



University of Science and Technology of Hanoi
Department of Information and Communication Technology

Bachelor thesis

Brain tumor classification using deep learning

Authors:
Nguyen The Hoang – BI12-171

Information and Communications Technology
Academic years: 2021 - 2024

Internal supervisor:
Dr. Nghiêm Thị Phương

Hanoi, 2024

Declaration

I hereby, Nguyen The Hoang, declare that my thesis represents my own work after doing an internship at ICTLab, USTH.

I have read the University's internship guidelines. In the case of plagiarism appearing in my thesis, I accept responsibility for the conduct of the procedures in accordance with the University's Committee.

Hanoi, September 2024

Nguyen The Hoang

Acknowledgment

First and foremost, I want to express my appreciation to Dr. Nghiêm Thị Phuong, my kindly supervisor, for supporting me at my project. I am grateful to have her as my supervisor for the internship.

Secondly, I would like to thank the University of Science and Technology (USTH) as a whole and specifically the ICT Lab for providing me with valuable knowledge, experiences, and the opportunity to work in a professional environment as a temporary member of the lab.

Finally, I would like to sincerely appreciate all the support and assistance I received from my family and friends to finish this project.

While this project represents a significant milestone, I acknowledge that there may be areas for further improvement, and I remain open to constructive feedback and suggestions to enhance my knowledge and skills.

Hanoi, September 2024

Nguyen The Hoang

Contents

Acronyms	5
List of Tables	6
List of Figures	7
1 Introduction	10
1.1 Context and Motivation	10
1.2 Project Scope	11
1.3 Thesis Organization	12
2 Materials and Methods	13
2.1 Materials	13
2.1.1 Dataset	13
2.1.2 Date Preprocessing	14
2.2 Methods	15
2.2.1 Convolutional Neural Network (CNN)	15
2.2.2 Convolution Layers	16
2.2.3 Activation Layer	16
2.2.4 Pooling Layer	17
2.2.5 Flatten	18
2.2.6 Softmax	18
2.2.7 Fully Connected Layers	18
2.2.8 Dropout	19
2.3 Model Architecture	20
2.3.1 Custom CNN	20
2.3.2 ResNet101	20
2.3.3 InceptionV3	21
2.3.4 Transfer Learning	22
2.4 Metrics	24

3 BACKGROUND KNOWLEDGE	27
3.1 Brain tumors	27
3.1.1 Definition and location	27
3.1.2 Nature of brain tumors	27
3.2 Types of brain tumors	28
3.2.1 Glioma Brain Tumors	28
3.2.2 Meningiomas Brain tumors	29
3.2.3 Pituitary brain tumors	30
3.3 Magnetic resonance imaging (MRI)	31
3.3.1 What is MRI	31
3.3.2 Basic principles	32
3.3.3 MRI imaging sequences	32
3.3.4 Properties of MRI images	33
3.3.5 Limitation of MRI images	33
4 Results and Discussions	34
4.1 Experiment Setup	34
4.1.1 Data	34
4.1.2 Framework	36
4.1.3 Model	37
4.2 Result	40
4.2.1 ResNet101	40
4.2.2 InceptionV3	43
4.2.3 Custom CNN	46
4.3 Discussion	49
5 Conclusion and Future Work	50
5.1 Conclusion	50
5.2 Future work	50
Bibliography	51

List of Acronyms

- **Malignant:** Cancerous
- **Benign:** Non-cancerous
- **Axial:** From top to down
- **Coronal:** Side to side
- **Sagittal:** From front to back
- **CNN:** Convolutional Neural Network
- **ReLU:** Rectified Linear Unit
- **MRI:** Magnetic Resonance Imaging

List of Tables

2.1	Basic Structure of a Confusion Matrix	25
2.2	Confusion Matrix for a Multi-Class Problem	25
3.1	Most common MRI Sequences and their Approximate TR and TE times	33
4.1	Classification Metrics of ResNet101	41
4.2	Performance metrics of InceptionV3	44
4.3	Classification Metrics of Custom CNN	47
4.4	Comparison of Classification Metrics	49

List of Figures

2.1	Sample MRI images of the dataset	13
2.2	Preprocessing example	14
2.3	General CNN architecture	15
2.4	Filter, input and output of the convolution operation	16
2.5	Some activation functions	17
2.6	Example of Max pooling	17
2.7	Flattening in CNN	18
2.8	Fully Connected Layer in CNN	19
2.9	Dropout layer in CNN	19
2.10	ResNet101 architecture	21
2.11	Original Inception architecture	21
2.12	InceptionV3 architecture	22
2.13	Transfer learning Strategy	23
2.14	Performance with and without applying Transfer learning	23
3.1	Glioma Brain tumor	28
3.2	Meningiomas Brain tumor	29
3.3	Pituitary Brain tumor	30
3.4	MRI sample picture	31
3.5	Example MRI images of 3 brain tumors types	32
4.1	Combined Dataset	34
4.2	Testing Labels After Combined	35
4.3	Validation Labels After Combined	35
4.4	Training Labels After Combined	36
4.5	Training Labels After Augmented	36
4.6	Training - Validation Accuracy and Loss of ResNet101	40
4.7	Confusion matrix of ResNet101	41
4.8	ResNet101 ROC Curves	42
4.9	Training - Validation Accuracy and Loss of InceptionV3	43
4.10	Confusion matrix of InceptionV3	44

