VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY
THE INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BRAIN TUMOR CLASSIFICATION USING COMBINATION PRETRAINED CONVOLUTIONAL NEURAL NETWORK (CNN) MODELS

By
Dang Quang Vinh

A thesis submitted to the School of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Information Technology/Computer Science/Computer Engineering

Ho Chi Minh City, Vietnam
2024

BRAIN TUMOR CLASSIFICATION USING COMBINATION PRETRAINED CONVOLUTIONAL NEURAL NETWORK (CNN) MODELS

APPROVED BY:

___________________________________ ,

Assoc. Prof. Nguyen Van Sinh

APPROVED BY:

___________________________________ ,

APPROVED BY:

___________________________________ ,

APPROVED BY:

___________________________________ ,

THESIS COMMITTEE

(Whichever applies)

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Assoc. Prof. Nguyen Van Sinh for his invaluable guidance, support and expertise throughout the research and writing process of this thesis report. His insights and feedback have been instrumental in shaping the direction and quality of this work.

I am also extremely grateful to all the teachers in the School of Computer Science and Engineering at the International University HCM for their dedication and commitment to providing quality education. Their expertise, knowledge and countless lectures have greatly contributed to my growth and development as a researcher and student. Their guidance and teachings have equipped me with the necessary skills and knowledge to undertake this research project.

Furthermore, I would like to extend my appreciation to the faculty and staff of the School of Computer Science and Engineering for creating a supportive and nurturing environment. Their tireless efforts in organizing seminars, workshops and academic events have enriched my learning experience and provided me with valuable opportunities to interact with experts in the field.

With a deep sense of gratitude, I would like to express my heartfelt appreciation to my family and friends. Their unwavering encouragement and unwavering belief in my capabilities have been an invaluable source of strength and motivation throughout my journey. Their constant support has been instrumental in helping me achieve my goals and overcome challenges along the way. I am truly fortunate to have such a wonderful support system, and I am eternally grateful for their love and belief in me.

This thesis report would not have been possible without the collective effort and support of these individuals and institutions. With a deep of gratitude to all who have been part of this incredible journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Brain tumors are one of the most serious and potentially life-threatening medical conditions. Each year, thousands of people die from brain tumors due to the rapidly growing tumor cells. In order to save lives worldwide, automatic brain tumor detection is critically important.

Quick and accurate identification of the tumor type is essential for determining the most effective treatment approach and improving outcomes. Currently, histopathological examination of biopsy samples by medical professionals is used for diagnosis and classification of brain tumors. However, this process can be time-consuming and depends on the assessing doctor's expertise.

These limitations underscore the need for fully automated deep learning methods for brain tumor classification. Convolutional neural networks (CNNs) are highly suitable for analyzing medical images like brain scans. CNNs are a type of deep learning that is able to learn visual features through convolutional and max pooling layers.

Recent research has frequently utilized transfer learning with pre-trained models to detect and classify brain tumors. Deep transfer learning techniques show strong potential for automatically identifying all brain tumor types using medical images. Automated analysis through transfer learning could streamline diagnosis and help optimize treatment planning.

In this thesis, my aims to improve the classification of brain tumors by combining the power of two pretrained convolutional neural network (CNN) architectures which are MobileNetV3 and InceptionV3. Leveraging a dataset obtained from Kaggle, consisting of brain images with and without tumors. This research seeks to improve the accuracy and robustness of brain tumor classification models. The proposed methodology involves fine-tuning and integrating the combine pretrained model to harness their complementary strengths in feature extraction and classification.

CHAPTER 1

INTRODUCTION

1.1. Background

As the central organ of the nervous system, the brain controls and manages critical bodily functions and daily activities. It receives sensory inputs from throughout the body, interprets and processes this information, and relays output signals to direct muscle movement accordingly. However, the formation of a brain tumor due to uncontrolled cell growth can significantly disrupt normal brain functioning. A brain tumor results from abnormal cell division within the brain that produces an excess population of cells. As this mass develops, it can damage surrounding healthy brain tissue and impair neurological processes. Some common symptoms caused by brain tumors include headaches, difficulties with memory and cognition, nausea, changes in personality or behavior, vision problems, and impaired speech. As the tumor progresses, it can increasingly impact a patient's mental capacities and thought patterns as the brain region(s) it affects govern specific neurological domains. The physical changes induced by a growing brain tumor often manifest both physically and cognitively for the patient. Proper diagnosis and treatment are important to help address symptoms, slow further tumor growth, and minimize functional impacts on the patient.

Benign and malignant brain tumor types can be broadly classified. Tumors that are not dangerous, also known as non-cancerous, are confined to the brain and exhibit less aggressive behavior compared to malignant tumors. Malignant tumors, on the other hand, are cancerous and have the ability to rapidly invade various parts of the body. The diagnosis and classification of brain tumors hold significant importance in medical research and healthcare. These tumors represent a diverse group of neoplastic growths originating within the central nervous system, encompassing a wide range of pathological and molecular subtypes. The challenges associated with brain tumors extend beyond their intricate diversity. They also have the potential to profoundly impact a patient's quality of life and overall survival. Therefore, accurate and timely assessment of brain tumors is paramount, demanding innovative approaches that combine cutting-edge technology with medical expertise. Automated brain tumor detection using machine learning (ML) algorithms offers a potential solution. However, distinguishing between meningioma, pituitary, and glioma tumors poses a more formidable task due to the substantial variations in their sizes, shapes, and intensities. Notably, these tumor types account for the highest incidence rates among all brain tumors. Magnetic resonance imaging (MRI) plays a crucial role in automatic medical image analysis, providing high-resolution information about brain structures. MRI techniques are extensively employed for the identification and study of brain tumors. In recent years, numerous automated methods have been developed for detecting and classifying brain tumors from MRI images. These methods leverage the power of machine learning and deep learning algorithms to analyze MRI data, aiding in the early detection and accurate diagnosis of brain tumors.

1.2. Problem Statement

Machine learning algorithms like Multi-Layer Perceptrons (MLPs) and Support Vector Machines (SVMs) have commonly been applied to detect brain tumors from medical images. However, deep learning techniques have significant advantages for this task. Deep learning enables the creation of hierarchical feature representations through the use of multiple processing layers. It extends traditional neural networks with the addition of numerous hidden layers between the input and output. This deeper structure allows for the learning of richer, more abstract feature representations from raw input data. In the domain of medical imaging analysis, deep learning has been widely utilized to tackle challenges involving image denoising, segmentation, and classification. For classification tasks like tumor detection, deep learning models can learn predictive visual features directly from pixel data, like MRI and CT scans, without relying on manual feature engineering. By leveraging large datasets and hierarchical feature extraction through many network layers, deep learning approaches have achieved state-of-the-art performance on various medical image analysis problems. For brain tumor classification in particular, deep convolutional networks have been demonstrated to outperform traditional machine learning classifiers by automatically discovering the most salient features for discrimination from imaging data alone.

In recent years, convolutional neural networks (CNNs) have emerged as the most widely adopted deep learning (DL) framework for accomplishing complex tasks. The CNN architecture comprises pooling layers, fully connected (FC) layers, and feedforward layers equipped with convolutional filters. One significant application of CNNs is the development of fully automated classifiers for brain tumor classification. CNN-based models offer superior speed compared to traditional machine learning (ML) algorithms, as they eliminate the need for manual feature extraction. However, constructing CNN models in the medical imaging domain presents several challenges. Firstly, medical image datasets are often limited in size due to the time-consuming nature of labeling each image. Secondly, training CNNs on small datasets can be challenging and may lead to overfitting. To address these challenges, transfer learning (TL) and fine-tuning techniques offer viable solutions. Transfer learning involves training a deep learning model on a large base dataset and then applying the acquired knowledge to a smaller target dataset. This approach enables the model to leverage the knowledge learned from the base dataset and adapt it to the specific task of brain tumor classification. Fine-tuning further enhances the performance of the pre-trained model by adjusting its parameters on the target dataset. By fine-tuning the model, it can learn the specific features and patterns present in brain tumor images, leading to improved classification accuracy. Within this context, I will explore a novel approach to brain tumor classification which is a methodology that harnesses the power of CNNs by combining multiple pretrained CNN models to unlock new dimensions of accuracy and reliability.

1.3. Scope and Objectives

Brain tumor detection and classification cannot be fully automated because of a lot of methods that defined tumor regions manually. I used the standard MRI image from Kaggle dataset to classify brain tumor such as tumor brain (meningioma, pituitary, glioma) and non-tumor brain. To enhance the accuracy and efficiency of brain tumor classification, I employ a sophisticated fusion of two powerful neural network architectures which are InceptionV3 and MobileNetV3. This cutting-edge approach leverages the strengths of both networks to create a robust and versatile system for identifying and categorizing tumors in the brain. The goal is to improve the process of transfer learning (TL) for classifying brain tumors. In this paper, I present the overall accuracy, precision from the new convolution neural networks (CNN) model by combining two pre-trained convolution neural networks (CNN) models. The following are some of the major contributions of my study:

- Checking and confirming the transfer learning (TL) concept for various deep neural networks.

- Checking and presenting the overall accuracy, precision from the combine convolution neural networks models.

My main goal is trying to improve the process of transfer learning (TL) for classifying brain tumors that can determine brain tumor. Collecting the brain tumor datasets is the first step in any medical image project. Here, I try to improve the process of transfer learning (TL) by combining two pre-trained CNN models which are InceptionV3, MobilenetV3 to detect of brain tumor.

1.4. Assumption and Solution

Based on combining two pre-trained convolution neural networks (CNN) models, I propose a technique that can segmentate and detect the brain tumor by comparing the accuracy, sensitivity, and additional performance metrics.

Within this research article, an innovative approach is presented to enhance the transfer learning (TL) process for the classification of brain tumors. This upgraded model will be necessary in the development of advanced deep learning models designed to identify brain tumors, with the potential for further refinement to significantly enhance the future effectiveness and accuracy in tumor classification. The ability to identify tumors with a high degree of precision holds immense global significance, as it plays a vital role in determining treatment modalities, treatment success rates, and the ongoing management of the disease. This advancement stands to positively impact and potentially save the lives of numerous individuals worldwide from brain tumors.

1.5. Structure of thesis

Based on my research, my thesis has six chapters. Following, we'll quickly go over the foundation of the chapters.

- **Chapter 1. Introduction:** I will briefly outline the overview, scope and objectives, problem statement and the assumption and solution of the thesis in this chapter. Soon, I will describe and classify brain tumors as well.
- **Chapter 2. Literature Review and Related Works:** I will discuss some related work that was done in the past and discuss the working method, benefits, and drawbacks.

- **Chapter 3. Methodology:** This chapter approach to classifying the tumor using fundamental image processing techniques and detecting the tumor using transfer learning with combined pre trained of convolution neural networks (CNN) models.
- **Chapter 4. Implements and Results:** In this chapter I will present the obtained results of the combine pre-trained models.
- **Chapter 5. Discussion and Evaluation:** In this chapter I will discuss the experimental findings, our proposed algorithm for performance evaluation, and performance measures of combine pre-trained models.
- **Chapter 6. Conclusion and Future work:** In this chapter I will discuss about result that I have finished and present a future work to prepare for thesis.

CHAPTER 2

LITURATURE REVIEW

2.1. Convolutional Neural Network

Convolutional neural networks (CNNs) [1] are neural networks with one or more convolutional layers that are mostly used for image processing, classification, and segmentation. A Convolutional neural network (CNN) is a sort of artificial neural network that responds to overlapping regions in the input by stacking individual neurons. These neurons can be thought of as filters that slide on the height and width of the input, extracting different patterns found in the image at each region. Unlike a fully connected network, a CNN takes advantage of the 2D structure of input data such as images by enforcing a local connectivity pattern between neurons of adjacent layers. This allows CNNs to take into account the strong spatial correlation between neighboring pixel values in images.

Besides that, another advantage of CNNs is compositionality which allows neurons to stack on preceding neurons and helps the network to learn more deeper features in a hierarchical manner. The features extracted from early layers represent simple patterns like edges and textures, while the features extracted from later layers represent complex patterns. This allows the network to learn increasingly higher levels of abstraction, just like the human visual cortex. The principle of building higher-level features from lower level is ascribed to the fact that CNNs are so powerful in computer vision tasks as they can mimic the biological neural network structure of the human brain.

CNNs perform particularly well on image-related tasks as they automatically learn filters that in the early layers detect low-level features like edges and gradients, while in higher layers detect higher-level composite features like combinations of edges that represent parts of objects or even entire objects. By enforcing local connectivity, weight sharing, and spatial sampling, CNNs are better than regular neural networks at handling the variability encountered in images like translation, rotation and other geometric and photometric transformations. For these reasons, CNNs have achieved state-of-the-art results in many areas of computer vision such as image classification, object detection and segmentation. They have helped advance applications like automatic driving, medical imaging analysis and visual content search engines.

Convolutional neural networks consist of a series of layers that can extract features from images. Some of the most common layer types used when building convolutional neural network models are convolutional layers, activation layers, pooling layers, fully connected layers, dropout layers, and batch normalization layers. Convolutional layers apply filters to input data to extract features, while activation layers introduce nonlinearity to the network via functions like ReLU [2]. Pooling layers reduce the dimensionality of representations to help control overfitting, with max pooling being a popular choice. Fully connected layers connect every neuron in one layer to every neuron in the next layer and are typically used after flattening CNN output for classification tasks. Dropout and batch normalization layers help address issues like overfitting as well. Dropout randomly drops units and their connections during training to prevent units from co-adapting too much, whereas batch normalization standardizes layer inputs to speed up training. By incorporating various combinations of these common layer types, convolutional neural network models are able to learn rich hierarchical feature representations of images that enable solving complex visual tasks

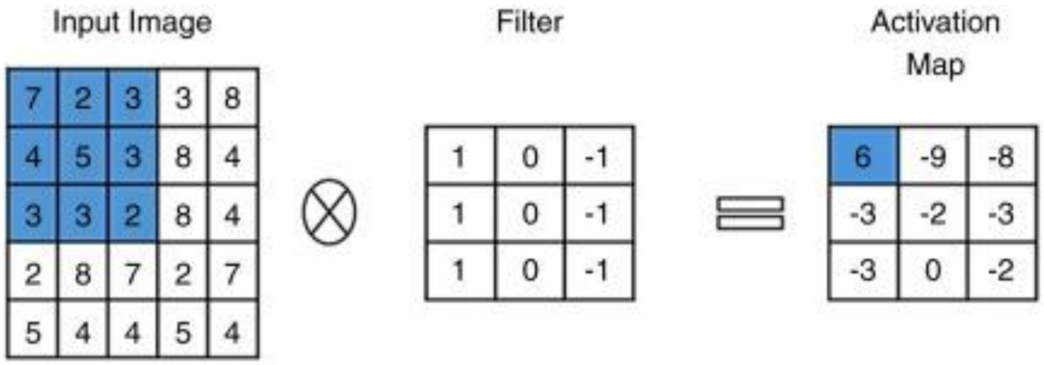
2.1.1. Convolutional layer

The convolutional layer is the primary computational unit of a convolutional network. It performs the bulk of parameter learning and feature extraction in the model. By using local connections and shared weights between units, convolutional layers aim to detect patterns in the input and learn translation invariant features. This helps address overfitting by vastly reducing the number of parameters compared to a fully connected network of the same size.

In a convolutional layer, small, typically 3x3 or 5x5, filter kernels are convolved across the width and height of the input volume. At each spatial position, the dot product is taken between the filter and the input region it is connected to. This sliding filter operation produces a 2D activation map of that filter at the output. Multiple filters with different weights are applied, learnings different features. Stacking all these activation maps along the depth dimension composes the volume output of the convolutional layer.

The network can learn hierarchical representations from low to high-level features as data passes through many convolutional and pooling layers. Early layers may detect edges and basic patterns, while deeper layers assemble these into more complex shapes and objects. By recursively applying this procedure of local filters, activations, and subsampling, convolutional neural networks are able to extract fundamental attributes within the input that are invariant to translation, rotation or small distortions.

As an example, figure 1 shows a simple 2D 5x5 input being convolved with a 3x3 filter. The filter learns weights to detect a particular pattern and slides over the input to generate an activation map, which is part of the output volume of that layer in the network. Through backpropagation, these filters can be trained to optimize feature detection for the task.



7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

1	0	-1
1	0	-1
1	0	-1

6	-9	-8
-3	-2	-3
-3	0	-2

Figure 1. Example of convolutional layer. [3]

This is the general equation when we applied convolutional layer.

The output layer = Input * Filter

$$= \sum_{y=0}^{column}(\sum_{x=0}^{row}) Input(x-p, y-q) Filter(x, y)$$

2.1.2. Pooling layer

The pooling operation is a key component of convolutional neural networks. It entails sliding a filter across each channel of the feature maps produced by the previous convolutional layer and summarizing or reducing the values within the filter's coverage region.

There are several main purposes of incorporating pooling layers. First, it reduces the dimensions of the feature maps, thereby significantly decreasing the number of parameters and computations in the network. This helps control overfitting and improve training/inference speed. Second, it causes the network to learn invariant features, since small changes in input positions will not change the pooled output.

Pooling also summarizes the activities of nearby neuron clusters. This has the effect of emphasizing the most prominent features detected while suppressing weaker ones. By subsampling the representation, the network can better deal with variance in scale and position.

The most common types of pooling are max and average pooling. Max pooling takes the maximum value within the filter region, effectively retaining the strongest activation. Average pooling calculates the average value, allowing multiple active neurons to jointly signal a feature detection.

Global pooling squeezes the entire spatial dimension into a single value per channel. This fully convolutional networks to handle inputs of any size during inference.

Overall, by interleaving pooling with convolutional layers CNNs gain robustness to image deformations and reductions in input dimensionality needed for subsequent processing. With pooling introducing nonlinearity and invariance on the spatial side of computations, it is a crucial component enabling CNNs to discern visual patterns.

2.1.3. Fully connected layer (FC layer)

In a neural network architecture, fully connected layers refer to those layers where each neuron is connected to all activations in the previous layer. This dense connectivity enables neurons to learn relationships between features extracted by earlier stages of the network.

Most convolutional neural network models deployed for computer vision tasks incorporate one or more fully connected layers at the end of the convolutional and pooling sections. At this point in the network, the spatial dimensions of the activations have typically been reduced through subsampling, resulting in a flattened feature vector representing the input.

The fully connected layers serve to learn high-level patterns and classifications by analyzing how lower-level features extracted by convolutional filters combine together across the image. They allow interactions between features detected in different spatial regions which may have become separated due to prior convolution and pooling stages.

Typical CNN architectures apply a final fully connected layer before using either a softmax [4]or sigmoid [5] classifier to generate predictions. The softmax layer computes a probability distribution over class labels for single-label classifications problems. Sigmoid can be used instead for multi-label predictions, where multiple classes may apply to a single input.

By enabling dense interconnections between neurons processing the entire feature representation, fully connected layers help CNNs classify images and regress real-valued outputs. They are an important component utilized by most state-of-the-art deep learning models for computer vision and other domains. Figure below presented example of fully connected layers in Convolutional Neural Networks.

