are generally more localized and structurally consistent in MRI imagery.

Overall, the mean accuracy across all folds was approximately 0.996, with an average precision of 0.995, average recall of 0.995, and average F1-score of 0.995. These consistently high results highlight the robust learning capability and strong generalization performance of the proposed hybrid model across diverse data splits. The low variance observed between folds (less than 0.01 difference across metrics) further validates the model's stability, indicating that its decision boundaries remain consistent regardless of the training subset used. The combination of convolutional, residual, and efficient scaling components within the hybrid architecture likely contributed to this stability by enabling multiscale feature extraction and enhanced gradient propagation. The minimal fluctuation of performance metrics across folds demonstrates that the model does not overfit to any particular partition of data, confirming its generalization strength, diagnostic reliability, and adaptability to unseen MRI inputs.

Such fold-wise consistency not only reinforces the robustness of the proposed system but also aligns with the reliability standards required for medical imaging applications, where predictive uniformity across varying patient datasets is critical. Thus, the 5-fold cross-validation results establish that the proposed hybrid model achieves state-of-the-art performance with exceptional consistency across all tumor types, validating its potential for deployment in real-world clinical brain tumor diagnosis systems. These findings ultimately support the use of hybrid CNN frameworks as dependable diagnostic tools, offering both accuracy and computational efficiency suitable for integration into clinical decision-support pipelines.

Table 3.4 5-FOLD CROSS-VALIDATION PERFORMANCE OF THE PROPOSED MODEL

Fold number	Disease's name	Precision	Recall	F1-score	Accuracy
Fold 1	Glioma	0.98	0.97	0.97	0.97
	Meningioma	0.97	0.94	0.95	
	No Tumor	0.99	0.98	0.99	
	Pituitary	0.94	1	0.97	
Fold 2	Glioma	0.99	1	1	0.99
	Meningioma	0.99	0.98	0.98	
	No Tumor	0.99	0.99	0.99	
	Pituitary	0.98	0.99	0.99	
Fold 3	Glioma	0.99	1	0.99	0.99
	Meningioma	0.99	0.99	0.99	
	No Tumor	1	1	1	
	Pituitary	0.99	0.99	0.99	
Fold 4	Glioma	1	0.99	0.99	0.99
	Meningioma	0.99	0.99	0.99	
	No Tumor	0.99	0.99	0.99	
	Pituitary	0.99	1	0.99	
Fold 5	Glioma	1	1	1	1
	Meningioma	1	0.99	0.99	
	No Tumor	1	1	1	
	Pituitary	0.99	1	0.99	

CHAPTER 4

DEMONSTRATION OF OUTCOME BASED EDUCATION (OBE)

This chapter outlines how the outcomes of the thesis align with the principles of Outcome Based Education, focusing on the achievement of specific Course Outcomes (COs) and Program Outcomes (POs).

4.1 Introduction

The industry witnessed a growing demand for engineers equipped not only with technical knowledge but also with problem solving, ethical reasoning, and communication skills all of which are fostered through OBE frameworks in engineering education.

4.2 Course Outcomes (COs) Addressed

The following table shows the COs addressed in EEE 4700 for Project and Thesis.

COs	CO Statement	POs	Put Tick (√)
CO1	Apply knowledge of electrical and electronic engineering to the solution of complex engineering problems.	PO1	√
CO2	Identify a contemporary real-life problem related to electrical and electronic engineering by reviewing and analyzing existing research works.	PO2	√
CO3	Determine functional requirements of the problem considering feasibility and efficiency through analysis and synthesis of information.	PO4	√
CO4	Select a suitable solution and determine its method considering professional ethics, codes and standards.	PO8	√
CO5	Adopt modern engineering resources and tools for the solution of the problem.	PO5	√
CO6	Prepare management plan and budgetary implications for the solution of the problem.	PO11	√
CO7	Analyze the impact of the proposed solution on health, safety, culture and society.	PO6	√
CO8	Analyze the impact of the proposed solution on environment and sustainability.	PO7	√
CO9	Develop a viable solution considering health, safety, cultural, societal and environmental aspects.	PO3	
CO10	Work effectively as an individual and as a team member for the accomplishment of the solution.	PO9	√
CO11	Prepare various technical reports, design documentation, and deliver effective presentations for demonstration of the solution.	PO10	√
CO12	Recognize the need for continuing education and participation in professional societies and meetings.	PO12	