4.11	InceptionV3 ROC Curves	45
4.12	Training - Validation Accuracy and Loss of Custom CNN	46
4.13	Confusion matrix of Custom CNN	47
4.14	Custom CNN ROC Curves	48

Abstract

Brain tumors are a dangerous and potentially fatal disorder that causes a large amount of death globally. In 2022, an estimated 321,731 new cases were reported, with 248,500 deaths attributed to brain tumors. Due to their location in the brain, which regulates vital body functions, brain tumors, whether malignant or benign, can be very harmful. As the disease advances, the prognosis frequently gets worse, thus early detection and treatment are essential.

To identify these kinds of brain tumors and establish whether they are benign or malignant, radiologists commonly employ imaging methods like MRIs and CT scans to assess the features of brain tissues, such as the size, form, and location of the tumors. As a result of technological developments, brain tumor detection, and tracking are using more and more imaging modalities.

In this context, our project focuses on the application of deep learning techniques for the classification of brain tumors. We utilize MRI images to train and evaluate two state-of-the-art deep learning models: ResNet101 and Inception-V3.

Keywords : Brain tumors classification, Glioma brain tumors, Meningioma brain tumors, Pituitary brain tumors, Deep learning, MRI, Convolutional neural networks.

Chapter 1

Introduction

1.1 Context and Motivation

Brain tumors have become a global health concern in recent years, impacting an increasing number of individuals. These tumors, caused by the abnormal and uncontrolled growth of cells within or around the brain, can lead to serious consequences if not identified and treated early. Improving outcomes, averting long-term impairments, and possibly saving lives all depend on early detection. This emphasizes how crucial it is to use modern methods for brain tumor early detection and accurate classification, such as MRI imaging and machine learning.

The Global Cancer Observatory(GLOBALCAN) estimates that there were about 321,731 new cases of primary malignant brain and central nervous system (CNS) tumors in 2022 alone, which resulted in 248,500 fatalities [1]. It is concerning that this incidence is rising and highlights the need for earlier diagnosis and better treatment alternatives[2]. The kind, location, and stage of the tumor at diagnosis all have a significant impact on the survival rates of individuals with brain tumors.

Magnetic Resonance Imaging (MRI) remains one of the most effective tools for diagnosing brain tumors [3]. It produces detailed images of the brain using powerful magnetic fields, enabling doctors to accurately assess the size, location, and shape of tumors. MRI's superior ability to visualize soft tissues makes it invaluable for understanding the characteristics of brain tumors. While MRI does have some drawbacks, such as longer scan times and potential discomfort for patients, it continues to be the gold standard for brain imaging due to its precision and reliability.

Despite the crucial insights provided by MRI scans, the volume and complexity

of these images can pose challenges for radiologists, making manual interpretation time-consuming and prone to error. This is where machine learning becomes a valuable asset. As a subset of artificial intelligence (AI), machine learning has the capacity to automatically analyze medical images and identify patterns that may be difficult for the human eye to detect[4]. By training models on labeled MRI images, machine learning algorithms can classify tumors based on their unique characteristics.

A major challenge in building effective machine-learning models for brain tumor classification is the limited availability of large, diverse medical imaging datasets[5]. These datasets often suffer from class imbalances or lack sufficient variability, which can negatively impact the model's performance. For instance, if a dataset contains a disproportionate number of images of benign tumors compared to malignant ones, the model may struggle to correctly identify malignant cases. To address this, researchers use data augmentation techniques. Data augmentation involves applying transformations such as rotations, flips, scaling, and zooming to existing MRI images, generating new variations. This helps artificially expand the dataset, introducing more variability and improving the model's ability to generalize to unseen data.

Another strategy to combat the issue of limited data is transfer learning[6]. Transfer learning uses pre-trained models that have been trained on large, diverse datasets, like ImageNet, and fine-tunes them for specific medical tasks such as brain tumor classification. This enables the model to leverage learned features from a general dataset while adapting to the specific domain of medical imaging.

The rising prevalence of brain tumors underscores the critical importance of early detection. While MRI remains the gold standard for diagnosing brain tumors, interpreting the images efficiently and accurately requires the support of advanced tools. Machine learning has emerged as a powerful solution for classifying brain tumors, offering the potential to improve both the speed and accuracy of diagnoses, ultimately enhancing patient care.

1.2 Project Scope

The main objectives are expected to be achieved in this internship:

- Using CNN model and pre-train model to classification Brain tumors MRI images.

-
- Calculate classification performance and compare performance between models.