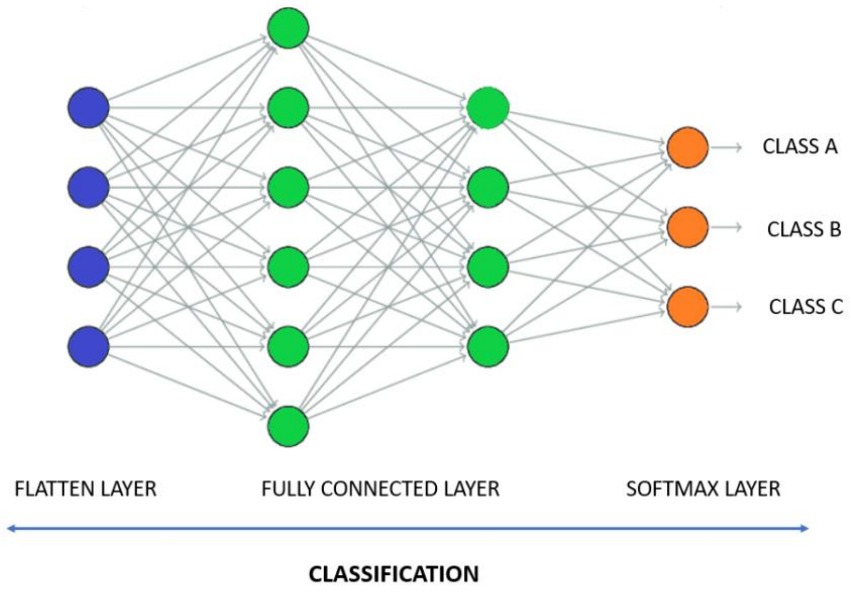


Figure 2. Example of fully connected layers in Convolutional Neural Networks [6]

2.1.4. Activation layer

Activation functions play a vital role in the functioning of artificial neural networks. They are applied to the weighted sum of inputs computed by each neuron in order to introduce nonlinearity. This allows neural networks to learn complex patterns from the data that would otherwise be impossible to model with just linear functions.

While activations do not typically undergo training themselves, they determine how information moves through the network and whether neurons fire or not. The choice of activation greatly impacts the network's expressive ability and trainability.

Some key properties activation functions provide include non-linear transformation of inputs, ranged outputs bounded within a spectrum like 0-1, and differentiability to allow backpropagation of errors during gradient descent optimization.

Strictly speaking, activation functions are not considered full network layers since no learnable parameters are associated with them directly. However, they serve to introduce nonlinearity between each layer and transform the neural network into a vastly more powerful model than a simple linear perceptron.

The three most commonly used activation functions include the sigmoid activation, which outputs values between 0-1 but saturates and has gradients that vanish for large inputs, the tanh activation that outputs values between -1 to 1 but also saturates, and the ReLU (rectified linear unit) activation, which is linear for positive inputs and outputs zero otherwise. The ReLU helps address the problems of saturation that sigmoid and tanh encounter and has been shown to accelerate the training of neural networks in comparison.

2.1.5. Softmax function

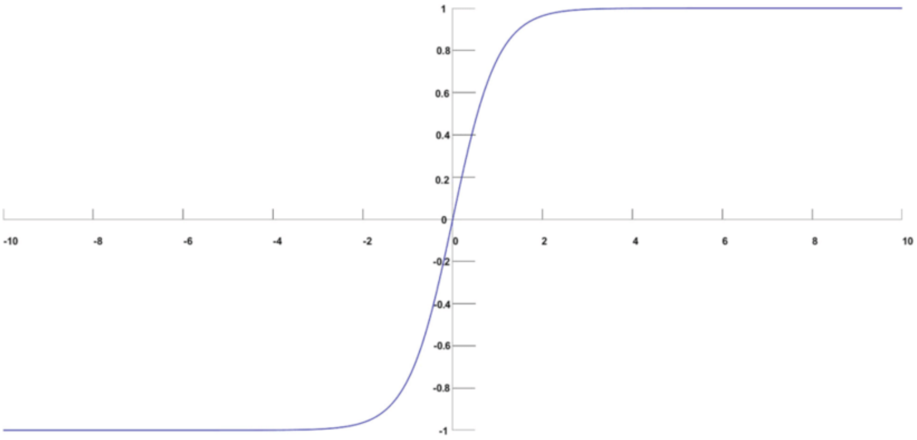


Figure 3. Softmax Function [7]

In the field of machine learning and deep learning, the softmax activation function [4] serves a particularly important purpose of calibrating model predictions for classification problems. As neural networks are tasked with learning complex patterns to discriminate between multiple classes, it is essential to transform their raw numerical outputs into a probabilistic format to indicate confidence in predictions.

The softmax function fulfills this role almost universally in neural network architectures developed for classification tasks. It is typically applied as the final layer of the model, where the network has encoded features and made inferences about the input, but its outputs must be normalized before reporting predictions. The raw output vector at this stage contains uncalibrated scores relating to each class, which if left as-is could not indicate the likelihood of class membership.

To enable probabilistic predictions, the softmax transforms these scores into a valid probability distribution by applying an exponentiation and normalization process. This calculates the likelihood of each class while ensuring the probabilities sum to one. As a result, the softmax allows a classifier to produce an interpretable output even from arbitrary real-valued score vectors learned through its parameters.

By squashing the scores into the range [0, 1] and enforcing them to add up to 1, the softmax transformation provides the calibration needed for classification problems. It remains a staple activation for the final layers of neural networks, allowing such models to effectively discriminate between multiple classes based on the confidence levels of their predictions. The softmax function is a mathematical operation that exponentiates each input score and normalizes the results. Let Z_i denote the raw score for class I, the softmax activation a_i for class i is computed as follows:

$$a = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Here, e^{z_i} represents the exponential function applied to the raw score of class i, and the denominator is the sum of exponentiated scores across all classes. The softmax function ensures that the output probabilities are in the range [0, 1] and collectively sum to 1, facilitating the interpretation of the network's predictions as probabilities. In the context of a thesis report, understanding the intricacies of the softmax function is crucial for comprehending the output layer's role in converting model predictions into meaningful probabilities for decision-making.

2.1.6. The process of normalizing batches.

The process of normalizing batches, also know as batch normalization is an important technique in deep learning that was introduced in 2015 [8] to help accelerate neural network training and improve model stability. Proposed by researchers Sergey Ioffe and Christian Szegedy. Batch normalization helps address the phenomenon known as "internal covariate shift" that can slow down training.

During training, as the distributions of network layers' inputs change on each iteration due to weight updates it forces the subsequent layers to continually adapt. This internal covariate shift makes the deeper layers' optimization difficult and slows convergence. Batch normalization helps address this by normalizing layer inputs to maintain a stable distribution with zero mean and unit variance.

We can compute the batch normalization from these equations:

- $\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta{}^2 + \varepsilon}}$; where $\hat{x}_i$ is denoted as output of the layer and μ_β , σ_β are computed over mini-batch. Besides that, ε is set to be e^{-7} to avoid dividing by zero
- $\mu_\beta = \frac{1}{M} \sum_{x=1}^{m} x_i$; where x is considered as mini-batch of activations
- $\sigma_\beta{}^2 = \frac{1}{m} \sum_{i=1}^{m} \left(x_i - \mu_\beta\right)^2$.

2.1.7. Drop out

Dropout layers [9] have proven to be an effective regularization technique for reducing overfitting in neural networks. Introduced by researchers in 2012 , dropout helps improve model generalization by preventing co-adaptation on the training data.

The key concept behind dropout is that it randomly drops out, or sets to zero, a portion of the neuron outputs from the previous layer during forward passes. This prevents units from co-adapting too much. A hyperparameter called the drop probability determines the percentage of neurons to drop out.

By varying the network architecture randomly at each training batch, dropout discourages reliance on specific feature combinations. It forces the network to learn redundant representations by preventing arbitrary feature combinations from working together. This enlarges the effective size of the training dataset.

During test time, all units are used but their outputs are scaled down by the drop probability. This ensemble-like effect reduces reliance on any single neuron and encourages the learning of robust features.

Numerous studies have demonstrated that dropout substantially improves generalization for both convolutional and recurrent networks by curbing overfitting. It boosts test set accuracy by changing the network dynamically during training. The dropout layer ensures multiple nodes remain active, increasing model robustness and generalizability to new data instances. Overall, dropout is a powerful and widely adopted regularization approach.

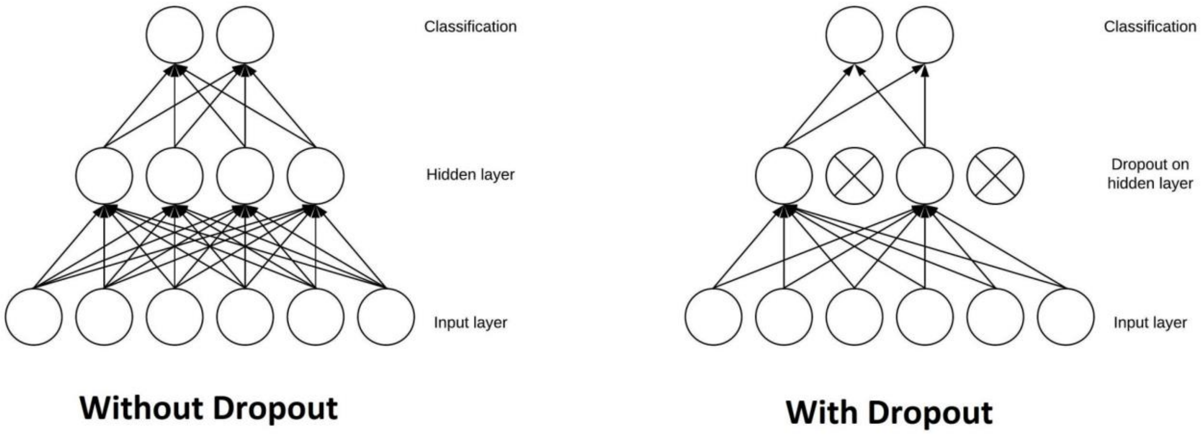


Figure 4. Example of neural network after applying Dropout layer. [10]

2.2. Transfer learning

Transfer learning [11] is a machine learning method that improves learning in a new target task by leveraging knowledge from a related source task that has already been learned (Pan & Yang, 2009). It is particularly useful when the target task has limited labeled training data available, as is common in many real-world applications.

Traditional machine learning approaches train models from scratch on each new task. However, this often requires large-labeled datasets that may be difficult or expensive to acquire. Transfer learning addresses this problem by utilizing models pre-trained on a source task with abundant data, and fine-tuning [12] them on the target task.

Transfer learning and traditional machine learning both aim to build models that can learn from data, but they differ in important ways. Traditional machine learning approaches train models from scratch on a specific task using its own dataset. In contrast, transfer learning leverages knowledge gained from training on one task and applies it to a different but related task.

In traditional machine learning, models like neural networks are trained on large-labeled datasets for a single problem. For example, a neural network may be trained on a labeled image dataset to classify photos into categories like cats, dogs, etc. The model learns features and patterns directly from the given dataset through its many parameters.

Transfer learning takes a pre-trained model usually developed on a very large-labeled dataset for a task like image classification, and "repurposes" it for a new related task with fewer training samples. For example, a model pre-trained on a massive-labeled image dataset can be used as a starting point for a new task like classifying medical images into disease categories. Only the final layers of the pre-trained model need to be retrained on the new smaller dataset, while keeping earlier feature extraction layers intact. This helps the model learn meaningful features from limited new data by leveraging what was learned from vast amounts of prior data.

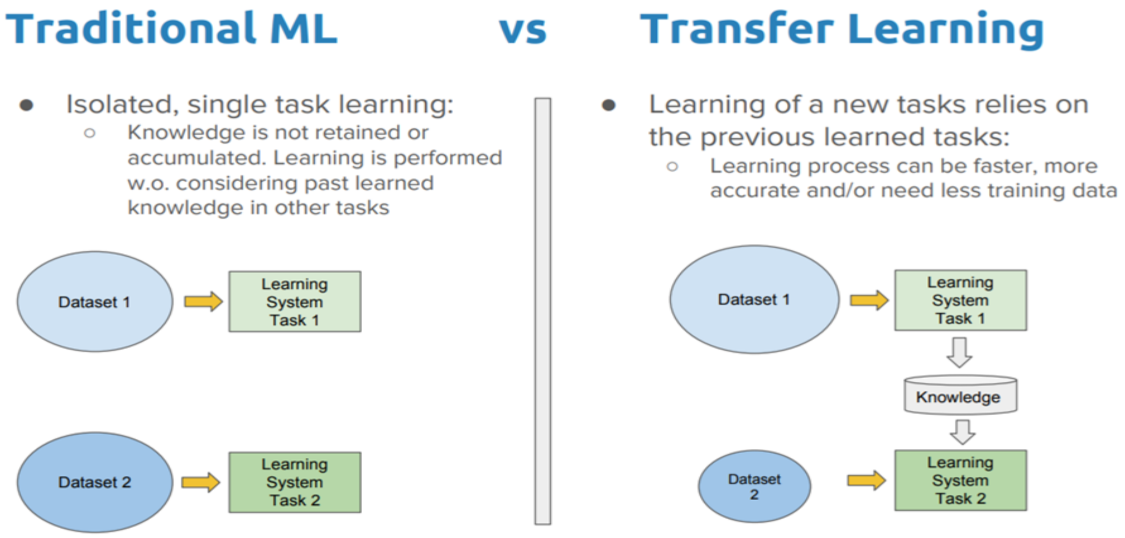

Traditional ML
vs
Transfer Learning
Isolated, single task learning:
Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks
Learning of a new tasks relies on the previous learned tasks:
Learning process can be faster, more accurate and/or need less training data
Dataset 1
Learning System Task 1
Dataset 2
Learning System Task 2
Dataset 1
Learning System Task 1
Knowledge
Dataset 2
Learning System Task 2

Figure 5. Comparison of machine learning and transfer learning [13]

Two popular convolutional neural network architectures (MobileNetV3, InceptionV3) that have proven highly useful for transfer learning through pretrained models.

2.2.1. InceptionV3 pre-trained CNN model

InceptionV3 [14] is a pre-trained convolutional neural network (CNN) model developed by Google that combines the strengths of the Inception architecture with residual connections. This unified approach results in a powerful and adaptable deep learning solution for computer vision tasks.

The Inception module has demonstrated an ability to simultaneously capture multi-scale image features through its parallel convolutional paths. However, the severe vanishing gradient problem during training can hamper optimization. InceptionV3 addresses this by integrating residual connections between layers, which effectively facilitate gradient flow.

Comprising an extensive 164 layers, InceptionV3's comprehensive architecture efficiently distributes computational loads across multiple convolutional branches within each module. This excels at both classifying images into categories as well as extracting discriminative features from raw pixel values.

Google trained InceptionV3 on large-scale datasets such as ImageNet [15] containing over 1 million photos spanning 1000 classes. This endows the model with profound proficiency in visually recognizing and differentiating an immense breadth of object classes.

Significantly, the knowledge gained from self-supervised pre-training on vast amounts of labeled image data translates to compelling feature representation capabilities. Researchers and practitioners can leverage these universal representations by fine-tuning only the classification layers for new domains with limited annotated examples. In summary, InceptionV3's fusion of Inception modules and residual connections in an extensively deep network, coupled with its powerful pre-trained weights, makes it a valuable asset for tackling image classification and feature extraction problems across computer vision applications. Figure below present the architecture of InceptionV3 pre trained model

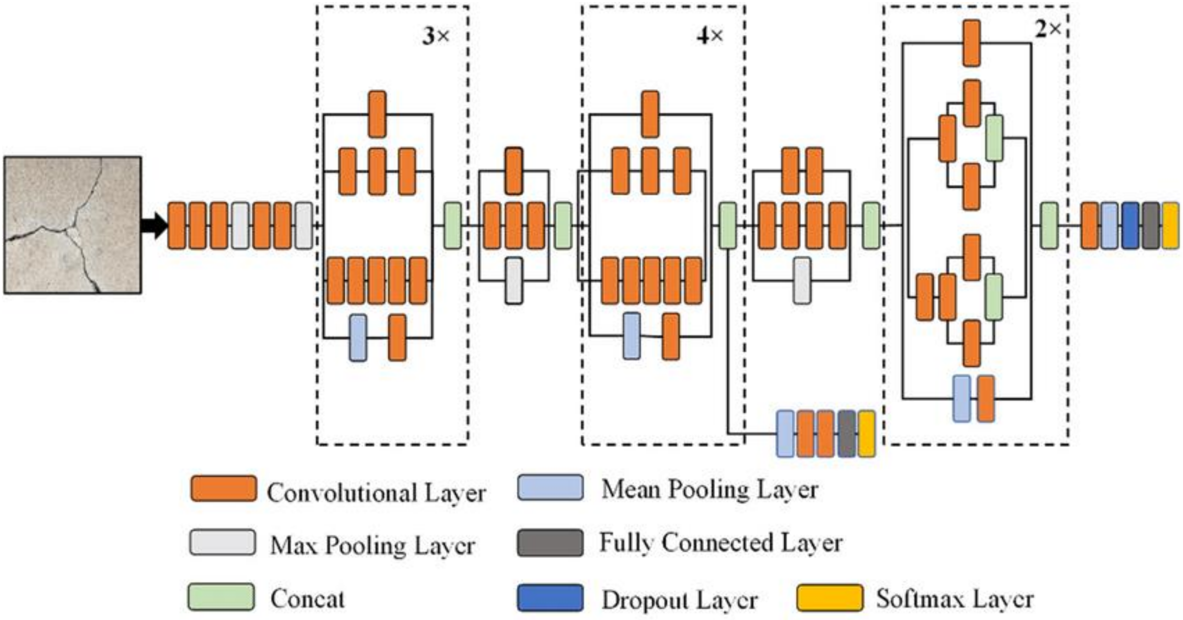


Figure 6. Architecture of InceptionV3 pre trained model [16]

2.2.2. MobileNetV3 pre-trained CNN model

MobileNetV3 [17] is a highly efficient pre-trained CNN model for mobile and Edge applications. MobileNetV3 is a convolutional neural network model developed by Google for deployment on mobile and embedded devices with stringent resource constraints. As the latest iteration in the MobileNet family, MobileNetV3 builds upon the successes of its predecessors, MobileNetV1 and MobileNetV2, in achieving superior performance with reduced parameters.

Central to MobileNetV3's architecture are innovative techniques such as inverted residuals and squeeze-and-excitation blocks that boost efficiency. It comes in two variants optimized for different use cases - MobileNetV3 Large prioritizes accuracy while MobileNetV3 targets low latency applications.

MobileNetV3's design places strong emphasis on achieving state-of-the-art results in computer vision tasks like image classification, object detection and segmentation, within the computational limitations of mobile processors and edge devices. At 53 layers deep, it efficiently captures hierarchical visual representations from input images. Prior to deployment, Google extensively trained MobileNetV3 on the ImageNet dataset containing over 14 million labeled images across 1000 object classes. This endows the model with the ability to classify images into these categories based on learned visual features.

By leveraging knowledge gained from vast pre-training, MobileNetV3 offers a performant pre-trained starting point well-suited for downstream transfer learning on mobile and Internet-of-Things platforms. With its focus on delivering robust deep learning under strict power and memory constraints, MobileNetV3 presents a compelling option for developers of real-time embedded vision applications. Figure below show the architecture of MobileNetV3 pre trained model.