4.3 Aspects of Program Outcomes (POs) Addressed

The following table shows the aspects addressed for certain Program Outcomes (POs) addressed in EEE 4700 for Project and Thesis.

	Statement	Different Aspects	Put Tick (√)
PO3	**Design/development of solutions**: Design solutions for complex electrical and electronic engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	Public health	√
		Safety	√
		Cultural	
		Societal	√
		Environmental	
PO4	**Investigation:** Conduct investigations of complex electrical and electronic engineering problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.	Design of experiments	√
		Analysis and interpretation of data	√
		Synthesis of information	√
PO6	**The engineer and society:** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice and solutions to complex electrical and electronic engineering problems.	Societal	√
		Health	√
		Safety	√
		Legal	
		Cultural	
PO7	**Environment and sustainability:** Understand and evaluate the sustainability and impact of professional engineering work in the solution of complex electrical and electronic engineering problems in societal and environmental contexts.	Societal	√
		Environmental	
PO8	**Ethics:** Apply ethical principles embedded with religious values, professional ethics and responsibilities, and norms of electrical and electronic engineering practice.	Religious values	
		Professional ethics and responsibilities	√
		Norms	
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	Comprehend and write effective reports	√
		Design documentation	√
		Make effective presentations	√
		Give and receive clear instructions	√
PO11	**Project management and finance:** Demonstrate knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	Engineering management principles	√
		Economic decision-making	√
		Manage projects	√
		Multidisciplinary environments	

4.4 Knowledge Profiles (K3 – K8) Addressed

The following table shows the Knowledge Profiles (K3 – K8) addressed in EEE 4700 for Project and Thesis.

K	Knowledge Profile (Attribute)	Put Tick (√)
K3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline	√
K4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline	
K5	Knowledge that supports engineering design in a practice area	√
K6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline	√
K7	Comprehension of the role of engineering in society and identified issues in engineering practice in the discipline: ethics and the engineer's professional responsibility to public safety; the impacts of engineering activity; economic, social, cultural, environmental and sustainability	√
K8	Engagement with selected knowledge in the research literature of the discipline	

The following table explains or justifies how the Knowledge Profiles (K3 – K8) have been addressed in EEE 4700/4800 (Project and Thesis).

K	Explanation/Justification
K3	The project involves a theory-based approach to machine learning (ML) and deep learning techniques
K4	
K5	The project involves designing and optimizing an engineering solution for brain tumor detection.
K6	Practical engineering applications in medical imaging technology
K7	This project directly addresses the ethical and societal impacts of engineering in healthcare. Automated brain tumor detection systems have significant implications for public safety and medical decision-making, as they may influence diagnosis and treatment plans.
K8	

4.5 Use of Complex Engineering Problems

The project conducted under EEE 4700 addresses the complex engineering problem of classifying brain tumors from MRI images using deep learning methods. The problem of classifying brain MRI scans from medical imaging data is complex due to the high dimensional data, variability in tumors, and the precise differentiation required for various tumor types. This effort needs sophisticated computational models which learn intricate and non-linear relationships in medical images.

The engineering challenge lies in developing a robust classification system that can accurately categorize tumors such as gliomas, meningiomas, and pituitary tumors. This task is multistage, encompassing data cleansing and preprocessing, feature representation, model selection, and performance assessment, all of which are design challenges using machine learning. The project demands not only technical proficiency but also analytical reasoning to interpret results and improve model performance through iterative refinement.