1.3 Thesis Organization

This thesis is divided into main chapters:

- Chapter 1: Introduces an overview of the internship including project scope.
- Chapter 2: Reviews materials used in this internship and methodology carried out for completing the project.
- Chapter 3: Background knowledge related to Brain tumors.
- Chapter 4: Analyses and discusses the experimental results obtained.
- Chapter 5: Concludes the report and presents future works.

Chapter 2

Materials and Methods

2.1 Materials

2.1.1 Dataset

Brain tumor classification (MRI) The dataset is available from Kaggle under the authority of Sartaj, is a comprehensive resource designed to aid in the accurate identification and classification of brain tumors[7]. This dataset, derived from MRI scans, is specifically structured to support the categorization of brain tumors into four distinct types: meningioma, glioma, pituitary tumor, and the absence of a tumor. Each category represents a critical aspect of brain tumor classification, reflecting the diversity and complexity found in real-world clinical environments. The inclusion of both tumor types and non-tumor images enhances the challenge, requiring models to accurately differentiate between various tumor types and identify cases where no tumor is present.

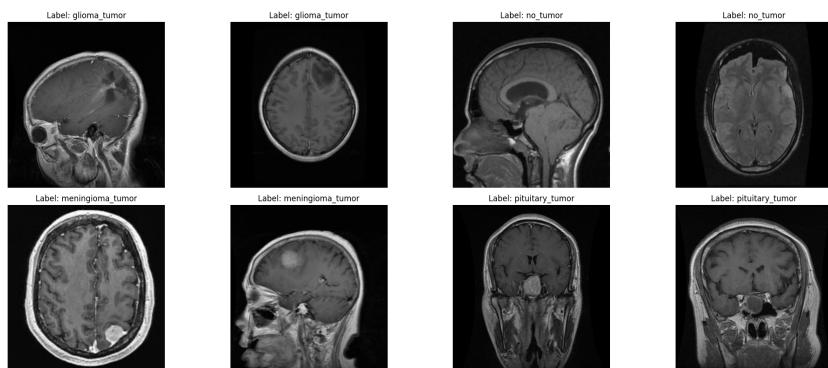


Figure 2.1: Sample MRI images of the dataset

The Brain Tumor Classification MRI dataset is particularly valuable for addressing some of the most pressing challenges in medical imaging. Accurate detection and classification of brain tumors are critical for patient outcomes, as they directly influence treatment decisions and prognosis. By providing a high-quality, diverse dataset, this resource supports the development of models that can potentially improve diagnostic accuracy, reduce human error, and enhance the efficiency of clinical workflows.

2.1.2 Date Preprocessing

Deep learning generally requires a large amount of input data and also they need to be balanced to achieve considerable accuracy. The best way to improve our model is to get more data. One common technique to deal with the problem of small and imbalance data is to generate the image by performing extensive data augmentation. It helps us to increase the size of the data set quickly while changing and adding new properties to the existing data to make the data abundance more abundant. Some of the techniques to augment training data are described as below:

- Shift horizontally and vertically: 10%
- Randomly flipping horizontally and vertically
- Randomly zooms in and out on images by up to 20%
- Randomly rotates images by up to 20 degrees

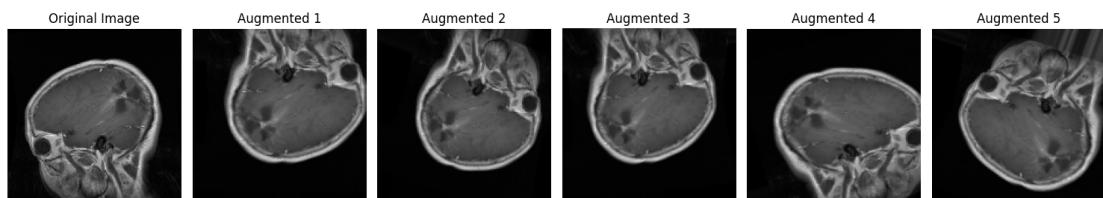


Figure 2.2: Preprocessing example

These techniques collectively enhance the diversity of the training data and allow the model to generalize better to new, unseen images.

2.2 Methods

2.2.1 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN or ConvNet) is a deep learning algorithm specifically designed for any task where object recognition is crucial such as image classification, detection, and segmentation. Many real-life applications, such as self-driving cars, surveillance cameras, and more, use CNN.