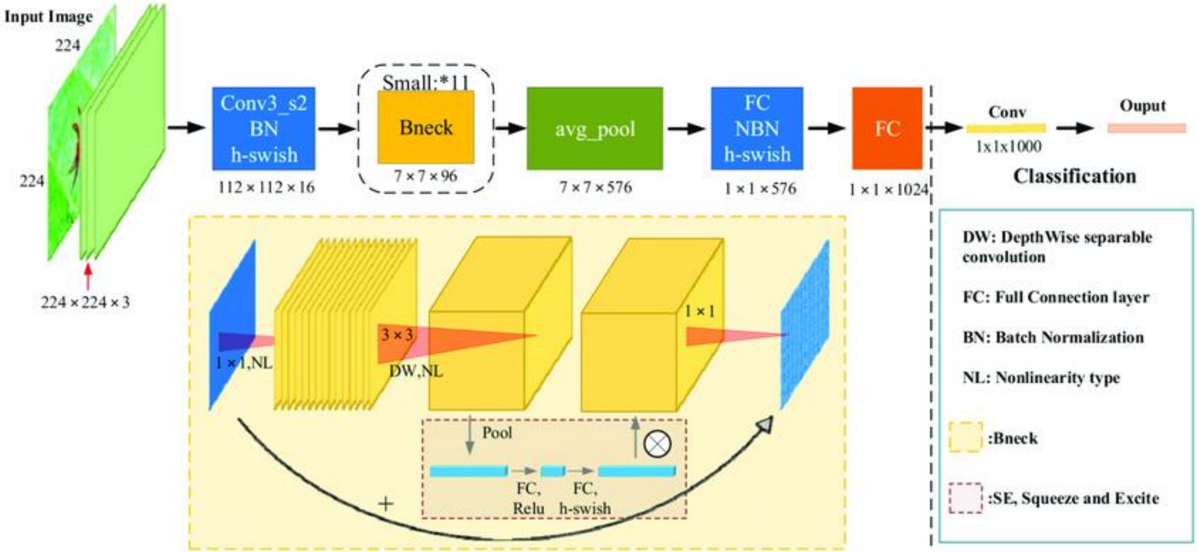


Figure 7. Architecture of MobileNetV3 pre trained model [18]

2.3. Tumor Detection in MRI Images: An Approach Based on Watershed and Threshold-Based Segmentation

This research [19] was published on March 2020 by Ghinwa M. Tarhini and Reda Shbib. In this research Ghinwa M. Tarhini and Reda Shbib proposed Threshold-based segmentation method and Morphological Operations method to classify brain tumor from Magnetic Resonance Imaging (MRI) image.

The flow of work for this research follows these steps. Firstly, all input images were collected from 155 MRI images scanned, then the image was converted from RGB to a greyscale image. These images are displayed as gray-scale images. In addition, this input image had a two-dimensional matrix and the pixel values of each image were between 0 and 255. Then, these images were reduced in noise by applying many filters, like Gaussian averaging filters. Besides that, the authors detected the sharp edges and the boundary of the tumor by using Gaussian high-pass filters. After these steps, the image had a higher quality than the original.

Secondly, the authors employed a multi-stage segmentation approach combining thresholding, watershed transformation, and morphological operations to partition the MRI images into homogeneous regions. Threshold segmentation [20] was first applied to convert the input grayscale images into binary images by selecting an adaptive intensity threshold. This initial segmentation separates pixels above and below the threshold. Next, watershed transformation [21] was utilized for further segmentation. Watershed approaches are well-suited for grouping pixels based on intensity gradients, however this is often show result in over-segmentation. Here, watershed segmentation was primarily used to evaluate the outputs of the previous thresholding stage rather than directly segment the input images. Finally, morphological operations were applied after watershed segmentation to detect and separate the tumor regions. Morphological operations such as erosion and dilation are frequently employed in medical image analysis tasks to extract shape features [22]. In this study, morphological opening and closing operations aided in distinguishing tumor pixels from other tissues.

This multi-step approach combining thresholding, watersheds, and morphology allowed for segmentation of the MRI images into regions exhibiting similar radiomic characteristics, facilitating the identification of tumor tissues. The selection of these established segmentation techniques was aimed at partitioning the images while preserving important radiological features for the subsequent classification task. After implementing the multi-stage segmentation pipeline consisting of thresholding, watershed transformation, and morphological operations, the authors visually evaluated the segmentation results. Figure 8 displays an exemplar input MRI slice and shows the corresponding output of the segmentation algorithm.

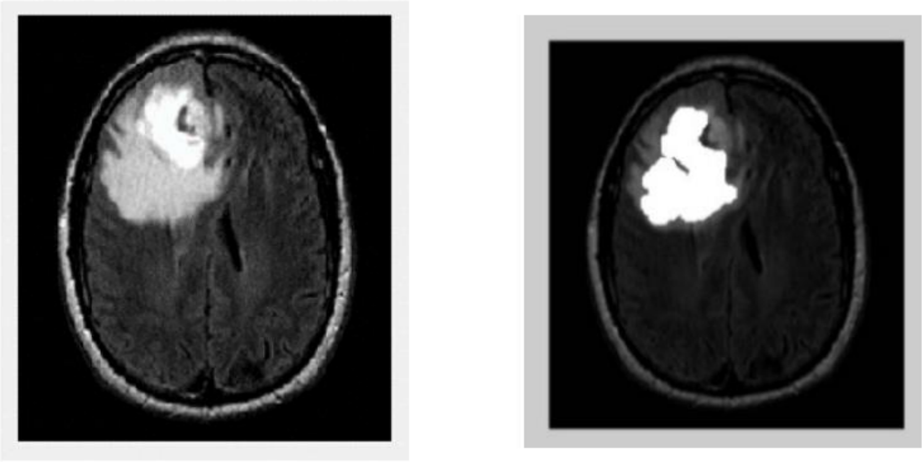


Figure 8. The input MRI image (left), the output image of the segmentation algorithm (right)

To evaluate the performance of the proposed algorithm, comparisons were made between the predictions of the algorithm and the ground truth measurements. The authors used the following metrics which is commonly used in machine learning and computer vision for classification tasks [23] were evaluated:

- True Positives (TP): The number of tumor pixels that were correctly identified as tumors by the algorithm.
- False Positives (FP): The number of non-tumor pixels that were incorrectly identified as tumor pixels by the algorithm.
- True Negatives (TN): The number of non-tumor pixels that were correctly identified as non-tumor pixels by the algorithm.
- False Negatives (FN): The number of tumor pixels that were incorrectly identified as non-tumor pixels by the algorithm.

Overall prediction accuracy was measured using:

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

Table 1 [19]. The accuracy of the proposed algorithm.

Method	Accuracy
Proposed Algorithm	0.888

2.4. Brain Tumor Classification with Deep Learning Neural Networks

This research [24] was published in June 2018 by H. Mohsen, El-Sayed A.El-Dahshan, El-Sayed M.El-Horbaty, and A.B. M.Salem. Mohsen et al. and his team used a Deep Neural Network to classify a dataset which have 66 medical images (MRI images) of the human brain to classify into four categories.

Mohsen et al. and his team developed a deep learning framework for automated brain tumor classification from MRI images. For data acquisition, Brain tumors exhibit significant heterogeneity, with over 120 histological varieties documented by the World Health Organization (WHO). The dataset compiled by Mohsen et al. comprised 66 MRI scans of the human brain selected to represent the four most common subclasses - normal tissues, glioblastoma, sarcoma, and metastatic bronchogenic carcinoma.

These images were obtained from publicly available archives of Harvard Medical School, which contains a large collection of MRI brain scans. By sourcing cases from this well-established repository, the study aimed to utilize representative imaging data reflective of variations observed in clinical practice.

A total of 66 scans were thus collated to capture diversity in appearances across both healthy and tumor-bearing anatomies. This sample size also allowed for sufficient data to train sophisticated deep learning models while balancing multiple diagnostic groups. The curated MRI collection served as the foundation for developing and validating the proposed methodology.

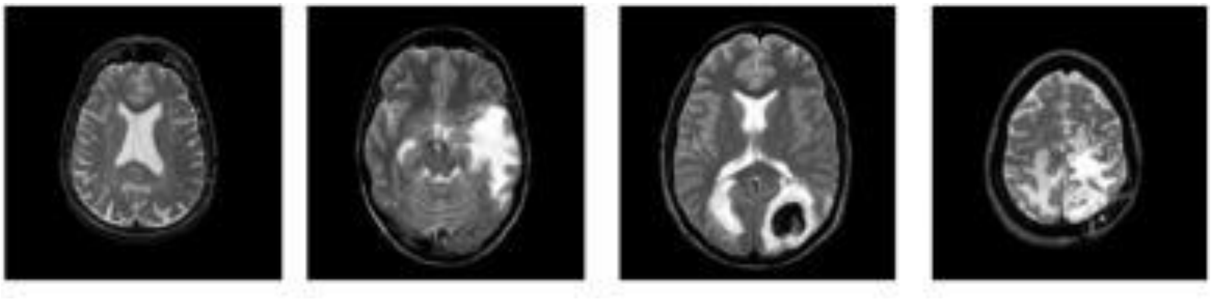


Figure 9. Example of brain MRIs image. (a) Normal (b) Metastatic bronchogenic carcinoma (c) Sarcoma (d) Glioblastoma (From left to right) [24]

Image segmentation is a crucial step in image processing and computer vision applications, including brain MRI analysis. However, segmenting brain MRI images poses significant challenges due to the need for accurate separation of tumor tissues from cerebrospinal fluid, gray matter, white matter and the skull. In this study, the author employed the Fuzzy C-means technique for image segmentation of brain MRI. The figure below illustrates a sample of brain image segmented using the Fuzzy C-means (FCM) clustering technique.

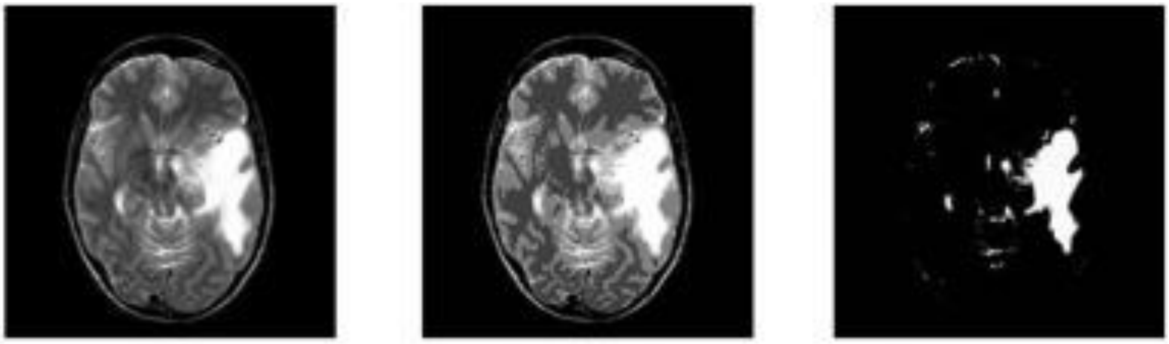


Figure 10. A sample of brain image segmented using FCM. (a) Original image; (b) Segmented Image using FCM; (c) Segmented tumor using FCM (From left to right) [24]

After using the Fuzzy C-means clustering technique (FCM) to segment the brain MRI image into five sections. These images are extracted by using the discrete wavelet transform (DWT) technique. Discrete wavelet transform (DWT) is a transformation that divides an input signal into several sets and this technique have a strength to extract fitting the features for any directions because this technique used high-pass and low-pass filter banks which are cascaded to extract features in a hierarchy.

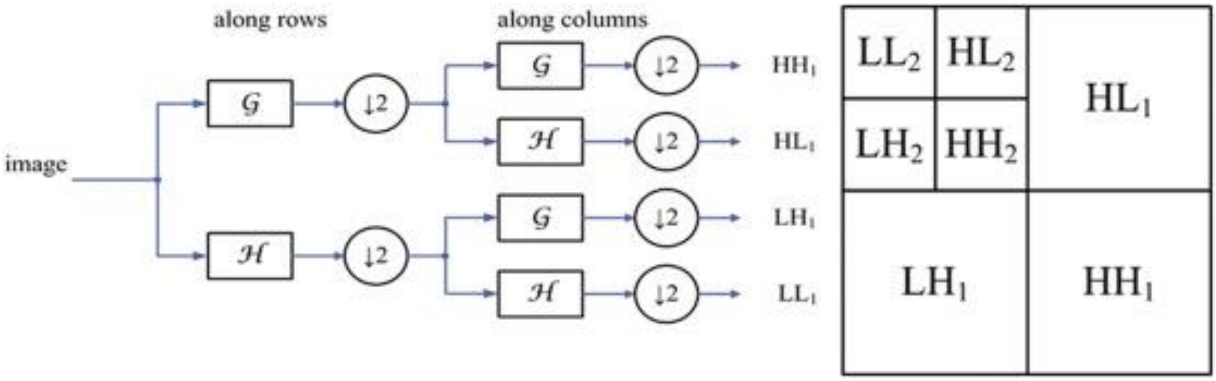


Figure 11. 2-levels DWT decomposition of an image. [24]

Figure above shows the functions h(n) and g(n) represent the coefficients of the high-pass and low-pass filters, respectively of a 2-levels DWT decomposition. The author extracted 1024 features for each brain MRI image by using a 3-levels decomposition of Haar wavelet. After that, the author utilized principal component analysis to estimate the initially extracted features. After following all of previous steps successfully, the author used deep neural network (DNN) to perform brain tumor classification on the resulting feature vector with the extraction of each the feature.

Furthermore, the author built and trained the deep neural network which have seven hidden layers by using 7-fold cross validation technique. Then the author estimated the selected classifier by utilizing alternative machine learning classification algorithms available in WEKA. The remaining machine learning classification algorithms include K-nearest neighbors (KNN) with K values of 1 and 3. The method of linear discriminant analysis (LDA) and the support vector machine (SVM) algorithm implemented as sequential minimal optimization (SMO-SVM).

Finally, the authors evaluated the result evaluated using metrics such as average classification rate, average recall, average precision, average F-Measure, and average area under the ROC curve (AUC). After comparing performance of all algorithms, the deep neural network (DNN) technique gave the beat result to classify the brain tumor.

Table 2 presents the performance metrics of several classification algorithms in the context of the research study. The evaluated algorithms include Deep Neural Network (DNN), K-Nearest Neighbors with K values of 1 and 3 (KNN K = 1 and KNN K = 3), Linear Discriminant Analysis (LDA), and Support Vector Machine with a polynomial kernel (SMO). The table provides valuable insights into the accuracy and efficiency of each algorithm, enabling researchers to compare and evaluate their performance for the specific task at hand. Figure 12 shows a comparison graph that illustrates the performance of different classification algorithms. The evaluated algorithms include Deep Neural Network (DNN), K-Nearest Neighbors with K values of 1 and 3 (KNN K = 1 and KNN K = 3), Linear Discriminant Analysis (LDA), and Support Vector Machine with a polynomial kernel (SMO).

Table 2. Performance of DNN, KNN K = 1 and 3, LDA and SMO. [24]

Algorithm	Classification rate	Recall	Precision	F-Measure	AUC (ROC)
DNN	96.97%	0.97	0.97	0.97	0.984
KNN (K =1)	95.45%	0.955	0.956	0.955	0.967
KNN (K =3)	86.36%	0.864	0.892	0.866	0.954
LDA	95.45%	0.955	0.957	0.955	0.983
SMO	93.94%	0.939	0.941	0.963	0.939

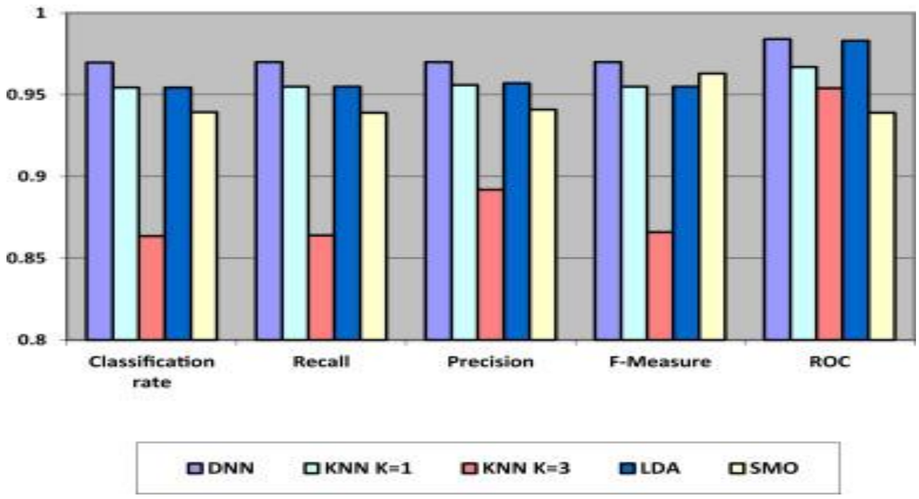


Figure 12. A comparative graph depicting the performance of DNN, KNN (K = 1 and 3), LDA, and SMO.[24]

2.5. Genetic Method with SVM Classifier for Brain Tumor Segmentation

This research paper [25] was published in March 2016 and presented a novel approach for brain tumor classification using a combination of Genetic Algorithm (GA) and Support Vector Machine (SVM) techniques. The research team, led by Dr. A.R. Kavitha, L. Chitra, and R. Kanaga, employed this hybrid technique to achieve accurate and efficient tumor classification. The proposed methodology involves several key steps. Firstly, the authors utilize a Wiener filter, a 2-D adaptive noise reduction technique, to preprocess the brain images and remove unwanted noise. The Wiener filter operates by analyzing each pixel's mean and variance within the image and subsequently generates a pixel-wise Wiener filter to effectively eliminate noise while preserving important image features.

This preprocessing step is crucial as it enhances the quality of the input images, leading to more accurate tumor classification in subsequent stages of the proposed technique. The authors' innovative combination of GA and SVM demonstrates their commitment to developing advanced medical imaging techniques for improved brain tumor diagnosis and treatment.

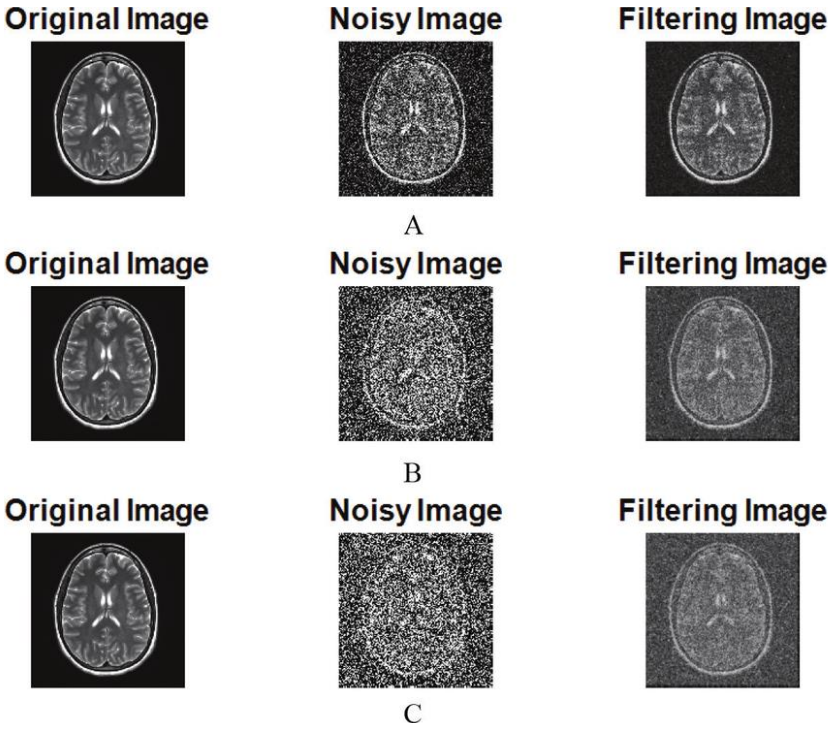


Figure 13. [25] Example of Wiener filter when applied it to the image

Secondly, the authors employed the Genetic Algorithm to partition the image into multiple regions. Building upon Goldberg's study [26] and other related research, an adaptive search procedure known as the Genetic Algorithm was utilized for this purpose.

Thirdly, the authors extracted the feature of the image from dataset to classify the stage of brain tumor by using Gray Level Co-occurrence Matrix (GLCM). After using Gray Level Co-occurrence Matrix (GLCM), the texture of image was characterized by estimating pairs of pixels with specified value.