This problem qualifies as complex because it does not have a single, straightforward solution. Different approaches such as the application of state-of-the-art convolutional neural networks (e.g., ResNet50, VGG16, EfficientNetB3) must be evaluated and adapted to achieve optimal classification accuracy. The engineering tools and platforms used, including Python, TensorFlow, and Keras, further support the professional execution of the solution, aligning with current industry practices in artificial intelligence and medical technology.

Apart from the technical aspects, the project encompasses issues such as the possible impact on healthcare, the trustworthiness of automated systems, and responsibility in medical diagnosis which adds ethical dimensions to the project. These factors justify the multidisciplinary approach of the project and its social significance, defining it as a complex engineering problem under the outcome-based education (OBE) model.

4.6 Socio-Cultural, Environmental, And Ethical Impact

Socio-cultural: From a socio-cultural perspective, the adoption of AI in tumor detection democratizes access to advanced healthcare services, especially in regions with limited medical expertise. By providing accurate and early diagnosis, the model can reduce cancer-related mortality and enhance patient outcomes. This project could also alleviate pressure on healthcare professionals, allowing them to focus on critical decision-making rather than manual image analysis, ultimately contributing to an improved healthcare system and societal well-being.

Environmental: The implementation of deep learning models in medical imaging can lead to significant efficiencies. By reducing the need for repeat diagnostic procedures due to human error, it minimizes the associated use of resources, such as medical equipment and consumables, thereby reducing medical waste. Moreover, enabling remote diagnosis reduces patient and clinician travel, which helps lower carbon emissions and conserves energy.

Ethical: The use of patient data for training models presents challenges related to data privacy and informed consent. To address these issues, the project will incorporate strict data protection protocols and adhere to regulatory guidelines to ensure compliance with privacy standards, such as anonymizing patient data. Additionally, to mitigate algorithmic biases that could lead to unequal healthcare outcomes, the model will be trained on diverse datasets and rigorously evaluated for fairness and accuracy. The ethical integrity of the model will be prioritized to ensure that its deployment supports equitable and trustworthy healthcare delivery.

4.7 Attributes of Ranges of Complex Engineering Problem Solving (P1 – P7) Addressed

The following table shows the attributes of ranges of Complex Engineering Problem Solving (P1 – P7) addressed in EEE 4700 for Project and Thesis.

P	**Range of Complex Engineering Problem Solving**	**Put Tick (√)**
Attribute	Complex Engineering Problems have characteristic P1 and some or all of P2 to P7:	
Depth of knowledge required	**P1:** Cannot be resolved without in-depth engineering knowledge at the level of one or more of K3, K4, K5, K6 or K8 which allows a fundamentals-based, first principles analytical approach	√
Range of conflicting requirements	**P2:** Involve wide-ranging or conflicting technical, engineering and other issues	√
Depth of analysis required	**P3:** Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models	√
Familiarity of issues	**P4:** Involve infrequently encountered issues	
Extent of applicable codes	**P5:** Are outside problems encompassed by standards and codes of practice for professional engineering	
Extent of stakeholder involvement and conflicting requirements	**P6:** Involve diverse groups of stakeholders with widely varying needs	
Interdependence	**P7:** Are high level problems including many component parts or sub-problems	√

The following table explains or justifies how the attributes of ranges of Complex Engineering Problem Solving (P1 – P7) have been addressed in EEE 4700/4800 (Project and Thesis).

P	**Explanation/Justification**
P1	The project requires a deep understanding of machine learning, deep learning, and image processing, particularly in relation to medical images (MRI scans).
P2	The project must balance a range of requirements, including model accuracy, computational efficiency, and real-world applicability in medical settings.
P3	The problem of accurate brain tumor segmentation and classification has no obvious solution. It requires originality in analysis and the development of sophisticated models to handle the variability in tumor shapes, sizes, and appearances across patients.
P4	
P5	
P6	
P7	This project consists of multiple interconnected components. These include preprocessing MRI data, building and training deep learning models, performing segmentation and classification tasks, and evaluating the model's effectiveness

4.8 Attributes of Ranges of Complex Engineering Activities (A1 – A5) Addressed

The following table shows the attributes of ranges of Complex Engineering Activities (A1 – A5) addressed in EEE 4700 for Project and Thesis.