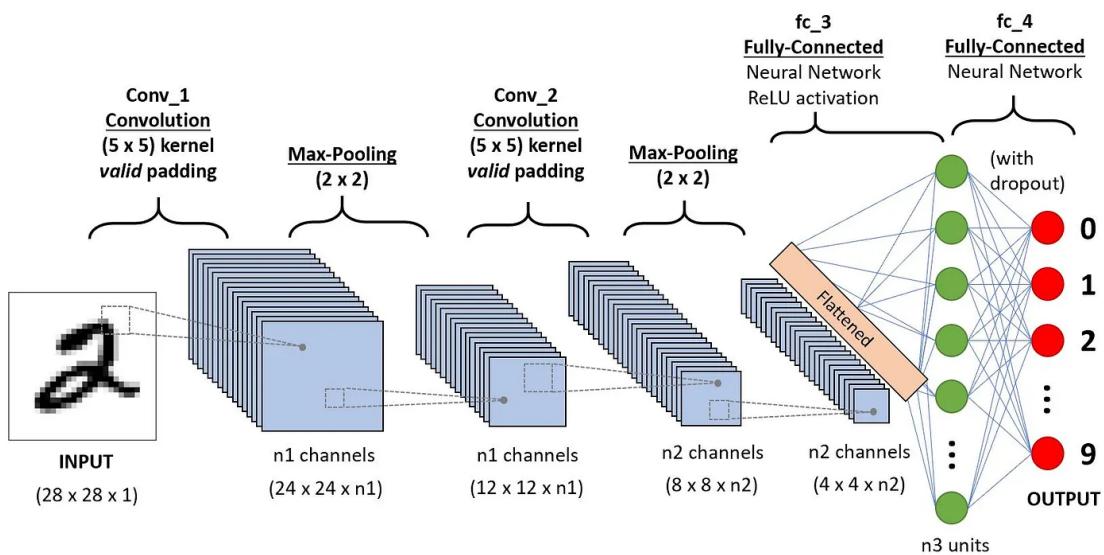


Figure 2.3: General CNN architecture

CNNs' architecture tries to mimic the structure of neurons in the human visual system composed of multiple layers, where each one is responsible for detecting a specific feature in the data. As illustrated in the image above, the typical CNN is made of a combination of four main layers:

- Convolutional layers
- Activation layer
- Pooling layers
- Fully connected layers

2.2.2 Convolution Layers

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. The sliding function applied to the matrix is called kernel or filter, and both can be used interchangeably, which is the application of a sliding window function to a matrix of pixels representing an image.

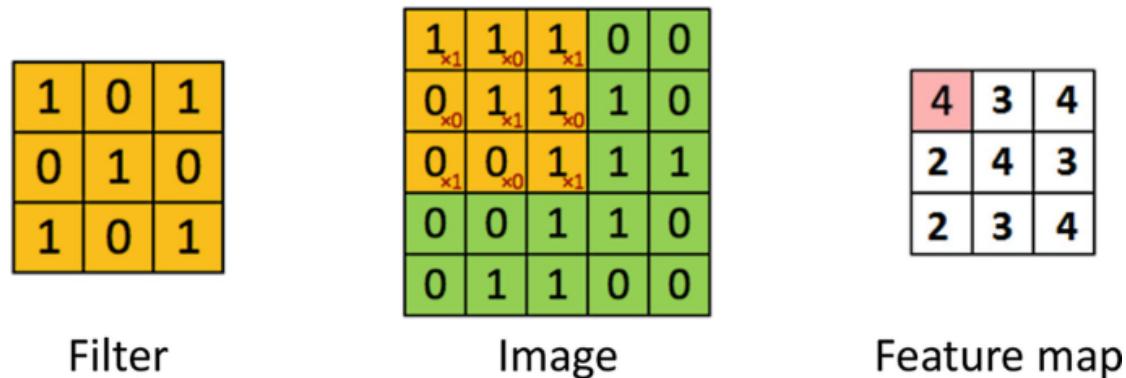


Figure 2.4: Filter, input and output of the convolution operation

The kernel matrix, or the filter, will go from the top-left corner to the right then from up to down, then will perform element-wise multiplication and sum the values of the product to get a specific number. This number represents a feature detected at that position in the image. The set of all these specific numbers, after the kernel has passed over the image called feature map.

2.2.3 Activation Layer

After each convolution operation, an activation layer will be applied, the result will pass through an activation function, the network will learn non-linearity relationships between the features in the image, hence making the network more robust for identifying different patterns. There are many activation layers such as:

- Step function
- Sigmoid function
- ReLU function

The most famous and widely used activation layer is ReLU (Rectified Linear Unit). It is defined as:

$$f(x) = \max(0, x)$$

meaning it outputs the input value if it is positive, and zero otherwise. This simplicity makes ReLU computationally efficient and easy to implement, significantly speeding up the training of neural networks.

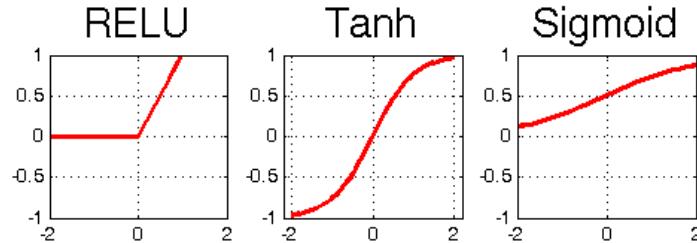


Figure 2.5: Some activation functions

2.2.4 Pooling Layer

The goal of pooling layer is to extract the most significant features from the feature map generated by the convolutional layer. This is done by applying some aggregation operations, which reduces the spatial dimensions of the feature map. By reducing the size, pooling not only lowers the memory and computational resources needed during training but also helps to generalize the model by preventing overfitting. This is because pooling layers simplify the network by reducing the number of parameters and retaining only the most important features.