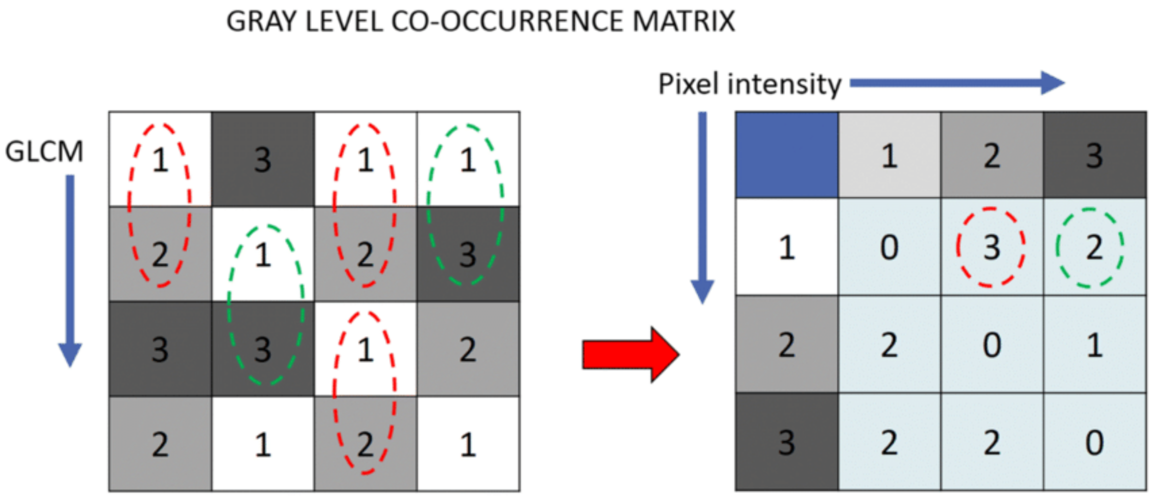


Figure 14. [25] Example of Gray-level co-occurrence matrix

Finally, the authors employed Support Vector Machine (SVM) for the identification of the brain tumor stage based on MRI images. The Support Vector Machine (SVM) utilized two primary classes for classification, namely linear and nonlinear. The fundamental objective of employing Support Vector Machine (SVM) was to establish a hyperplane among datasets, facilitating the delineation of their respective classes. Additionally, the purpose was to train the machine with pre-existing data and subsequently leverage Support Vector Machine (SVM) to identify an optimal hyperplane that maximizes the separation distance from the nearest data points.

After the authors applied wiener filter, reducing noise while preserving important image details. The overall approach was developed and tested within the MATLAB R2013a software environment. As shown in figure 15, the proposed preprocessing technique yields processed images with lower noise levels and improved clarity compared to the original images.

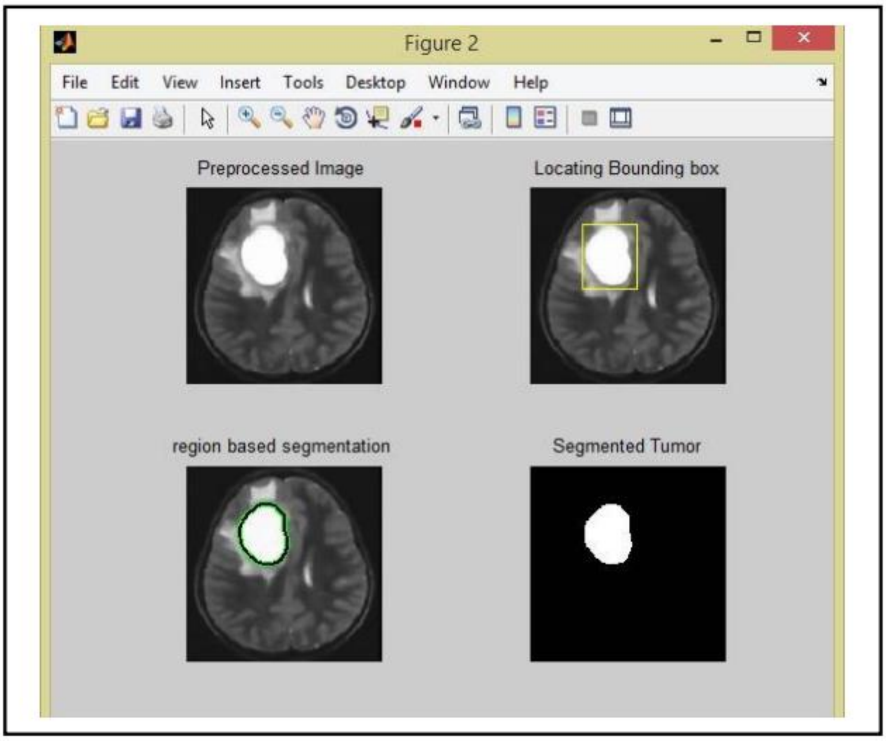


Figure 15. Image of the segmented brain image [25]

2.6. Fine-Tuning Convolutional Deep Features for the Classification of Brain Tumors in MRI Images

Convolution neural network (CNNs) become more popular in image processing because of it effective and high accuracy. But training a CNN is very challenging since it cost a lot of time. This research [27] was published in March 2017 by Kaoutar B. Ahmed, Lawrence O. Hall, Dmitry B. Goldgof, Renhao Liu, and Robert A. Gatenby. To solve this problem K.B. Ahmed and his team using pre-trained convolutional neural networks (CNNs) models then fine-tuned these CNN's models to improve the performance with a small dataset.

The proposed technique that the author used to detect the brain tumor in medical images include these steps. Firstly, the authors utilized an anonymous data collection which have supplied from the H. L. Moffitt Cancer Research Center. This dataset includes of twenty-two cases of brain tumor. Each case consists of four MRI image of brain. After that the authors separated twenty-two cases to two parts which are twelve short term of time survival and ten long term of time survival. Figure 16 illustrates a representative slice featuring with both short and long-term durations obtained from the Flair imaging sequence. The tumor mask delineation was carried out manually, with guidance by radiologists (R.G.) and using T1-weighted scans as a reference.

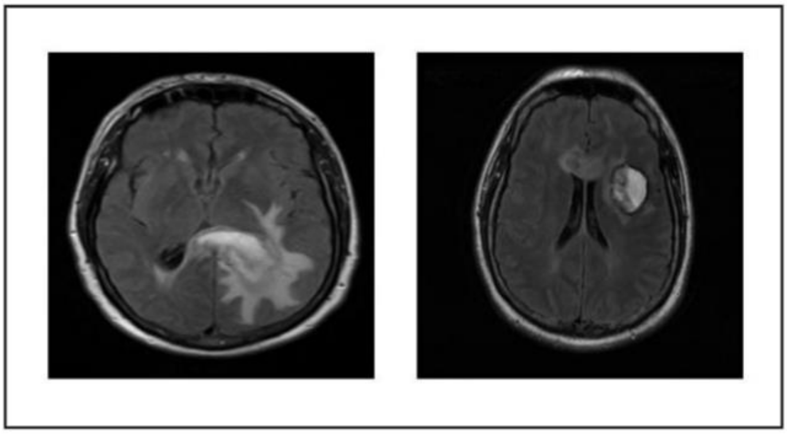


Figure 16. A visualization of a long- and short-term survival slice from the Flair sequence. [27]

After that, the author prepared the data preprocessing by creating a region of interest (ROI) in each brain tumor-containing slice then the authors resized the MRI image to 224x224 and 40x40 for the part around the tumor to fit the input image size of convolution neural network (CNNs) architecture. Figure 17 illustrates instances of bounding boxes, capturing either exact or tightly fitted contours around the tumor, along with the process of cropping tumor images using a standardized box size. Both types of boxes undergo warping to achieve the requisite size exemplifying the method employed for accurately outlining and adjusting the dimensions of the tumor regions of interest.

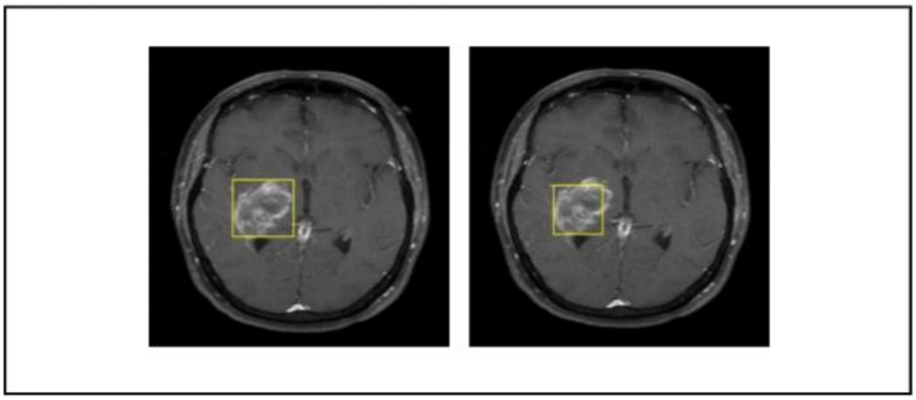


Figure 17. Images visualize of the bounding boxes and resizing pictures of brain tumors [27]

After that, the authors used pre-trained Fast convolution neural network (CNN-F) to extract deep features then predicted short term and long term of survival time based on these features. As depicted in figure 18, CNN-F is composed of 8 trainable layers, with 5 being convolutional layers and the remaining 3 being fully connected layers. The input requirement for the network is the image which has a 224×224 pixels RGB representation. To accommodate grayscale images, we adapted the code through the deactivation of hidden units corresponding to the green and blue channels, focusing solely on data extracted from the red (R) channel, which possesses the lowest frequency. This modification enables the incorporation of grayscale images into the network architecture.

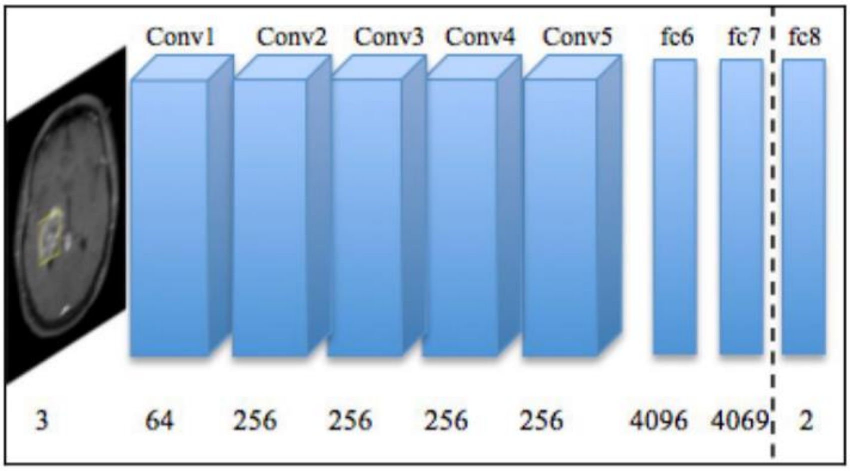


Figure 18. Pre-trained CNN-F architecture. [27]

After feature extraction step, the authors fine tuning the pre-trained model by creating the new fully connected layer has two neurons by keeping the class weights of two categories which have the maximum activation among 1000 initial classes. After Convolutional Neural Network (CNN) model was fine-tuned, the Convolutional Neural Network (CNN) model was optimized for ILSVRC12 with learning rate less than original learning rate

Finally, the authors conducted training of the Weka RandomForest classifier utilizing The extracted features from the second fully connected layer were used for learning. The parameter for the number of trees was configured to 200. For feature selection, we opted for Weka's implementation of symmetrical uncertainty to identify the optimal 7 features. The selection of the feature count was constrained to not exceed one-third the size of the dataset. Leave-one-out cross-validation was employed for testing purposes. The accuracies of prediction pertaining to precise tumor patches are detailed in Table 3, while table 4 presents the outcomes derived from the analysis of trimmed patches.

Table 3. The RandomForest classifier's accuracies determined on "exact" tumor patches. [27]

MRI Sequencing	Deep features before treatment.	Deep features after treatment
Flair	63.64%	63.64%
ADM	31.82%	50%
T2-weighted	18.18%	27.27%
T1-weighted	68.18%	27.27%

Table 4. The RandomForest classifier's accuracies determined on "cropped" tumor patches. [27]

MRI Sequencing	Deep features before treatment.	Deep features after treatment
Flair	63.64%	50%
ADM	27.27%	45.45%
T2-weighted	72.73%	36.36%
T1-weighted	40.91%	50%

2.7. An Innovative Approach for Brain Tumor Identification and Coloring from 2D MRI Images

I.Haq, N.Ullah, T.Mazhar, M.A. Malik and Iqra Bano published this article [28] on june 6, 2022 . The focus of the research revolves around the development of a comprehensive and fully automated system designed for the removal of skulls from brain image. A notable aspect of this system involves the initial step of binarizing grayscale images, a process integral to subsequent stages of segmentation. The segmentation process specifically aims at distinguishing normal brain tissue from areas affected by tumors, thereby enhancing the precision of tumor cell colorization.

Initially, the inaugural phase of this research involves the ingestion of human brain MRI images sourced from a specified dataset. The dataset employed in this study comprises two distinct sets of brain images, each serving as a focal point for investigation and result presentation. The first set is derived from brain MR images obtained from BVH (Bahawal Victoria Hospital), and the second set is sourced from RIDER Neuro MRI. This deliberate selection of diverse brain images from distinct sources enhances the robustness and generalizability of the study's findings, thereby contributing to a more comprehensive exploration of the research objectives. In this study, the researcher incorporated a total of 35 images sourced from the BVH dataset and an additional 19 images from the RIDER Neuro MRI dataset. Among these, 24 images depict brains afflicted with tumors, while 11 images portray brains without tumors. The brain images encompass variations in imaging modalities, including T1, T2, and FLAIR sequences. All images within the dataset share a standardized 512 × 512 pixels in size and were uniformly storaged at bitmap-formatted, ensuring consistency and facilitating cohesive analysis throughout the research endeavor. The MRI images from the BVH dataset are depicted in figure 19 and the MRI images from the RIDER dataset are presented in figure 20.

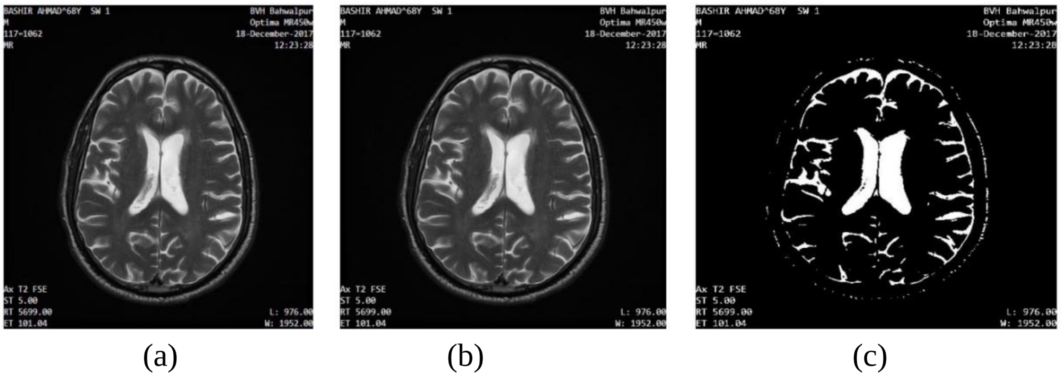


Figure 19. Illustrates the transformation process of images from BVH MRI dataset (a) Original image; (b) image converted to grayscale; (c) binarized image [28]

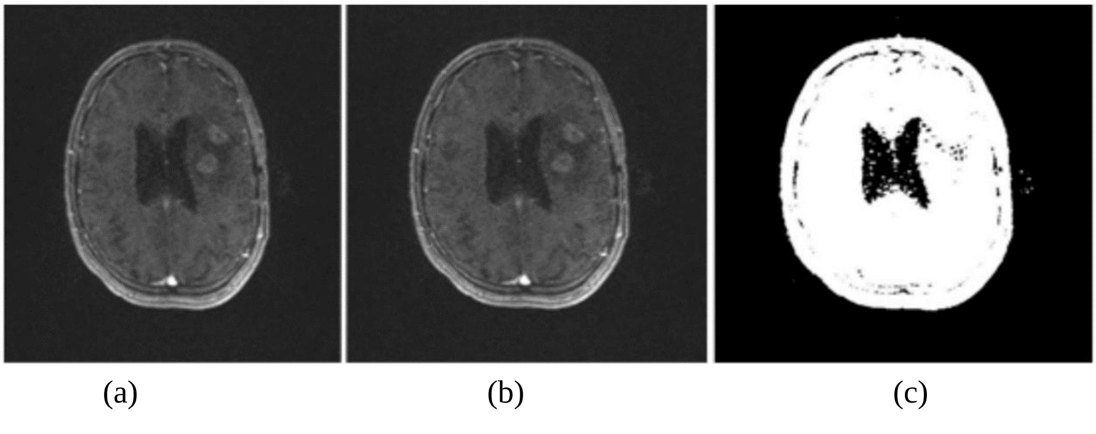


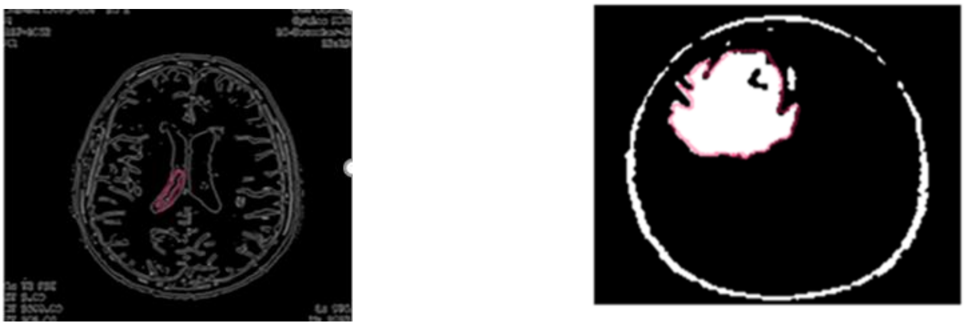
Figure 20. Illustrates the transformation process of an image from Rider MRI dataset (a) Original image; (b) grayscale image; (c) image converted to binary [28]

Upon the completion of the input of MRI images, the subsequent phase involves the conversion to bitmap format. In this critical step, DICOM images are transformed into bitmap images, wherein the DICOM image input is saved as raw pixel data. The conversion process varies for 2D DICOM data and 3D MHD format; RadiAnt DICOM Viewer is utilized for 2D DICOM data, while the MITK tool facilitates the conversion of 3D MHD format into bitmap image format. Subsequently, a weighted average technique is applied to enhance image quality, effectively mitigating noise. Following noise reduction, the RGB images in bitmap format are converted into grayscale images, employing an average weighted method to compute the average RGB value per pixel. Gaussian high-pass filtering is then employed, selectively adjusting pixel intensities to refine image quality. This technique is instrumental in detecting pixels with black values. Following skull removal, a gray toning method is applied to identify tumor regions, and a probability operation is executed to convert the grayscale image into binary form, distinguishing between black (0) and white (1) components, thereby producing a binary image. After that, the authors used the watershed algorithm to segmentate the area in image.

After applied watershed segmentation algorithm, the authors used two techniques to mask the brain tumor area which are edge detection technique and morphological operations technique to select the region of interest (ROI) of the brain image. To detect the region of brain, the authors used a lot of diverse edge finding methods The mentioned techniques encompass various image processing methods, such as the Gaussian Laplacian, Sobel, Prewitt, Roberts, zero crossing, and sharp edge algorithms. Furthermore, morphological operations technique is applied to segregate the region of interest (ROI).