A	**Range of Complex Engineering Activities**	**Put Tick (√)**
Attribute	Complex activities means (engineering) activities or projects that have some or all of the following characteristics:	
Range of resources	**A1:** Involve the use of diverse resources (and for this purpose resources include people, money, equipment, materials, information and technologies)	√
Level of interaction	**A2:** Require resolution of significant problems arising from interactions between wide-ranging or conflicting technical, engineering or other issues	
Innovation	**A3:** Involve creative use of engineering principles and research-based knowledge in novel ways	√
Consequences for society and the environment	**A4:** Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation	√
Familiarity	**A5:** Can extend beyond previous experiences by applying principles-based approaches	√

The following table explains or justifies how the attributes of ranges of Complex Engineering Activities (A1 – A5) have been addressed in EEE 4700/4800 (Project and Thesis).

A	**Explanation/Justification**
A1	The project involves the use of diverse resources such as medical MRI datasets, computational tools (deep learning frameworks, cloud computing resources), and technological expertise in machine learning and image processing.
A2	
A3	The project leverages innovative engineering principles by applying unsupervised ML and CNN techniques to brain tumor segmentation and classification.
A4	The project has significant societal consequences due to its potential application in the healthcare sector. Early and accurate detection of brain tumors can save lives, reduce treatment costs, and improve patient outcomes.
A5	The project may extend beyond previous engineering experiences by applying machine learning principles in new ways, particularly unsupervised techniques in brain tumor segmentation, which is less commonly encountered compared to supervised methods.

CHAPTER 5

CONCLUSION

5.1 Summary

The early and accurate diagnosis of brain tumors is one of the most critical challenges in medical imaging, as it directly impacts patient treatment planning and overall survival rate. This study aimed to address the limitations of existing computer aided diagnostic systems by developing a hybrid deep learning model that integrates the complementary strengths of three established architectures VGG16, ResNet50, and EfficientNetB3. The proposed model was designed to achieve high classification accuracy while maintaining computational efficiency, thereby overcoming the typical trade-off between performance and complexity observed in many state-of-the-art networks.

The experimental framework utilized a publicly available MRI dataset comprising 7,023 images categorized into four classes: glioma, meningioma, pituitary tumor, and no tumor. A rigorous preprocessing pipeline was implemented, which included image resizing, normalization, RGB conversion, and extensive data augmentation to enhance model generalization. The hybrid model was trained and evaluated using 5-fold cross-validation, ensuring that performance metrics were not biased toward any specific subset of the data.

The results demonstrated that the proposed hybrid architecture consistently achieved outstanding classification performance, attaining an overall accuracy of 98.77%. The confusion matrices confirmed that the majority of predictions were correct, with strong diagonal dominance observed across all tumor categories. The Glioma, Pituitary, and No Tumor classes achieved near-perfect precision, recall, and F1-scores, while the Meningioma class exhibited only minimal variability, underscoring the robustness of the proposed system. Notably, the model achieved this high performance with only 577,604 trainable parameters and a total model size of approximately 2.20 MB, indicating a compact and computationally efficient design suitable for deployment in real-time clinical applications.

When compared to standalone models, the proposed hybrid network significantly improved accuracy and generalization while reducing overfitting tendencies. The smooth convergence

patterns of training and validation curves further validate the model's stable learning behavior. These findings suggest that the hybrid model can serve as a reliable diagnostic aid for radiologists, reducing manual interpretation time and mitigating inter-observer variability in tumor diagnosis.