The most common aggregation functions that can be applied are:

- Max pooling: the maximum value of the feature map
- Average pooling: the average of all the values
- Sum pooling: sum of all the values of the feature map

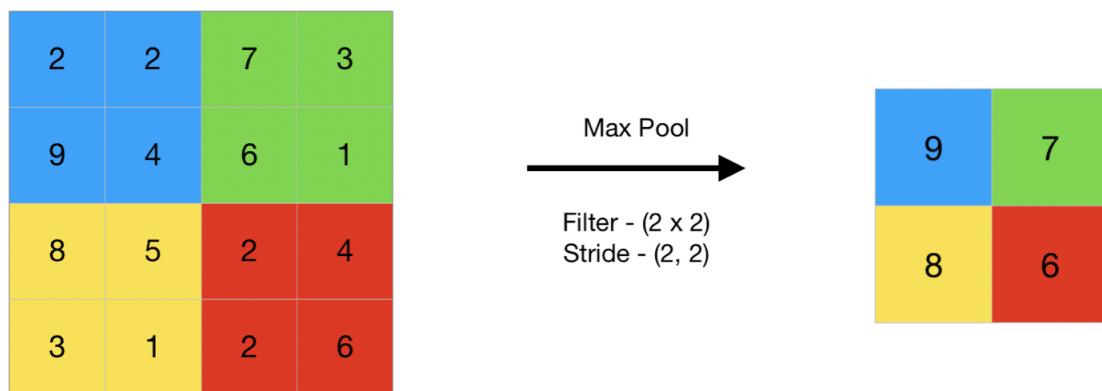


Figure 2.6: Example of Max pooling

2.2.5 Flatten

The Flatten layer is a crucial component in neural networks, particularly when transitioning from convolutional layers to fully connected (dense) layers. The Flatten layer converts the multi-dimensional output of previous layers (such as convolutional or pooling layers) into a one-dimensional vector. This step is essential because fully connected layers require a one-dimensional input to perform their computations.

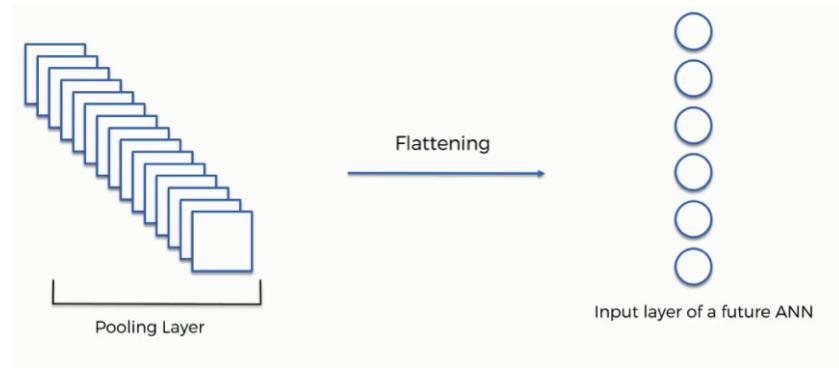


Figure 2.7: Flattening in CNN

2.2.6 Softmax

Softmax commonly used as the final layer of a neural networks for classification tasks. The softmax function takes the output of the last fully connected layer and converts it into a probability distribution across all possible classes. Each output value is transformed into a probability between 0 and 1, and the sum of all probabilities equals 1. The class with the highest probability is chosen as the network's prediction. The softmax layer is crucial for multi-class classification tasks, as it provides a clear probabilistic interpretation of the network's predictions.

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

2.2.7 Fully Connected Layers

Fully connected layers, also called dense layers, are typically used in the final stages of CNN. In a fully connected layer, each neuron is connected to every neuron in the previous layer, forming a dense structure. This setup allows the model to learn complex relationships between features by combining information from all neurons in the preceding layer. The fully connected layers are responsible for

making final predictions or decisions based on the high-level features extracted by the earlier layers of the network.

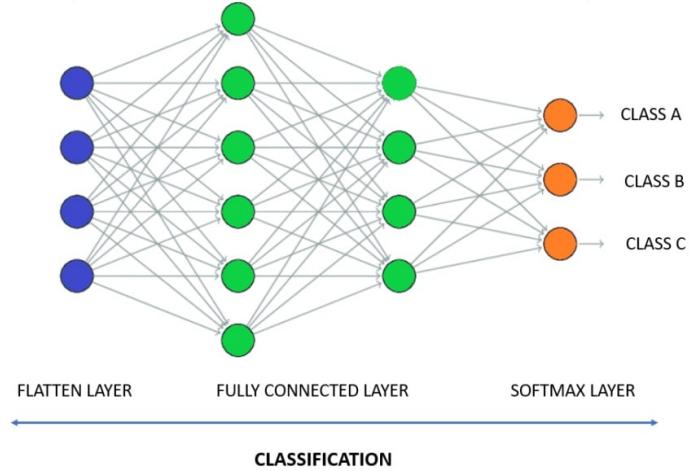


Figure 2.8: Fully Connected Layer in CNN

2.2.8 Dropout

The term dropout refers to dropping out units (both hidden and visible) in a neural network. Dropout is a regularization technique used to improve the generalization capability of neural networks, especially those with a large number of parameters. During training, dropout randomly drops (i.e., sets to zero) a fraction of the neurons in the network at each training iteration. This process forces the remaining neurons to adapt and learn more robust features from the input data, rather than relying on specific neurons. By preventing neurons from co-adapting too closely, dropout helps reduce overfitting, leading to better performance on unseen data.