After the region masking stage, the region of interest (ROI) of the brain is used to discriminate the color by assigning distinct colors to the gray matter, white matter, and CBF color then the authors used colorization methods based on indices to apply color to distinct regions within the brain and areas of abnormality. After this step, these images were applied to distinguish the region of brain from different color.

Finally, the research focuses on evaluating the quality of colorized medical images through the application of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). The PSNR score serves as an indicator of visual information and accuracy in colorized pictures, with higher scores denoting better quality. The SSIM measures the structural similarity between input and output images, reflecting the systemic accuracy of the process. A dataset comprising 35 brain images for simulation and assessment, data from the Bahawal Victoria MRI dataset and the Rider MRI dataset are utilized. MATLAB R2018a is employed for image processing, revealing that detecting brain tumors in these images is challenging for ordinary observers. The watershed algorithm is then applied for efficient area segmentation as a precursor to tumor identification. This algorithm utilizes the concept of catchment basins and intensity-based pixel grouping to address errors value of segmentation. To reduce error, the segmentation marker is carefully selected from local gradient minim. The study presents processed and evaluated results, illustrated in Figure 21 providing insights into the effectiveness of the proposed approach in enhancing medical image quality and segmentation accuracy.



(a) (b)

Figure 21. The result of BVH MRI dataset and Rider MRI dataset (from left to right) [28]

Furthermore, the authors calculated the area segmentation accuracy and image of brain tumor detection from 35 images of brain tumor was 100%. Hence, the precision of both area segmentation and the classification of brain tumors in the images was established at 100%. The result was presented in table 5. This outcome stems from the system's ability to successfully carry out classification and colorization without any instances of misidentification during the process.

Table 5 [28] The outcome value from the author’s method.

Brain Tumor dataset			Normal Brain dataset		
Total	Accurately Determined	Accuracy result	Total	Accurately Determined	Accuracy result
24	24	100%	11	11	100%

2.8. Diagnosis and Classification of Brain Tumors using Pre-Trained Convolutional Neural Networks

In this research [29], authored by Dmytro Filatov from Aimech Technologies Corp. in San Francisco, USA and Ghulam Nabi Ahmad Hassan Yar from the Department of Electrical and Computer Engineering at Air University in Islamabad, Pakistan, the focus is on the critical aspects of brain tumor diagnosis and classification. The human brain, being a pivotal organ, faces the peril of tumors, making accurate diagnosis and classification imperative for effective treatment. Leveraging Medical Resonance Imaging (MRI) and Convolutional Neural Networks (CNNs) with transfer learning from pre-trained models, the authors aim to achieve a threefold goal: diagnosing brain tumors, classifying specific tumor types (Glioma, Meningioma, and Pituitary), and optimizing efficiency through the utilization of pre-trained models. The paper not only reviews previous work but also details the methodology, dataset, results, and concludes with prospects for future enhancements in refining the quality of brain tumor diagnosis and classification.

There are three main stages for this research. Firstly, image augmentation is applied to the input MRI images to address the challenge of limited data. Augmentation involves generating image variants through translations, rotations, scaling, shearing, and flipping (horizontal and vertical). This process is crucial for expanding the dataset, and the augmented images are treated as distinct, increasing the overall dataset size. The specific augmentations employed in this research include a 90-degree rotation, horizontal flip, and vertical flip.

After the completion of the image augmentation phase, the investigation employs two CNN classification models, namely ResNet50 and EfficientNet. ResNet50, introduced by Microsoft in 2015, adopts residual blocks with identity mapping to overcome challenges such as vanishing gradient, enabling the construction of deeper networks. In contrast, EfficientNet, developed by Google in 2019, incorporates width scaling, depth scaling, and resolution scaling, resulting in enhanced performance. A novel architecture is introduced for the specific task of classification, incorporating layers such as pooling, flattening, dense, and dropout. This customized architecture is meticulously crafted to address the distinctive characteristics inherent in the brain tumor classification problem. In the implementation step, the models is carried out on Google Colab with specific hardware specifications, including 12.5GB of RAM and a Tesla P100-PCIE-16GB GPU. Input images are maintained at a size of 512x512x3 during training. The learning rate is set to 0.001, with a batch size of 8, and categorical crossentropy is chosen as the loss function. The dataset, comprising Glioma, Meningioma, Pituitary, and Normal classes, is split into training and validation sets at an 80% to 20% ratio. During training, validation loss is monitored, and the model is saved at each occurrence of the minimum validation loss. Early stopping is applied if the validation loss does not decrease for 9 consecutive epochs. The maximum number of epochs is set to 50. The dataset distribution into training and testing images is presented in Table 6.

Table 6: [29] Dataset distribution into training and testing images

Type	Glioma	Meningioma	Pituitary	Normal
Total	1621	1645	1757	2000
Training	1321	1339	1457	1595
Validation	300	306	300	405

Finally, diverse pre-trained models are enlisted for the classification task, and Table III provides an overview of the classification outcomes, encompassing training and validation accuracy as well as loss metrics. Remarkably, the performance of EfficientNet models surpasses that of ResNet models significantly, with EfficientNetB1 emerging as the top-performing model, showcasing the highest levels of both training and validation accuracy. The graphical representation in figure 22 illustrates the training accuracy and loss trends across all models, highlighting the consistent improvement in training accuracy and reduction in training loss for EfficientNet models, while ResNet50 exhibits comparatively less stable behavior. Analogous patterns are discernible in figure 23 concerning validation accuracy and loss, further underscoring the consistent superiority of EfficientNet models over ResNet50. Any aberration, such as the validation loss spike observed in EfficientNetB1 at epoch 6, is addressed, showcasing subsequent improved performance. The comprehensive classification results outlined in Table III affirm the superior accuracy and loss metrics associated with EfficientNet models in comparison to ResNet50. The table 7 show the results of accuracy and loss value from training and validate process for different models

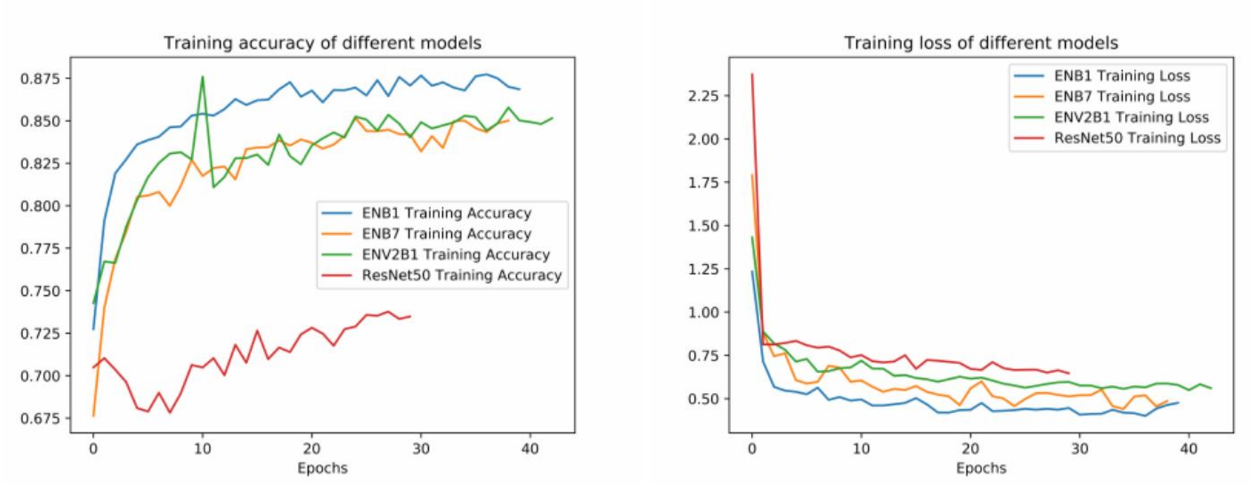


Figure 22. Accuracy during training (left) and loss (right) for various models. [29]

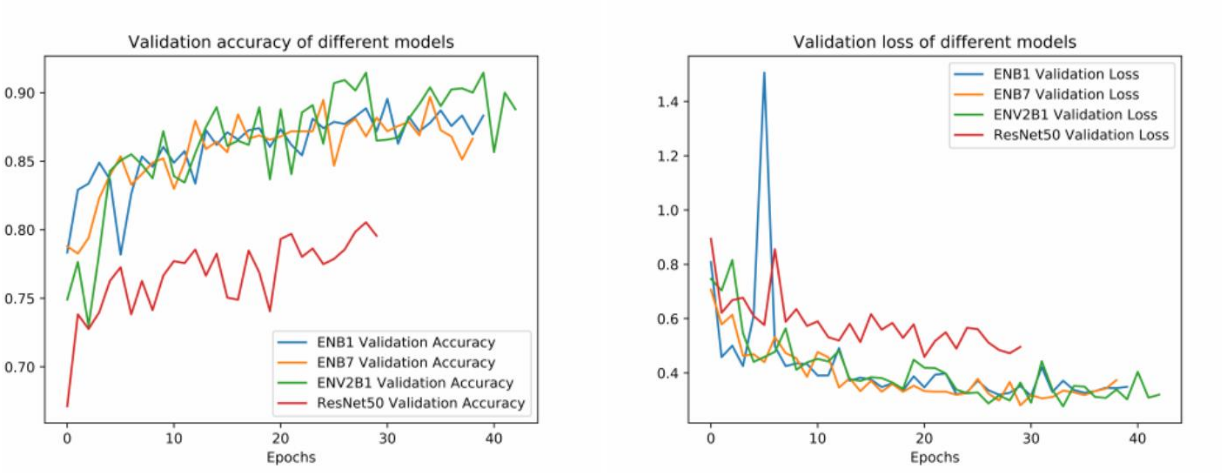


Figure 23. Accuracy during validation (left) and loss (right) across various models. [29]

Table 7. Classification results for different models [29]

Model	Epochs	Training Accuracy	Training Loss	Val Accuracy	Val Loss
EfficientNetB7	39	0.8419	0.5129	0.8818	0.2807
EfficientNetV2B1	43	0.8491	0.5695	0.8917	0.2768
EfficientNetB1	40	0.8767	0.4076	0.8955	0.3152
ResNet50	30	0.7282	0.6719	0.7932	0.4593

CHAPTER 3

METHODOLOGY

3.1. Proposed method

This chapter approach to classifying the tumor using fundamental image processing techniques and detecting the tumor using transfer learning by combining two pre trained of convolution neural networks (CNN) models.

All proposed method of this research to build a pre-trained model includes the following steps:

- First, I downloaded 7023 MRI brain images from Kaggle which is a public dataset. This dataset includes MRI images of normal brain and brain has tumor (glioma tumor, meningioma tumor, pituitary tumor).
- Secondly, I moved to preprocessing step which are resized and rescaling the dataset to fit with pre-trained model.
- Thirdly, I split the dataset into two parts which are train set and validate set. The train dataset used for training pre-trained deep neural network model. The validate dataset used to evaluate efficient of the model after it trained in every epoch. On the other hand, I downloaded 3160 MRI brain images from Kaggle which is a public dataset. This dataset also includes MRI images of normal brain and brain has tumor (glioma tumor, meningioma tumor, pituitary tumor) and used this dataset to evaluate efficient of the model after it trained.
- Finally, I built the model by combining two pre-trained deep neural network models (InceptionV3, MobileNetV3) by using techniques is called "Model Ensemble" or "Model Fusion" then evaluate the accuracy when the model classified MRI brain images into four class which are normal brain, glioma tumor, meningioma tumor and pituitary tumor The new transfer learning model also have some last replaced layers to classify.

The figure 24 presented my proposed method for brain tumor classification which is combining two convolutional neural network (CNN) pre-trained models (InceptionV3, MobileNetV3)

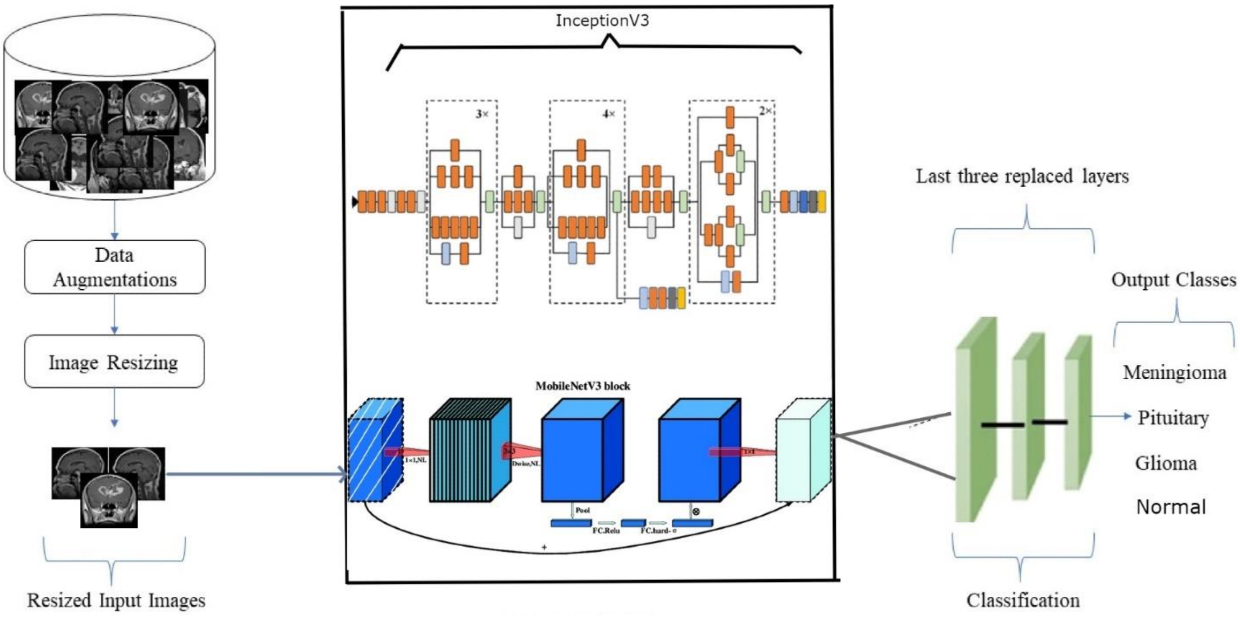


Figure 24. Proposed method for brain tumor classification with combine CNN pre-trained model (InceptionV3, MobileNetV3)

3.2. Dataset

3.2.1. Training dataset

First of all, I chosen the dataset from Kaggle [30] which is a subsidiary of Google LLC. This subsidiary provides a lot of public datasets and allow user to use these public datasets to build model in machine learning or data science environment. This dataset has 7023 MRI brain images. These MRI images include 2000 MRI images of normal brain, 1621 MRI brain images depicting glioma tumors, 1645 MRI images of brain has meningioma tumor and 1757 MRI images of brain has pituitary tumor. Figure below show the sample of brain MRI image from Kaggle for training process.

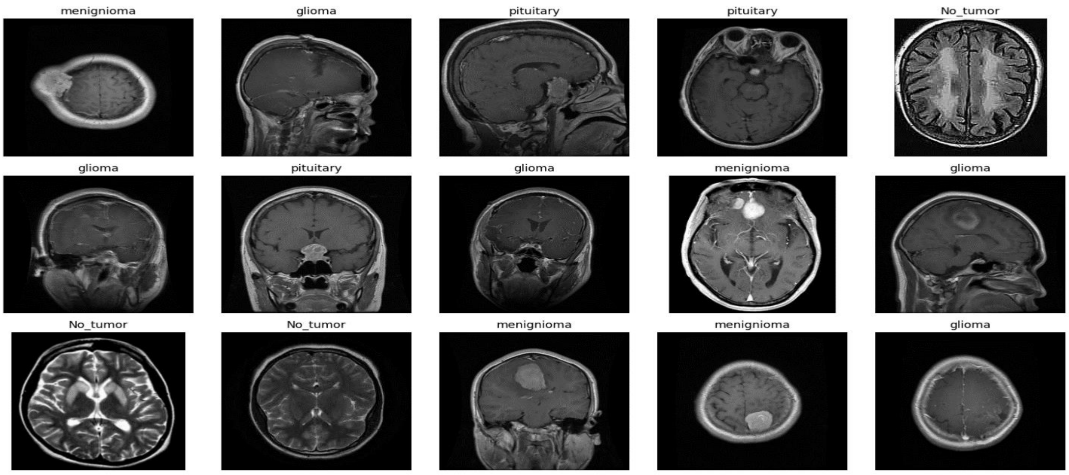


Figure 25. Sample of brain MRI image for training process. [30]

3.2.2. Evaluation dataset

In this step, I chosen the dataset from Kaggle [31] which is a subsidiary of Google LLC. This subsidiary provides a lot of public datasets and allow user to use these public datasets to build model in machine learning or data science environment. This dataset has 3160 MRI brain images. These MRI images include 396 MRI images of normal brain, 937 MRI brain images depicting glioma tumors, 926 MRI images of brain has meningioma tumor and 926 MRI images of brain has pituitary tumor and I used this dataset to evaluate efficient of the model after it trained.

Figure below show the sample of brain MRI image from Kaggle for evaluating process.

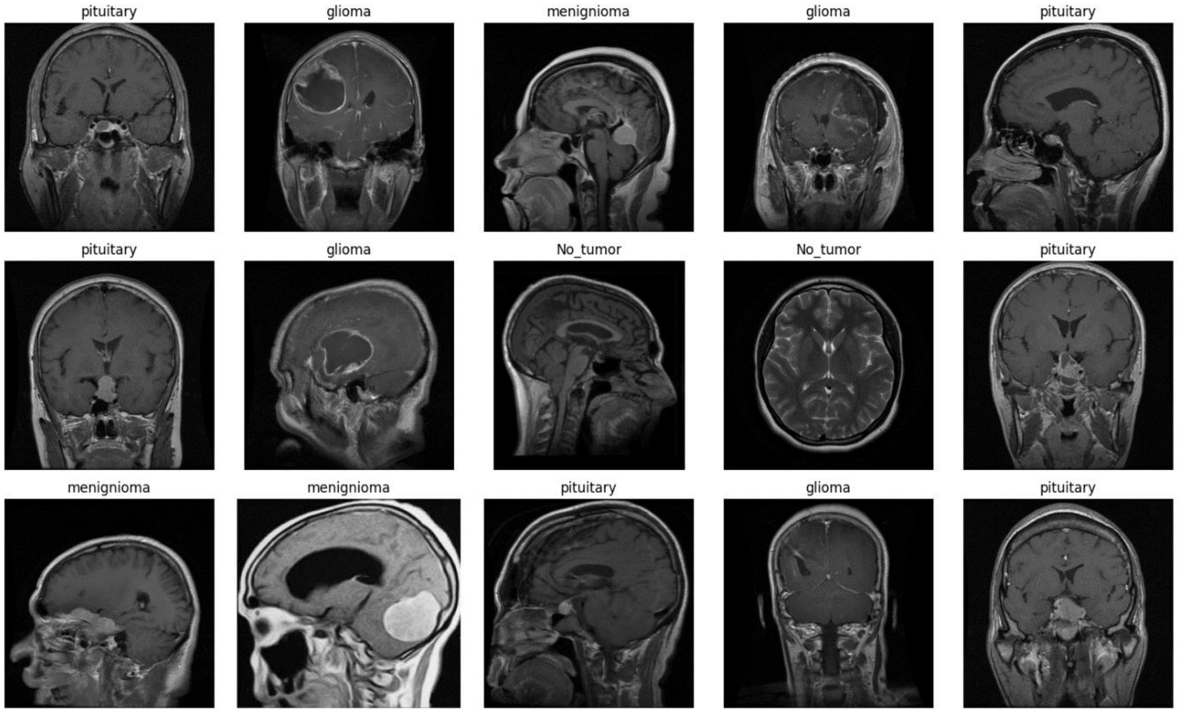


Figure 26. Sample of brain MRI images for evaluating process. [31]

3.3. Preprocessing

In this pivotal stage of the research process, my objective is to meticulously engage in the transformation of raw data. This undertaking is indispensable as it aims to elevate the preparedness and optimally facilitate the subsequent processing of the data when embarking on the construction of machine learning models or deep neural network (DNN) models. Employing a meticulous approach, I employ various preprocessing techniques to refine and enhance the raw data, ensuring that it is primed for the intricate demands of the model-building phase.