In conclusion, this research successfully demonstrates that integrating multiple CNN architectures into a unified hybrid model can effectively enhance the accuracy, robustness, and computational efficiency of brain tumor classification systems. The approach provides a strong foundation for automated medical image analysis and can be extended to other neurological imaging applications.

5.2 Future Scope

Future research can focus on expanding this framework to 3D MRI volumes for volumetric tumor segmentation, integrating attention mechanisms or transformer-based modules for improved contextual learning, and exploring federated or privacy preserving learning frameworks to enable secure clinical deployment. Additionally, validating the model on larger, multi-institutional datasets will further establish its generalizability and readiness for real-world clinical integration.

References

[1] Mount Elizabeth Hospitals, “Brain Tumours,” 2025. Accessed: May 13, 2025.

[2] M. L. Bondy, M. E. Scheurer, B. Malmer, J. S. Barnholtz-Sloan, F. G. Davis, D. Il’yasova, C. Kruchko, B. J. McCarthy, P. Rajaraman, J. A. Schwartzbaum, S. Sadetzki, B. Schlehofer, T. Tihan, J. L. Wiemels, M. Wrensch, and P. A. Buffler, “Brain tumor epidemiology: Consensus from the brain tumor epidemiology consortium,” Cancer, vol. 113, pp. 1953–1968, 10 2008.

[3] R. L. Siegel, A. N. Giaquinto, and A. Jemal, “Cancer statistics, 2024,” CA: A Cancer Journal for Clinicians, vol. 74, pp. 12–49, 1 2024.

[4] American Cancer Society, “Key Statistics for Brain and Spinal Cord Tumors,” 2025. Accessed: May 13, 2025.

[5] Ostrom, Q. T., Gittleman, H., Liao, P., Rouse, C., Chen, Y., Dowling, J., ... & Barnholtz-Sloan, J. S. (2018). CBTRUS statistical report: Primary brain and other central nervous system tumors diagnosed in the United States. *Neuro-Oncology*, 20(suppl_4), iv1–iv86.

[6] Global Burden of Disease Cancer Collaboration. (2019). Global, regional, and national cancer incidence, mortality, and prevalence. *JAMA Oncology*, 5(12), 1749–1768.

[7] Louis, D. N., Perry, A., Wesseling, P., Brat, D. J., Cree, I. A., Figarella-Branger, D., ... & Ellison, D. W. (2021). The 2021 WHO classification of tumors of the central nervous system: a summary. *Neuro-Oncology*, 23(8), 1231–1251.

[8] G. Lee, M. D. Does, R. Avila, J. Kang, K. D. Harkins, Y. Wu, W. E. Banks, M. Park, D. Lu, X. Yan, J. U. Kim, S. M. Won, A. G. Evans,J. T. Joseph, C. L. Kalmar, A. C. Pollins, H. Karagoz, W. P. Thayer, Y. Huang, and J. A. Rogers, “Implantable, bioresorbable radio
frequency resonant circuits for magnetic resonance imaging,” Advanced Science, vol. 11, 7 2024.

[9] B. N. Joe, M. B. Fukui, C. C. Meltzer, Q. shou Huang, R. S. Day, P. J. Greer, and M. E. Bozik, “Brain tumor volume measurement: Comparison of manual and semiautomated methods,” Radiology, vol. 212, pp. 811– 816, 9 1999.

[10] T. A. Soomro, L. Zheng, A. J. Afifi, A. Ali, S. Soomro, M. Yin, and J. Gao, “Image segmentation for mr brain tumor detection using machine learning: A review,” IEEE Reviews in Biomedical Engineering, vol. 16, pp. 70–90, 2023.

[11] M. Nazir, S. Shakil, and K. Khurshid, “Role of deep learning in brain tumor detection and classification (2015 to 2020): A review,” Computerized Medical Imaging and Graphics, vol. 91, p. 101940, 7 2021.

[12] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., ... & van Ginneken, B. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88.