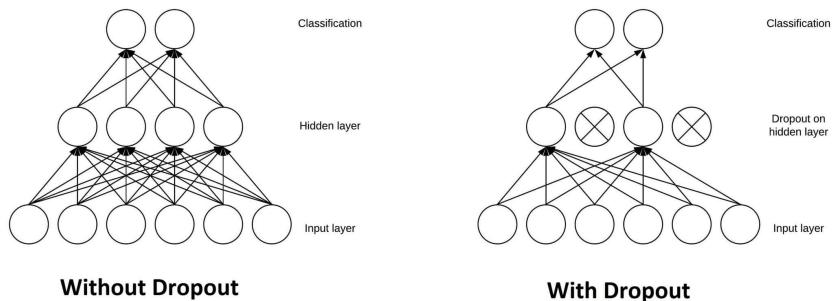


Figure 2.9: Dropout layer in CNN

2.3 Model Architecture

2.3.1 Custom CNN

I designed a custom CNN architecture to specify for classification brain tumor. The input to the model consisted of RGB images resized to 150x150 pixels. The architecture comprised three convolutional layers, each followed by max-pooling:

- First Convolutional Block: A 2D convolutional layer with 32 filters of size 3x3, followed by a ReLU activation and 2x2 max-pooling.
- Second Convolutional Block: A 2D convolutional layer with 64 filters (3x3), followed by ReLU activation and 2x2 max-pooling.
- Third Convolutional Block: A final 2D convolutional layer with 64 filters (3x3), followed by ReLU activation and 2x2 max-pooling.

The convolutional blocks were followed by a flatten layer, which converted the feature maps into a 1D vector. This was passed through a fully connected dense layer with 64 units and a ReLU activation. A dropout rate of 0.5 was applied to prevent overfitting. Finally, a softmax layer with four units was used for classification into one of the four tumor types.

The model was compiled using the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy loss. The model was trained for 25 epochs with a batch size of 32, and accuracy was used as the performance metric.

2.3.2 ResNet101

ResNet101 is a deep convolutional neural network that is part of the ResNet (Residual Network) family, which is a widely used convolutional neural network (CNN) architecture in the field of computer vision[8].

ResNet revolutionized deep learning by solving the challenges of training very deep neural networks, particularly the vanishing gradient and degradation problems. By introducing residual connections, ResNet allowed information and gradients to flow more easily through the network, enabling the training of deeper models without a drop in performance. This innovation led to significantly improved accuracy in tasks like image classification, as demonstrated by ResNet's success in the ImageNet challenge. Its design principles have since become a foundation for modern neural network architectures, making ResNet one of the most influential advancements in deep learning.

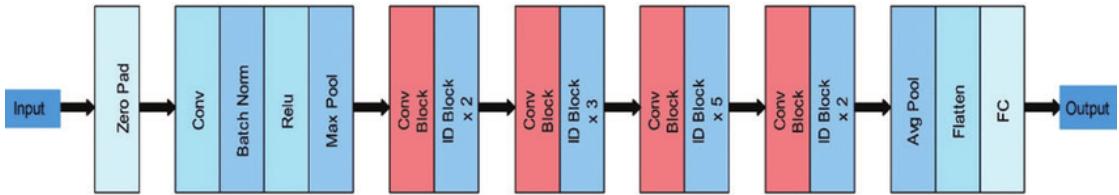


Figure 2.10: ResNet101 architecture

ResNet101 refers to a deep convolutional neural network (CNN) from the Residual Network (ResNet) family that consists of 101 layers. ResNet101 is highly suitable for brain tumor classification using deep learning. Its ability to learn detailed representations and apply transfer learning makes it a powerful tool for detecting and classifying tumors from MRI scans, leading to improved diagnosis and treatment planning.

2.3.3 InceptionV3

InceptionV3 is a deep convolutional neural network architecture that's widely used for image classification tasks. Developed by Google, it builds on the Inception series of models, which are known for their efficient and effective approach to deep learning[9]. Inception V3 base on the original Inception (GoogLeNet) architecture.