3.3.1. Rescaling data

Our origin MRI brain images are RGB image which can be viewed as three different images (a red scale image, a green scale image and a blue scale image) and the pixel value from these images have the range from 0 to 255 (0 is black and 255 is white) and these values are too high for the model to process. In this step, I rescaled the image by changing the pixel value of the image which have pixel value in range from 0 to 255 of pixel value to the target pixel value in range from 0 to 1. This method also called rescale 1./255. To rescale these pixel values, I used ImageDataGenerator function from Keras to rescale 1./255.

Since some images have high pixel value range or low pixel value range and these images all process the same model so if we not scaling the image have high range of pixel value, these images will have a lot of advantages to determine how to update weights. Therefore, rescaling images will make all images in the same manner.

3.3.2. Image resizing

When we build the Deep learning (DN) model, this model was trained faster on small image which have a small size a pixel. Since a big input image will require a lot of neural network to learn therefore this will increase the time of training model. So resizing the image was very necessary in training deep learning model.

In this dataset, I had a lot of images of MRI brain and the size of these images are different so resizing these images also make all the image have a same size and removed superfluous detail of the MRI brain image.

In this paper, I tried to built the model by combining two pre-trained Convolutional Neural Network (CNN) models (InceptionV3, MobileNetV3). These Convolutional Neural Network (CNN) models required a size of 229 × 299 MRI brain image to learn effectively.

3.3.3. Data splitting

I divided the dataset which is 7023 MRI brain images into two parts: train set and validation set.

- The train set contains all MRI brain images to training the convolutional neural network (CNN) model so this dataset required a lots image than test set and validation test. Therefore, I split this dataset from the main dataset as 80% which are approximately 5618 MRI brain images.
- The validation set contains data to validate the model to show how efficient of the model after it was trained in every epoch. An epoch means one complete pass of the training dataset when building the Convolutional Neural Network (CNN) model. I split this dataset from the main dataset as 20% which are approximately 1405 MRI brain images.
- Finally, I downloaded 3160 MRI brain images from Kaggle which is a public dataset to make it become the test set. This dataset also includes MRI images of normal brain and brain has tumor (glioma tumor, meningioma tumor, pituitary tumor). The test set contains data to validate the model to show how efficient of the model after it was trained.

3.4. Building combined pre-trained convolutional neural network models

In this phase, I attempted to construct a model by integrating two pre-trained Convolutional Neural Network (CNN) models (InceptionV3 and MobileNetV3) through techniques known as "Model Ensemble" or "Model Fusion." During this stage, the input data after undergoing image resizing and rescaling was processed through both pre-trained models to extract features. Subsequently, I utilized the concatenate function from the NumPy library to merge the features extracted from the InceptionV3 and MobileNetV3 pre-trained models and resulting in a combined feature vector.

To enable the combined model to classify four distinct objects which are normal brain tissue and brain tissue affected by tumors (glioma, meningioma, and pituitary tumors), I fine-tuned the combined (InceptionV3, MobileNetV3) model by incorporating additional fully connected layers, including a flatten layer and dense layers. Furthermore, I incorporated Batch Normalization and Dropout layers to enhance the effectiveness of the transfer learning model's classification capabilities.

At the culmination of the model's architecture, I employed the softmax activation function to facilitate the classification of the four aforementioned objects. Additionally, I utilized the SGD optimizer algorithm to optimize the model's training process, resulting in increased efficiency and reduced training time. The maximum number of epochs was set to 30. The pseudocode below show the process of creating a combined deep learning model using two pre-trained models, specifically InceptionV3 and MobilnetV3.

```
Set the input shape to (299, 299, 3).
Create a single input layer for both base models InceptionV3 and MobileNetV3, using the defined input shape.
Create the base models using InceptionV3 and MobileNetV3, initialized with weights from 'imagenet', excluding the top layer, and using the input layer.
Freeze the layers of the base models by setting the 'trainable' attribute of each layer to False.
Process the output of InceptionV3_base_model:
    Obtain the output of InceptionV3_ base_model.
    Flatten the output of InceptionV3_ base_model.
    Add a Dense layer with 128 units and 'relu' activation to the output of InceptionV3_ base_model after Flatten.
Process the output of MobileNetV3_ base_model:
    Obtain the output of MobileNetV3_ base_model.
    Flatten the output of MobileNetV3_ base_model.
    Add a Dense layer with 128 units and 'relu' activation to the output of MobileNetV3_ base_model after Flatten.
Combine the outputs of the two base models using concatenation.
Normalize the combined outputs using BatchNormalization.
Apply dropout to the normalized outputs with a rate of 0.5.
Create the final output layer with a Dense layer of 4 units and 'softmax' activation.
Create the combined model with the input layer and the final output layer.
Compile the model with 'categorical_crossentropy' loss, SGD optimizer with a learning rate of 0.001, and 'categorical_accuracy' as the metric.
```

CHAPTER 4

IMPLEMENT AND RESULTS

4. Experiments

In this paper, I used Keras, an open-source library for Neural Network to implement two convolutional neural network (CNN) models by using transfer learning with two pretrained convolutional neural network (InceptionV3, MobileNetV3) and these training were performed on Google Colab which allow to access free GPU. Each model had maximum 30 epochs and processed approximately seven minutes for each epoch. All of proposed method for each model concluded download the brain MRI image, resize, rescale the dataset, classify normal brain MRI image and tumor brain MRI image. This experiment has been performed on Kaggle dataset which is a subsidiary provides a lot of public datasets to build model in machine learning.

4.1. User requirement analysis

In this study, I build the transfer learning CNN model by combining two pre-trained models (InceptionV3, MobileNetV3) from Keras, an open source library for Neural Network which was developed in 2005 by Francois Chollet. This library provided a Python interface for artificial neural networks. I conducted model training on my laptop, equipped with a 12th gen Intel Core i7-12700H processor, without the need for external resources. Utilizing the laptop's powerful processor, which features 14 cores, along with 16GB of RAM. This is allowed me to efficiently train the model. This approach proved effective, minimizing the training time compared to other setups that might rely on external resources like free GPUs on platforms such as Google Colab. I also used Pandas library to analyze and manipulate the data.

4.2. Brain MRI images from Kaggle

4.2.1. Brain MRI images for training and validate process

I evaluated and compared the accuracy of combine pretrained models from brain MRI images from Kaggle [30] which is a subsidiary of Google LLC. This subsidiary provides a lot of public datasets and allow user to use these public datasets to build model in machine learning or data science environment. This dataset has 7023 MRI brain images and has four categories which are normal brain and brain has tumor (glioma tumor, meningioma tumor, pituitary tumor). These MRI images include 2000 MRI images of normal brain, 1621 MRI brain images depicting glioma tumors, 1645 MRI images of brain has meningioma tumor and 1757 MRI images of brain has pituitary tumor. Some sample of brain MRI image are showed in these figures below. Table 8 showed the number of images in two categories from brain MRI images

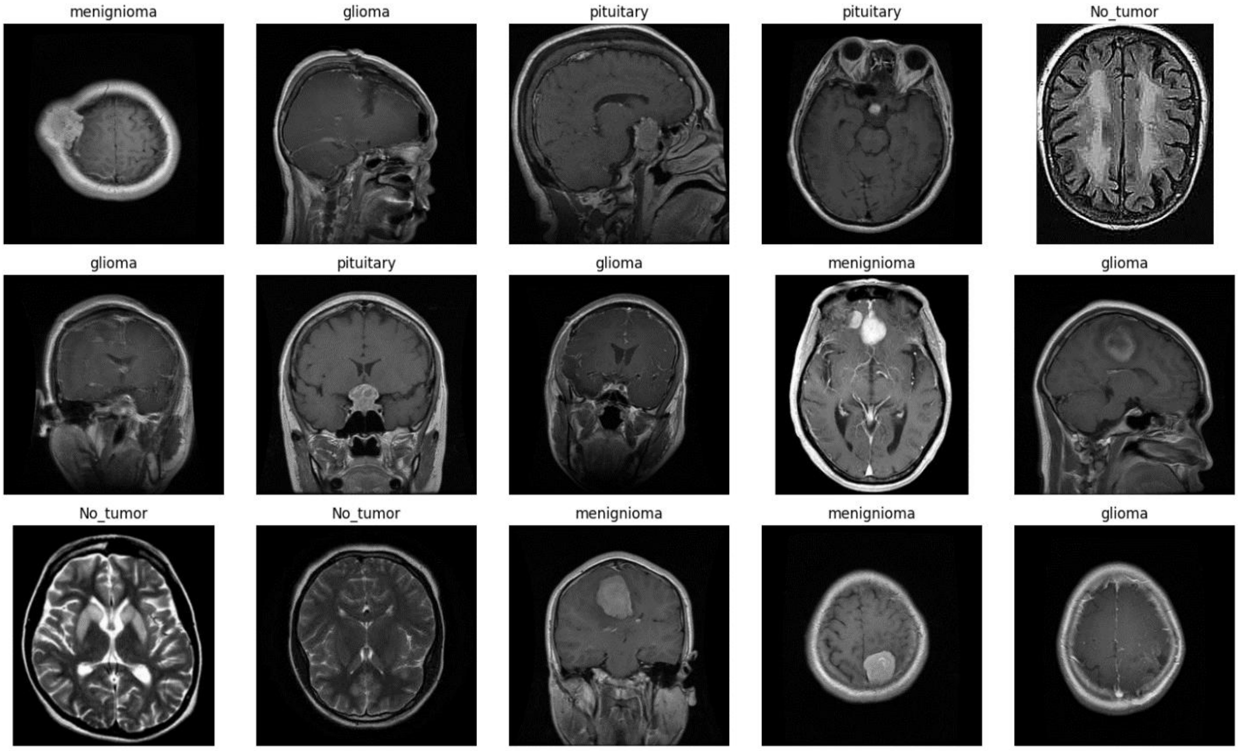


Figure 27. Sample of brain MRI image dataset [30]

Table 8. Number of images in four categories from brain MRI images for training process. [30]

	Normal brain	Glioma tumor	Meningioma tumor	Pituitary tumor
Number of images	2000	1621	1645	1757

Image below showed the distribution of brain MRI images across classes for training and validating process.

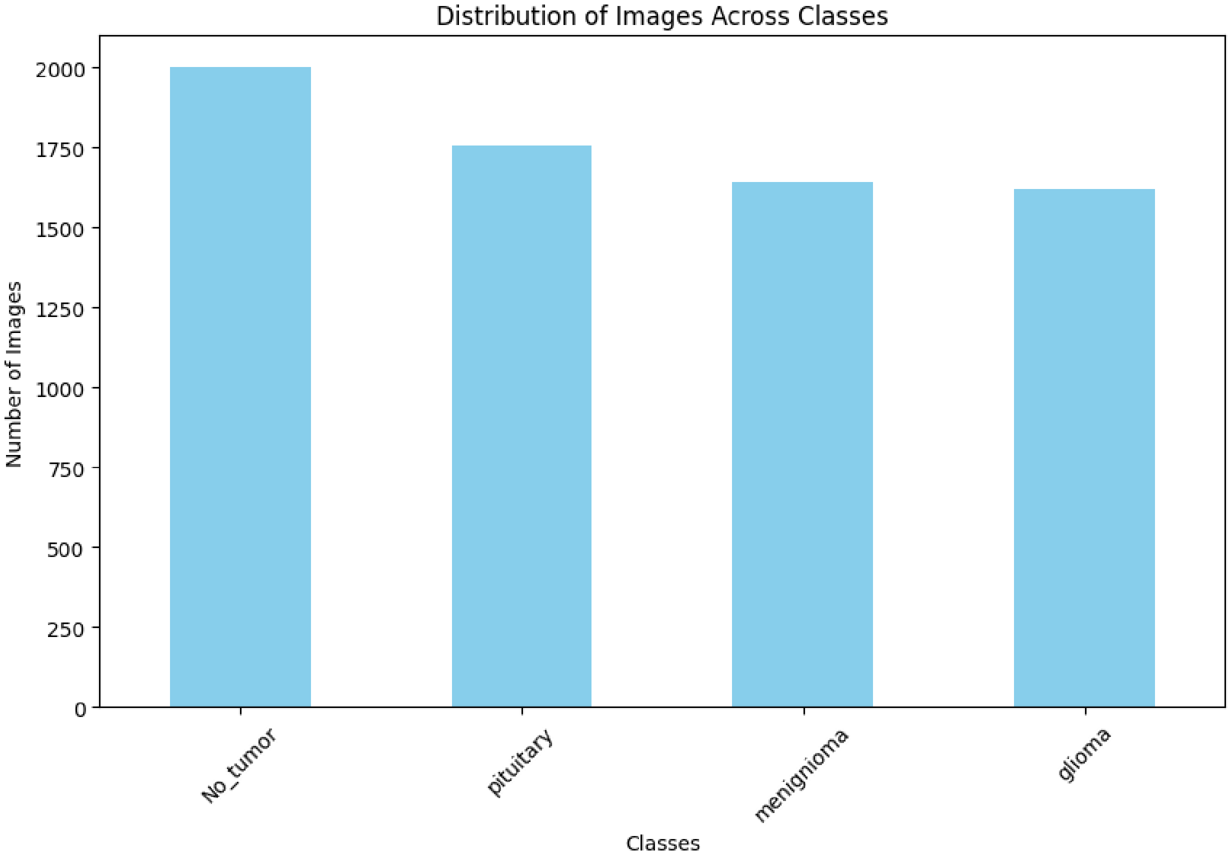


Figure 28. the distribution of brain MRI images from Kaggle dataset for training process [30]

4.2.1.1. Brain MRI images for testing process

I evaluated and compared the accuracy of combine pretrained models from brain MRI images from Kaggle [31] which is a subsidiary of Google LLC. This subsidiary provides a lot of public datasets and allow user to use these public datasets to build model in machine learning or data science environment. This dataset has 3160 MRI brain images and has four categories which are normal brain and brain has tumor (glioma tumor, meningioma tumor, pituitary tumor). These MRI images include 396 MRI images of normal brain, 937 MRI brain images depicting glioma tumors, 926 MRI images of brain has meningioma tumor and 926 MRI images of brain has pituitary tumor and I used this dataset to evaluate efficient of the model after it trained. Some sample of brain MRI image are showed in these figures below. Table 9 showed the number of images in four categories from brain MRI images.

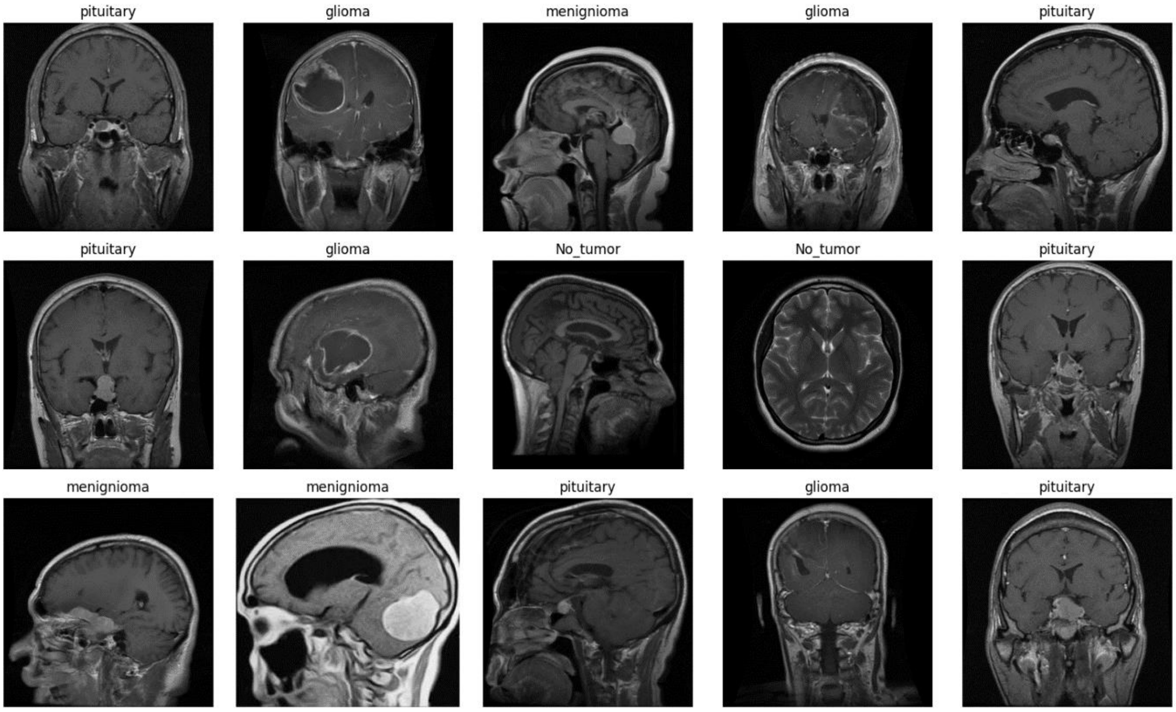


Figure 29. Sample of brain MRI image for testing process [31]

Table 9. Number of images in four categories from brain MRI images for evaluation process. [31]

	Normal brain	Glioma tumor	Meningioma tumor	Pituitary tumor
Number of images	396	937	926	901

Image below showed the distribution of brain MRI images across classes for testing process.