[13] S. K. Shil, F. P. Polly, M. A. Hossain, M. S. Ifthekhar, M. N. Uddin, and Y. M. Jang, “An improved brain tumor detection and classification mechanism,” pp. 54–57, IEEE, 10 2017.

[14] K. S. Sankaran, A. S. Poyyamozhi, S. S. Ali, and Y. Jennifer, “A Conceptual and Effective Scheme for Brain Tumor Identification Using Robust Random Forest Classifier,” pp. 109–118. 2022.

[15] L. O. Chua, “CNN: A vision of complexity,” *International Journal of Bifurcation and Chaos*, vol. 07, pp. 2219–2425, 10 1997.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 9 2014.

[17] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *ICML*, PMLR, pp. 6105–6114, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE, pp. 770–778, 6 2016.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” IEEE, pp. 1–9, 6 2015.

[20] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” IEEE, pp. 2261–2269, 7 2017.

[21] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 10 2010.

[22] M. S. A. Yajid, O. Abdeljaber, B. Jayaprakash, P. Rani, N. Bhosle, M. J. Ramudu, A. Alkhayyat, and D. Singh, “A novel technique for still image classification using hybrid learning: Benchmark dataset,” *National Academy Science Letters*, 11 2024.

[23] E.-S. A. El-Dahshan, T. Hosny, and A.-B. M. Salem, “Hybrid intelligent techniques for mri brain images classification,” Digital Signal Processing, vol. 20, pp. 433–441, 3 2010

[24] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, “A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning,” *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 757–775, Feb. 2020, doi: 10.1007/s00034-019-01246-3.

[25] X. Liu and Z. Wang, “Deep Learning in Medical Image Classification from MRI-based Brain Tumor Images,” arXiv preprint arXiv:2408.00636, Aug. 2024, doi: 10.48550/arXiv.2408.00636.

[26] O. Özkaraca *et al.*, “Multiple Brain Tumor Classification with Dense CNN Architecture Using Brain MRI Images,” *Life*, vol. 13, no. 2, p. 349, Jan. 2023, doi: 10.3390/life13020349.

[27] E. Irmak, “Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework,” *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, no. 3, pp. 1015–1036, Sep. 2021, doi: 10.1007/s40998-021-00426-9.

[28] A. Bang, V. Bajpai, S. Magar, R. Bagde, and P. Jamadar, “Integrating Deep and Machine Learning Techniques for Brain Tumor Detection,” in *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, IEEE, May 2024, pp. 440–445. doi: 10.1109/InCACCT61598.2024.10551095.

[29] A. Çinar and M. Yildirim, “Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture,” *Medical Hypotheses*, vol. 139, p. 109684, Jun. 2020, doi: 10.1016/j.mehy.2020.109684.

[30] F. Zulfiqar, U. Ijaz Bajwa, and Y. Mehmood, “Multi-class classification of brain tumor types from MR images using EfficientNets,” Biomedical Signal Processing and Control, vol. 84, p. 104777, Jul. 2023, doi: 10.1016/j.bspc.2023.104777.

[31] M. Nickparvar, "Brain tumor classification (MRI)," Kaggle, 2021. [Online]. Available: https://doi.org/10.34740/kaggle/dsv/2645886

[32] Huyen, “An Overview of VGG16 and NIN Models,” Medium, Aug. 31, 2021. [Online]. Available: https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484

[33] N. A. A.-H. And and M. Prince, “A Classification of Arab Ethnicity Based on Face Image Using Deep Learning Approach,” IEEE Access, vol. 9, pp. 50755–50766, 2021, doi: 10.1109/ACCESS.2021.3069022.

[34] A. Soleimanipour, M. Azadbakht, and A. Rezaei Asl, “Cultivar identification of pistachio nuts in bulk mode through EfficientNet deep learning model,” Journal of Food Measurement and Characterization, vol. 16, no. 4, pp. 2545–2555, Aug. 2022, doi: 10.1007/s11694-022-01367-5.