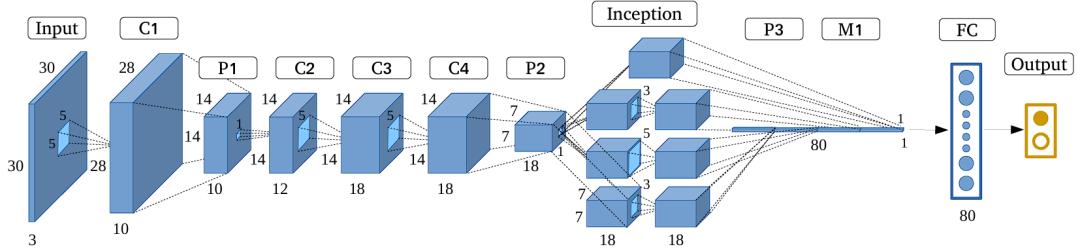


Figure 2.11: Original Inception architecture

The InceptionV3 network consists of 42 layers organized into multiple Inception modules, which are its key architectural feature. These modules use parallel convolutions with different filter sizes (1x1, 3x3, and 5x5) to capture multi-scale information efficiently. To reduce computational cost, InceptionV3 employs factorized convolutions, breaking larger filters into sequences of smaller ones (e.g., a 5x5 convolution becomes two 3x3 convolutions). The architecture also introduces batch normalization after each convolutional layer to stabilize training and improve generalization. Auxiliary classifiers are integrated at intermediate stages to combat

vanishing gradients in this deep network. The model begins with standard convolutional and pooling layers, followed by stacked Inception modules interspersed with reduction modules for downsampling, and concludes with global average pooling and fully connected layers for classification. When trained on the ImageNet dataset, InceptionV3 achieved a top-5 error rate of 3.46%, marking a significant improvement over its predecessors and establishing it as a benchmark in computer vision tasks.

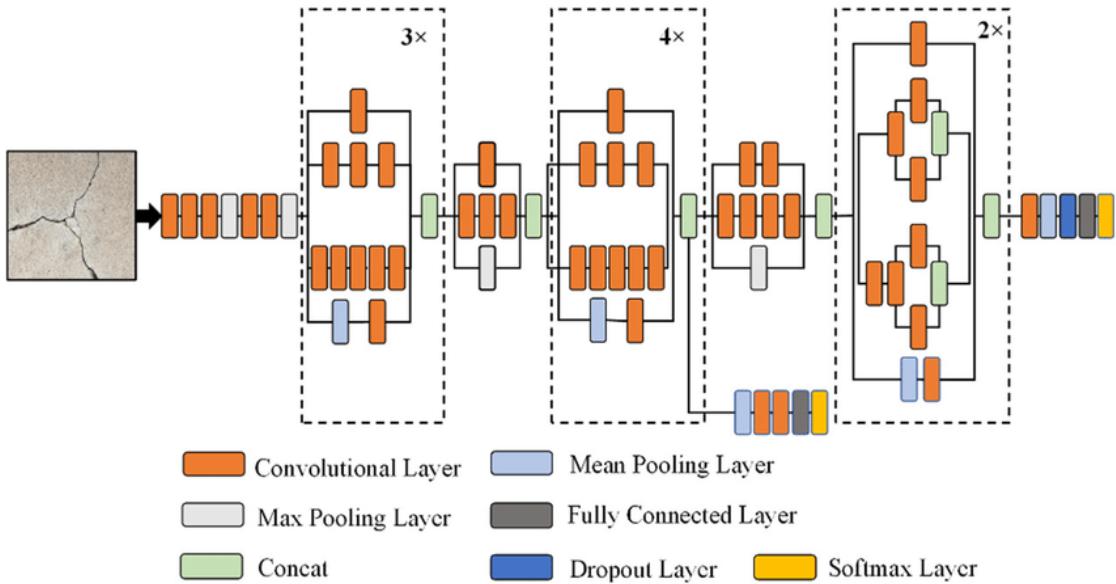


Figure 2.12: InceptionV3 architecture

2.3.4 Transfer Learning

Transfer learning is the re-use of a pre-trained model on a new problem. It's popular in deep learning because it can train deep neural networks with comparatively little data. This is very useful in the data science field since most real-world problems typically do not have millions of labeled data points to train such complex models.

So how does Transfer learning work? Transfer learning involves leveraging a pre-trained model to address a new, yet related problem, making it a powerful approach in deep learning. This method capitalizes on the knowledge acquired from a previous task to enhance performance on a different but related task, especially when the new task lacks extensive data. Instead of building a model from scratch, transfer learning utilizes patterns learned from solving one problem to improve generalization on another. This technique is particularly valuable in domains like computer vision and natural language processing, where large datasets

and substantial computational resources are typically required. Although not a standalone technique, transfer learning serves as a strategic methodology within machine learning, enabling more efficient use of pre-trained models to tackle new challenges.