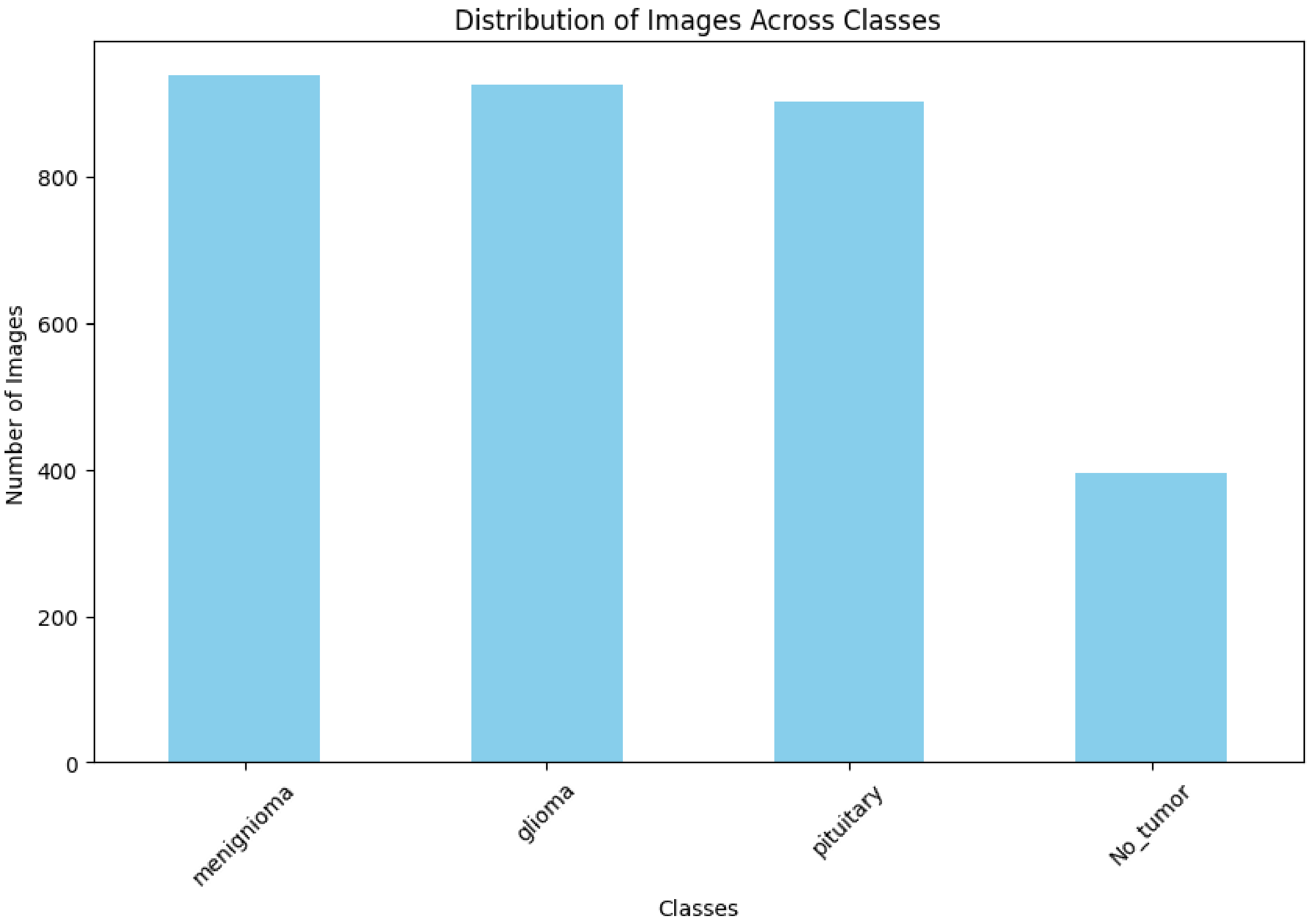


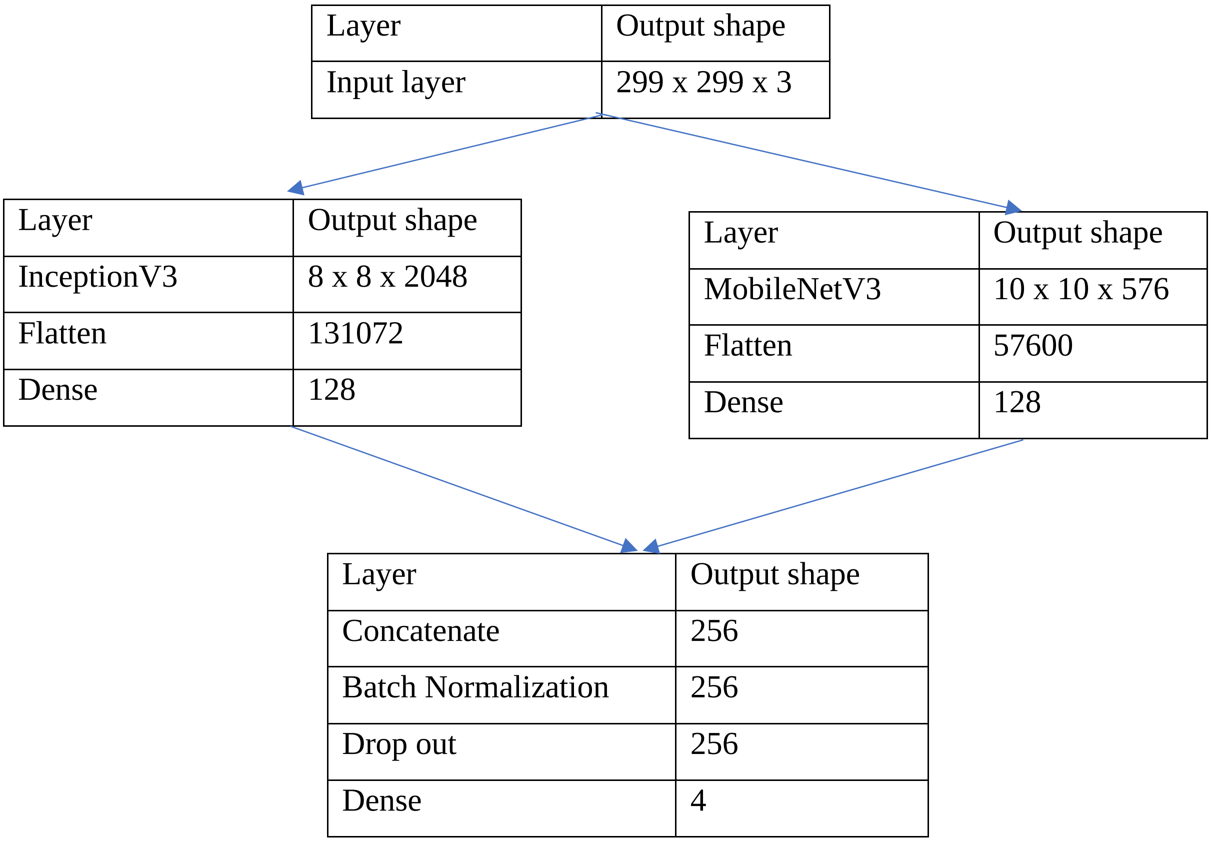
Figure 30. the distribution of brain MRI images from Kaggle dataset for evaluation process [31]

4.3. Experiments with Combine model (InceptionV3, MobileNetV3) pre-trained model

In this section, the focus shifts towards exploring a hybrid approach by combining the strengths of two pre-trained models, namely InceptionV3 and MobileNetV3, to construct a convolutional neural network (CNN) model using transfer learning. The experimental process involves initially leveraging the feature extraction capabilities of InceptionV3 as the base model for the CNN architecture. Subsequently, the model undergoes fine-tuning through the incorporation of additional fully-connected layers to enhance its performance.

The architecture of the resulting combined model is presented in the table below, showcasing the integration of InceptionV3 and MobileNetV3 for a more comprehensive and diverse feature representation.

Table 10. Architecture of combine CNN model by transfer learning with InceptionV3 and MobileNetV3 pretrained model.



Layer	Output shape
Input layer	299 x 299 x 3

Layer	Output shape
InceptionV3	8 x 8 x 2048
Flatten	131072
Dense	128

Layer	Output shape
MobileNetV3	10 x 10 x 576
Flatten	57600
Dense	128

Layer	Output shape
Concatenate	256
Batch Normalization	256
Drop out	256
Dense	4

This experimental endeavor aims to harness the unique characteristics of both InceptionV3 and MobileNetV3, taking advantage of their respective strengths in feature extraction and computational efficiency. The synergy of these pre-trained models is anticipated to yield a more robust and versatile CNN architecture, capable of achieving enhanced performance across various tasks.

The detailed analysis of the experimental outcomes, including performance metrics, fine-tuning strategies, and any observed synergies between the combined models, will be discussed in subsequent sections. Through these experiments, we seek to gain insights into the potential benefits and challenges associated with the fusion of diverse pre-trained models in the real of convolutional neural networks.

The architecture of network in table 10 included InceptionV3 pre-trained model and MobileNetV3 pre-trained model with two fully connected layers. Firstly, I applied the InceptionV3 pre-trained model followed by a Flatten layer to convert the data into a 1-dimensional array and a Dense layer. Secondly, I applied MobileNetV3 pre-trained model followed by a Flatten layer to convert the data into a 1-dimensional array and a Dense layer. After that I merge two output from two pre trained CNN models by using concatenate function from numpy library to combine features extracted from InceptionV3 pre-trained model and MobileNetV3 pre-trained model. This technique is called "Model Ensemble" or "Model Fusion" then applied by a batch normalization to avoid overfitting while training the CNN model and a Dense layer followed by a softmax activation function. The last fully connected layer had to classify the brain have tumor (glioma tumor, meningioma tumor, pituitary tumor) or do not have tumor from brain MRI images.

I started training the proposed neural network architecture of combined CNN models (InceptionV3, MobileNetV3) with 30 epochs, I calculated the loss with categorical cross entropy function and implemented the SGD optimizer algorithm. Overall training process is presented by these figures below.

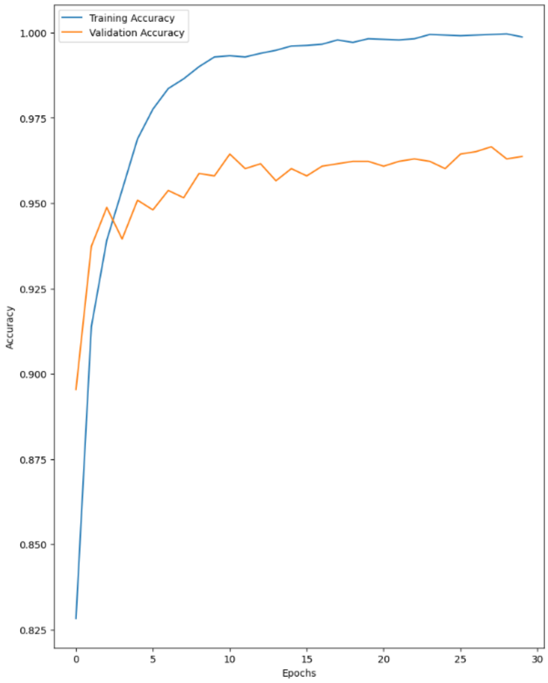


Figure 31. The validation accuracy and training accuracy of the combine CNN pre-trained model

According to the validation accuracy and training accuracy from the figure 31, the number of epoch is on the x-axis and the point of accuracy is on y-axis. The training accuracy is represented by the blue line, while the orange line represents the validation accuracy. From epoch 1 to epoch 3, the validation accuracy had about 0.94 of accuracy value and continue increased the value to the end of the training and had 0.9637 of accuracy value. At epoch 30, the training accuracy is 0.9988, the validation accuracy is 0.9637. I also evaluated the accuracy from the test dataset and I had a result which is 0.9380.

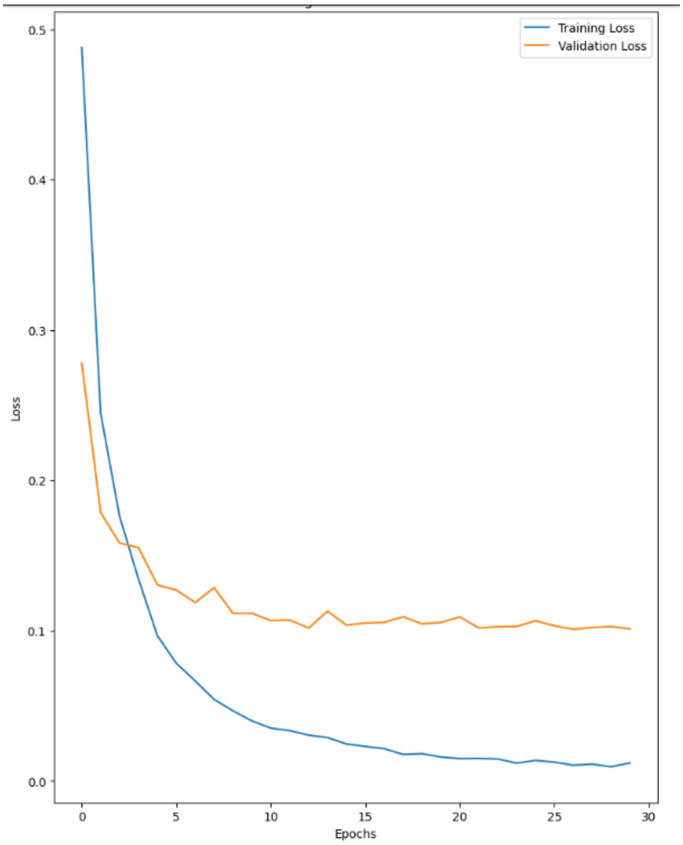


Figure 32. The validation loss and training loss of the combine CNN pre-trained model

According to the validation loss and training loss from the figure 32, the number of epochs is on the x-axis and the point of loss is on y-axis. The blue line represents the training loss, while the orange line represents the validation loss. From epoch 1 to epoch 3, the model had a clear sign of overfitting but after epoch 4, the value of validation loss had been decreased to the end of training. At epoch 7, the validation loss and training loss started to have a good fit which had a small point of loss until the end of the training. At epoch 30, the training loss is 0.0120, the validation loss is 0.1012. I also evaluated the loss from the test dataset and I had a result which is 0.2690. The table below show the accuracy value and loss value from training process and testing process of convolutional neural network (CNN) model using transfer learning by combining two pre-trained models (InceptionV3, MobileNetV3).

Table 11. Accuracy value and loss value from training process and testing process of CNN model using transfer learning by combining (InceptionV3, MobileNetV3) pre-trained models

	Training dataset	Validation dataset	Test dataset
Accuracy value	0.9988	0.9637	0.9380
Loss value	0.0120	0.1012	0.2690

The table shows the accuracy and loss values obtained from training a convolutional neural network (CNN) model using transfer learning by combining the InceptionV3 and MobileNetV3 pre-trained models. Transfer learning allows knowledge gained from pre-training on a large dataset like ImageNet to be transferred to the downstream task of classifying a new dataset.

When evaluating on the training dataset, the combined model achieved an extremely high accuracy of 99.88% and a very low loss value of 0.0120, indicating it was able to fit the training data very well without overfitting. On the validation dataset, the accuracy dropped slightly to 96.37% but the loss increased to 0.1012, suggesting the model was still generalizing well to data it had not been trained on. When finally evaluating on the held-out test dataset, the accuracy was 93.80% and loss 0.2690. While these numbers are lower, they still demonstrate the model performed well on completely unseen data.

The key point is that by combining two pre-trained models (InceptionV3, MobileNetV3) was able to leverage their separate strengths, resulting in even better performance compared to using either model alone. By drawing from two different architectures pre-trained on the same large dataset, the combined model achieved good generalization ability without overfitting, as evidenced by its high accuracy and low loss values on the training, validation, and test sets. In summary, combining multiple pre-trained models enhanced the transfer learning model's performance.

4.4. Result

After I implemented and built combine convolutional neural network (CNN) models by combining two pre-trained models, this session would show the performance of the combine pre-trained models which used to classify the brain MRI image from the public dataset brain MRI images from Kaggle into four categories: normal brain, glioma tumor, meningioma tumor, pituitary tumor. Based on the value in table 12, the convolutional neural network (CNN) model with combining pre-trained MobileNetV3 model and InceptionV3 had the value of training accuracy was 0.9988, training loss was 0.0120, validation accuracy was 0.9637, validation loss was 0.1012 and testing accuracy was 0.9380 and testing loss was 0.2690. In this study, I also implemented and built two combine convolutional neural network (CNN) models and use two models to classify the brain MRI image into four categories: normal brain, glioma tumor, meningioma tumor, pituitary tumor with the same dataset from combine model to show and compare the performance of three models (InceptionV3, MobileNetV3, Combine model). The table below show the accuracy value and loss value from training process and testing process of three convolutional neural network (CNN) models.

Table 12. Accuracy value and loss value from training process and testing process of three CNN models using transfer learning.

Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Testing Accuracy	Testing Loss
InceptionV3	0.9995	0.0068	0.9609	0.1254	0.9297	0.3077
MobileNetV3	0.9106	0.2525	0.8762	0.3573	0.7870	0.7845
Combine model (InceptionV3, MobileNetV3)	0.9988	0.0120	0.9637	0.1012	0.9380	0.2690

Table 12 presents the results of training convolutional neural network models using transfer learning with three different pre-trained models on a given classification task: the InceptionV3, MobileNetV3, and a combined model using both InceptionV3 and MobileNetV3. The accuracy and loss are reported on the training, validation, and test sets to evaluate each model's ability to fit the training data as well as generalize to new examples.

The InceptionV3 model achieved exceptionally high training accuracy of 99.95% and low training loss of 0.0068, demonstrating it had sufficient capacity and learned the patterns in the training data very well. However, its performance dropped more noticeably on the validation and test sets, with the accuracy decreasing to 96.09% and 92.97% respectively. This suggests that while InceptionV3 was able to memorize the training examples, it did not generalize as well and may have started to overfit the training data.

In contrast, the MobileNetV3 model achieved exceptionally high training accuracy of 91.06% and low loss value which is 0.2525 from training process. Besides, its validation and test accuracy were at 87.62%-78.70%%. MobileNetV3 on its own did transfer normal to this new task.

Comparing these models to the combined model, it achieved marginally training performance alone with 99.88% accuracy and very low 0.0120 loss. Most importantly, its validation and test set accuracy and loss were consistently better than both individual models. The combined model achieved 96.37%/0.1012 validation accuracy/loss and 93.80%/0.2690 test accuracy/loss. The accuracy value from validate process of the combined model outperforms that of the InceptionV3 model by 0.0028 and surpasses that of the MobileNetV3 model by 0.0875. Similarly, the testing accuracy of the combined model exceeds that of the InceptionV3 model by 0.0083 and surpasses that of the MobileNetV3 model by 0.151. This demonstrates that by leveraging features learned from both the InceptionV3 and MobileNetV3 architectures pre-trained on ImageNet, the combined model was able to generalize more robustly without overfitting.

One of the key reasons the combined model was able to achieve superior performance compared to the individual models is the architectural diversity it leveraged from InceptionV3 and MobileNetV3. These two pretrained networks have distinct architectural designs - InceptionV3 utilizes multi-scale convolutions within modular blocks, while MobileNetV3 employs depth wise separable convolutions to improve computational efficiency. By combining the pretrained weights from both models, the combined model gained access to complementary low-level and mid-level features learned by each architecture through its distinct design. This allowed it to form a more robust unified representation of the data domain. In addition, through architectural diversity, the combined model enjoyed an increased model capacity and representational power compared to either of the individual models alone. By incorporating parameters from two large pre-trained networks pre-trained on ImageNet, the combined model had more total parameters available to fit the complex patterns in the training data for the given classification task. This increased capacity and diverse set of learned features subsequently enabled the combined model to better fit the training data distribution and generalize to new examples during validation and testing. In essence, both the architectural diversity from InceptionV3 and MobileNetV3, as well as the enlarged model size of their ensemble, contributed to the combined model outperforming the standalone models in its ability to learn from and apply its knowledge to new data.

CHAPTER 5

DISCUSSION AND EVALUATION

5.1. Discussion

5.1.1. Integration of InceptionV3 and MobileNetV3 Pretrained Models

The integration of InceptionV3 and MobileNetV3 pre-trained models in the brain tumor classification framework represents a strategic decision to harness the advantages of both architectures. InceptionV3 is renowned for its ability to capture intricate patterns and details, while MobileNetV3 offers computational efficiency. This combination is especially relevant in the context of medical image analysis, where accurate classification necessitates a nuanced understanding of the diverse structures of brain tumors.

InceptionV3, with its inception modules and multi-scale processing, excels in extracting complex features from medical images. Its ability to capture subtle patterns and variations makes it well-suited for classifying brain tumors which often exhibit intricate structures and heterogeneous characteristics.

On the other hand, MobileNetV3's lightweight architecture and efficient use of depth-wise separable convolutions enable faster processing and reduced computational complexity. This is particularly advantageous in medical image analysis, where dealing with large datasets and real-time processing requirements is crucial.

By combining the strengths of InceptionV3 and MobileNetV3 the proposed framework strikes a balance between accuracy and efficiency. The InceptionV3 backbone provides a robust feature extraction capability, while the MobileNetV3 layers facilitate efficient processing and reduce computational overhead. This combination enables the framework to achieve accurate brain tumor classification while maintaining computational feasibility, making it suitable for practical applications in medical imaging.

5.1.2. Dataset Characteristics and Challenges

The incorporation of the Kaggle dataset, consisting of 7023 MRI brain images, enriches the study by providing a comprehensive dataset that encompasses normal brains and three distinct tumor types: glioma, meningioma, and pituitary. The diversity of the dataset captures a wide range of brain abnormalities, enabling the model to develop a robust generalization capability across various tumor types. However, it is important to acknowledge the challenges associated with this dataset. Class imbalance, where the distribution of samples across different classes is uneven, can pose a challenge for classification models. Additionally, variations in image quality, such as noise, artifacts, and differences in image acquisition protocols, can introduce additional complexities. To ensure the model's effectiveness, it is crucial to carefully address these challenges. Techniques such as oversampling, under-sampling or assigning different weights to different classes can be employed to mitigate the impact of class imbalance. Additionally, image pre-processing techniques can be applied to enhance image quality and reduce noise, thereby improving the model's ability to extract meaningful features.

5.1.3. Transfer Learning and Generalization

Transfer learning plays a crucial role in the proposed brain tumor classification framework, particularly given the limited size of the medical dataset. By leveraging pre-trained models on ImageNet, a vast and diverse dataset, the model gains valuable knowledge from a wide range of images. This knowledge transfer enables the network to learn relevant features and enhances its generalization capabilities to unseen medical data.

The pre-trained weights obtained from ImageNet supply a strong foundation for the model's training process. They serve as a starting point, allowing the model to converge more quickly and efficiently. This is especially beneficial when working with limited medical datasets, as it reduces the risk of overfitting and improves the model's ability to capture the intricate patterns associated with brain tumors.

Furthermore, transfer learning facilitates the adaptation of the pre-trained model to the specific task of brain tumor classification. The model can fine-tune its parameters based on the medical dataset, gradually specializing in distinguishing between normal and tumor tissues. This fine-tuning process leverages the pre-trained knowledge while also incorporating the unique characteristics of brain tumor images, resulting in improved classification performance.

5.1.4. Comparison of Watershed and Threshold-Based segmentation for detecting a brain tumor in MRI images

In the realm of brain tumor detection, two distinct approaches have been explored, each leveraging different methodologies and technologies. The first approach centers around traditional image processing techniques, specifically thresholding and watershed segmentation. This method, applied to MRI images, proves adept at addressing the intricate structures of cerebrospinal fluid (CSF), white matter (WM), gray matter (GM) of the brain. Notably, the inclusion of noise removal through low pass filters during the pre-processing phase enhances segmentation effectiveness. Comparative analysis against alternative algorithms such as SVM and GLCM showcases superior performance, with the proposed technique demonstrating a commendable accuracy of 88.8%, surpassing both SVM and GLCM. The reliance on pixel intensities in threshold-based segmentation, coupled with the strategic application of morphological operations, contributes to its efficacy.

In contrast, the second approach embraces the power of deep learning, employing a transfer learning model that combines the strengths of InceptionV3 and MobileNetV3. This method begins with the acquisition of a substantial dataset comprising 7023 MRI brain images, encompassing both normal brains and those afflicted with glioma, meningioma, and pituitary tumors. The preprocessing step involves resizing and rescaling to align with the pre-trained model. Through a meticulous division of the dataset into training and validation sets, the model undergoes training and subsequent evaluation. The transfer learning model achieves a remarkable classification accuracy of 0.9637, with a minimal loss of 0.1012. This outcome is attributed to the adept utilization of advanced artificial neural networks. and the meticulous structuring of the model, including the replacement of last layers for precise classification. In a comparative assessment, the two approaches reveal distinct strengths. The traditional image processing approach excels in handling the complexities of brain tissue structures and noise removal, while the deep learning model capitalizes on the capabilities of pre-trained neural networks, yielding a higher accuracy rate. The choice between these approaches hinges on specific requirements, computational considerations, and the characteristics of the dataset at hand. The transfer learning model, with its emphasis on leveraging pre-trained neural networks, emerges as a potent contender, showcasing superior accuracy and efficiency in the intricate task of classifying brain tumor images.

5.1.5. Comparison with classification using deep learning neural networks for brain tumors

In the pursuit of enhancing brain tumor classification in MRI images, two distinct methodologies have been explored. The first approach employs a DNN learning architecture with a multi-step process. It commences with the acquisition of a brain MRI dataset, followed by image segmentation using Fuzzy C-means. Feature extraction is then carried out through discrete wavelet transform (DWT), and dimensionality reduction is achieved using Principal Component Analysis (PCA). The final step involves classification using a DNN based on the World Health Organization's (WHO) classification system, specifically focusing on three types of malignant tumors: Glioblastoma, Sarcoma, and Metastatic Bronchogenic Carcinoma. Experimental results, conducted using MATLAB for the initial steps and Weka 3.9 for classification, showcase the effectiveness of the proposed methodology. The DNN classifier, in conjunction with DWT feature extraction, demonstrates superior performance across various metrics compared to alternative classifiers, The classification rate achieved is 96.97%, with a recall, precision, and F-Measure of 0.97 and AUC (ROC) of 0.984. On the other hand, a contrasting approach involves the construction of a pre-trained model for brain tumor classification. This method begins by downloading 7023 MRI brain images from Kaggle, encompassing both normal brains and those afflicted with various tumor types. The dataset undergoes preprocessing, including resizing and rescaling to align with a pre-trained model. Subsequently, the dataset is split into training and validation sets, where the former is utilized for training a model, and the latter is employed for evaluating model efficiency after each epoch. An additional dataset of 3160 MRI brain images is downloaded for further evaluation. The model is built by combining two pre-trained deep neural network models, InceptionV3 and MobileNetV3, with some last layers replaced for classification. The transfer learning model achieves an impressive accuracy of 96% and a low loss of 0.1. This approach leverages the strengths of pre-trained models to efficiently classify MRI brain images into four classes: normal brain, glioma tumor, meningioma tumor, and pituitary tumor. In comparing these two methodologies, the DNN-based approach demonstrates robust performance with explicit metrics, achieving high accuracy and reliability in classifying various types of brain tumors. On the other hand, the transfer learning model showcases efficiency and accuracy in classification, benefitting from the generalization capabilities of pre-trained deep neural networks. The choice between these approaches may depend on factors such as the interpretability of results, computational complexity, and the specific requirements of the medical imaging application.

5.1.6. Comparison with method using fine-tuning convolutional deep features to classify MRI brain tumor.

In the realm of brain tumor analysis, two distinctive methodologies have been employed, each contributing to the overarching goal of classification and prediction. The first methodology involves a meticulous data preprocessing phase, focusing on magnetic resonance imaging (MRI) slices depicting glioblastoma multiforme (GBM) cases. The extraction of region of interest (ROI) from these images, encompassing the brain tumor, is accompanied by normalization of gray level values and resizing to align with the requirements of a convolutional neural network (CNN). Two approaches, "exact" and "crop," are explored for creating tumor patches, demonstrating flexibility in adapting to varying tumor sizes. Feature extraction involves the utilization of a pre-trained CNN architecture (CNN-F), modified for grayscale images. Fine-tuning is performed, replacing the last fully connected layer for 2-class classification tasks. Results indicate that the pre-treatment features, especially from the T1-weighted and Flair sequences, achieve the highest accuracy. Concurrently, an alternative approach delves into the construction of a pre-trained model for brain tumor classification. Beginning with the acquisition of a diverse dataset of 7023 MRI brain images, comprising normal and tumor-afflicted brains, the preprocessing phase involves resizing and rescaling to align with a pre-trained model. The dataset is then partitioned into training and validation sets, with the former used for training a composite model comprising InceptionV3 and MobileNetV3. The resultant transfer learning model, with additional last replaced layers, exhibits an impressive accuracy of 96% and a low loss of 0.1. Upon comparison, the first methodology demonstrates meticulous handling of individual MRI slices, emphasizing feature extraction from pre-treatment images using a specialized CNN architecture. In contrast, the second approach leverages the strengths of pre-trained models, combining multiple architectures for an efficient transfer learning model. While the former achieves notable accuracy in predicting short-term or long-term survival, the latter excels in classifying brain images into specific tumor categories. The choice between these methodologies may depend on the specific objectives of the analysis, the nature of the available data, and computational considerations. The first approach emphasizes fine-tuning and customization of features, whereas the second approach capitalizes on the generalization capabilities of pre-trained models, resulting in an accurate and efficient classification system.

5.1.7. Comparison with method using pre-trained convolutional neural networks to classify of brain tumors.

The research conducted by Dmytro Filatov and Ghulam Nabi Ahmad Hassan Yar focuses on brain tumor diagnosis and classification using pre-trained convolutional neural networks (CNNs). The authors employ medical resonance imaging (MRI) and utilize CNNs with transfer learning from pre-trained models, specifically ResNet50 and EfficientNet, to achieve accurate diagnosis and classification of brain tumors. Their threefold goal includes diagnosing brain tumors, classifying specific tumor types (Glioma, Meningioma, and Pituitary), and optimizing efficiency through the use of pre-trained models. The methodology involves image augmentation in the pre-processing phase to address limited data challenges, the implementation of ResNet50 and EfficientNet models, and execution on Google Colab with specific hardware specifications.

On the other hand, the proposed method involves the acquisition of a dataset of 7023 MRI brain images from Kaggle, comprising normal brains and brains with tumors (Glioma, Meningioma, Pituitary). The pre-processing step involves resizing and rescaling the dataset to align with pre-trained models, followed by dataset splitting into training and validation sets. The model is built by combining two pre-trained deep neural network models, InceptionV3 and MobileNetV3, with additional layers for classification. The evaluation is performed on a separate dataset of 3160 MRI brain images from Kaggle. The results indicate a validation accuracy of 96.37%, validation loss of 0.1012, accuracy value from training process is 99.88%, and loss value from training process is 0.0120.

Comparatively, both methods employ pre-trained models for brain tumor classification, but with variations in the choice of models and dataset handling. While Filatov and Yar use ResNet50 and EfficientNet with image augmentation, the proposed method incorporates InceptionV3 and MobileNetV3 with dataset resizing and rescaling. The results show that the proposed method achieves a high validation accuracy of 96% and low validation loss, indicating robust performance. The comparison suggests that both approaches leverage transfer learning effectively, with the proposed method showcasing competitive results in terms of accuracy and loss metrics. Further analysis and exploration of these methods could provide insights into the strengths and limitations of different pre-trained models for this critical task in medical imaging.

Table 13: Classification results on different pre-trained models

Model	Epochs	Training Accuracy	Training Loss	Val Accuracy	Val Loss
EfficientNetB7	39	0.8419	0.5129	0.8818	0.2807
EfficientNetV2B1	43	0.8491	0.5695	0.8917	0.2768
EfficientNetB1	40	0.8767	0.4076	0.8955	0.3152
ResNet50	30	0.7282	0.6719	0.7932	0.4593
Combine model (MobileNetV3, InceptionV3)	30	0.9988	0.0120	0.9637	0.1012

5.2. Evaluation

The evaluation of the proposed methodology for building a pre-trained model in this research involves a systematic and comprehensive process. The initial step entails the acquisition of a substantial dataset comprising 7023 MRI brain images from Kaggle, a publicly available repository. This dataset is diverse, encompassing images of both normal brains and brains afflicted with various tumor types, including glioma tumor, meningioma tumor, and pituitary tumor. The inclusion of such diverse cases ensures a robust evaluation of the model's performance across different pathological conditions.

Following data acquisition, the preprocessing phase is meticulously executed. The dataset undergoes resizing and rescaling procedures to conform to the specifications of the pre-trained model. This step is crucial for ensuring compatibility and optimal utilization of the features learned by the pre-existing neural network architectures, InceptionV3 and MobileNetV3.The dataset is then judiciously partitioned into two subsets: train set and validate set, signifying a principled division for training and evaluation purposes. The train set serves as the foundation for training the pre-trained deep neural network model, leveraging the wealth of information contained within the MRI images. Concurrently, the validate set is utilized to assess the model's efficiency after each epoch of training, providing insights into its learning trajectory and generalization capabilities.

Furthermore, to enhance the model's versatility and evaluation robustness, an additional dataset consisting of 3160 MRI brain images is downloaded from Kaggle. This dataset mirrors the diversity of the original dataset, featuring normal brains and brains with different tumor types.

It is used to assess how effectively the trained model performs, ensuring that the model's predictive capabilities extend beyond the training data.

The final stage involves the construction of the model, wherein the features learned from both InceptionV3 and MobileNetV3 are amalgamated. Additionally, the model is equipped with new last replaced layers tailored to classify MRI brain images into four distinct classes: normal brain, glioma tumor, meningioma tumor, and pituitary tumor. The transfer learning model exhibits an impressive accuracy of 96% and a low loss of 0.1, validating the effectiveness of the proposed methodology.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Conclusion

In conclusion, this thesis has introduced and explored a novel hybrid approach in the realm of convolutional neural networks (CNNs) for brain tumor classification. The focal point was the integration of two pre-trained models InceptionV3 and MobileNetV3, through transfer learning, aiming to leverage their distinctive strengths in feature extraction and computational efficiency. The resulting combined CNN architecture, as depicted in table11, demonstrated a comprehensive feature representation, anticipating enhanced performance across various tasks related to brain MRI image classification. The detailed analysis encompassed fine-tuning strategies, performance metrics, and observed synergies between the integrated models. Through systematic experiments, we sought to gain insights into the potential benefits and challenges associated with fusing diverse pre-trained models in the realm of CNNs.

The proposed architecture effectively utilized InceptionV3 and MobileNetV3, each with its pre-trained layers and incorporated two fully connected layers for comprehensive brain tumor classification. The strategic division of the dataset into training, validation and test sets facilitated a thorough evaluation of the model's performance. The experimental results, showcasing a categorical accuracy of 0.9637 on the validation set and a test set accuracy of 0.9380, underscore the efficacy of the proposed hybrid CNN model in accurately discerning between normal brain and brain tumors, including glioma, meningioma, and pituitary tumors.

The achievement of high accuracy on both the validation and test sets suggests the robustness and generalizability of the model, affirming its potential for real-world applications in medical image analysis. However, ongoing research efforts should continue to explore the model's adaptability to diverse datasets and its scalability to address a broader range of clinical scenarios. In essence, this work contributes to the evolving field of medical image analysis by presenting an innovative approach that combines pre-trained models, paving the way for future advancements in the development of more effective and versatile CNN architectures for brain tumor classification.

6.2. Future work

Future work in the field of brain tumor classification using convolutional neural networks (CNNs) holds promising avenues for advancement and refinement. One crucial area of exploration involves enhancing the interpretability of the model's decisions. Developing methodologies that provide insights into the reasoning behind the CNN's classifications, such as incorporating attention mechanisms or layer-wise relevance propagation, can contribute to better understanding and acceptance by medical professionals.

Another compelling direction for future research is the exploration of multi-center datasets. Extending the study to incorporate data from different medical institutions with diverse imaging protocols, scanners, and patient demographics can improve the generalizability of the model. Addressing the challenges posed by such variability will be essential in ensuring the reliability and effectiveness of the CNN across a broader spectrum of clinical scenarios. Additionally, efforts to validate the model's performance on external datasets and across different populations will contribute to its robustness and real-world applicability. These future endeavors are crucial steps toward advancing the field of medical image analysis and establishing CNN-based models as valuable tools in clinical practice.

REFERENCES

[1] Keiron O'Shea, Ryan Nash, "An Introduction to Convolutional Neural Networks," *arXiv preprint arXiv:1511.08458,* 2015.

[2] Vinod Nair, Geoffrey E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th international conference on machine learning (ICML-10),* pp. 807-814, 2010.

[3] Sakib Mostafa, Fang-Xiang Wu, "Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images," *Neural Engineering Techniques for Autism Spectrum Disorder,* 2021.

[4] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *Neurocomputing: Algorithms, architectures and applications,* pp. 227-236, 1990.

[5] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, no. 6088, pp. 533-536, 1986.

[6] "Indian Tech Warrior," 22 8 2024. [Online]. Available: https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/.

[7] Leixian Shen, Qingyun Zhang, Guoxu Cao, He Xu, Fall Detection System Based on Deep Learning and Image Processing in Cloud Environment, Springer International Publishing, 2019, pp. 590 - 598.

[8] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint arXiv:1502.03167,* 2015.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[10] baeldung, "Baeldung on Computer Science," 14 April 2023. [Online]. Available: https://www.baeldung.com/cs/ml-relu-dropout-layers.

[11] Sinno Jialin Pan, Qiang Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 22, no. 10, pp. 1345-1359, 2010.

[12] Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson, "How Transferable Are Features in Deep Neural Networks?," *Advances in Neural Information Processing Systems,* pp. 3320-3328, 2014.

[13] "Towards Data Science," 17 November 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2818-2826, 2016.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *2009 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 248-255, 2009.

[16] Luqman Ali, Fady Alnajjar, Hamad Jassmi, Munkhjargal Gochoo, Wasif Khan, Mohamed Serhani, "Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures," *Sensors,* vol. 21, no. 5, p. 1688, 2021.

[17] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam., "Searching for MobileNetV3," *IEEE/CVF International Conference on Computer Vision*, pp. 1314-1324, 2019.

[18] Linhui Wang, Wangpeng Shi, Yonghong Tang, Zhizhuang Liu, Xiongkui He, Hongyan Xiao, Yu Yang, "Transfer Learning-Based Lightweight SSD Model for Detection of Pests in Citrus," *Agronomy,* vol. 13, no. 7, p. 1710, 2023.

[19] Ghinwa M. Tarhini and Reda Shbib, "Detection of Brain Tumor in MRI Images Using Watershed and Threshold-Based Segmentation," *International Journal of Signal Processing Systems,* vol. 8, no. 1, pp. 19-25, 2020.

[20] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62-66, 1979.

[21] Serge Beucher, Christian Lantuéjoul, "Use of Watersheds in Contour Detection," *Proc. Internationa Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation,* 1979.

[22] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 3rd ed., Prentice Hall, 2008.

[23] Maria Sokolova, Guy Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management,* vol. 45, no. 4, pp. 427-437, 2009.

[24] Heba Mohsen , El-Sayed A. El-Dahshan, El-Sayed M. El-Horbaty, Abdel-Badeeh M. Salem , "Classification using deep learning neural networks for brain tumors," *Future Computing and Informatics Journal,* vol. 3, no. 1, pp. 68-71, 2018.

[25] A. R. Kavitha, L. Chitra, R. Kanaga, "Brain Tumor Segmentation using Genetic Algorithm with SVM Classifie," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 5, pp. 1468-1473, 2016.

[26] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[27] Kaoutar Ben Ahmed, Lawrence O. Hall, Dmitry B. Goldgof, Renhao Liu, Robert A. Gatenby, "Fine-tuning convolutional deep features for MRI-based brain tumor classification," *Medical Imaging 2017: Image Processing,* vol. 10134, 2017.

[28] Inayatul Haq, Najib Ullah, Tehseen Mazhar, Amir Malik, Iqra Bano, "A Novel Brain Tumor Detection and Coloring Technique from 2D MRI Images," *Applied Sciences,* vol. 12, no. 11, p. 5744, 2022.

[29] Dmytro Filatov, Ghulam Nabi Ahmad Hassan Yar, "Brain Tumor Diagnosis and Classification via Pre-Trained Convolutional Neural Networks," *ArXiv,* vol. abs/2208.00768, 2022.

[30] M. NICKPARVAR, "kaggle Brain Tumor MRI Dataset," [Online]. Available: https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset.

[31] SARTAJ, "Kaggle Brain Tumor Classification (MRI) datasets," 2020. [Online]. Available: https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri/discussion.