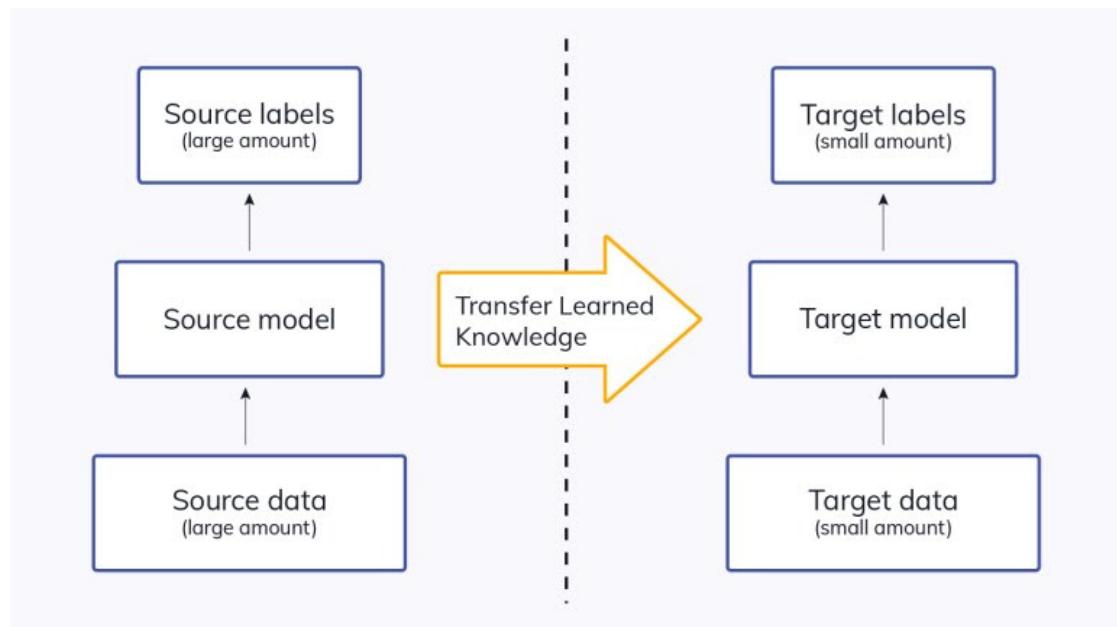


Figure 2.13: Transfer learning Strategy

So, if we use Transfer learning the correct way, like in the case of a cat and dog classification problem, it will require fewer epochs to achieve the desired accuracy compared to training the model from scratch. In fact, the accuracy may even surpass what could be achieved without applying Transfer learning.

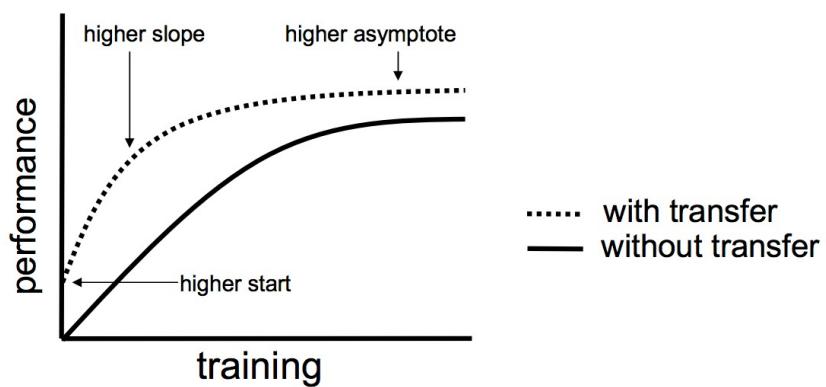


Figure 2.14: Performance with and without applying Transfer learning

2.4 Metrics

To evaluate the performance of our implemented method on the dataset, I have used the following metrics to assess the model's effectiveness in classification:

- **Accuracy:** Represents the overall correctness of the model in classifying tumors. It is calculated as the ratio of correct predictions to the total number of predictions.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2.4.1)$$

- **Sensitivity (Recall):** Measures the ability of the model to correctly identify positive cases (i.e., tumors). This metric is especially important for minimizing false negatives, ensuring that tumors are not missed.

$$\text{Sensitivity (Recall)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2.4.2)$$

- **Specificity:** Indicates the model's ability to correctly identify negative cases, reducing false positives in tumor classification.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \quad (2.4.3)$$

- **Precision:** Reflects the proportion of true positive predictions out of all predicted positives. A higher precision indicates fewer false positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2.4.4)$$

- **F1 Score:** Provides a balanced measure of the model's performance by calculating the harmonic mean of precision and recall, especially useful when dealing with imbalanced datasets.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4.5)$$

- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** This metric evaluates the model's ability to distinguish between classes (tumor vs. non-tumor) across various classification thresholds, with higher values indicating better discriminatory power.

Last but not least is **Confusion Matrix**, the confusion matrix is a powerful tool used to evaluate the performance of classification models by providing a detailed breakdown of correct and incorrect predictions. Unlike individual metrics such as accuracy, precision, or recall, the confusion matrix offers a comprehensive view by displaying the number of true positives, true negatives, false positives, and false negatives.

The basic structure of a confusion matrix:

- True Positive (TP) - Your model predicted the positive class
- True Negative (TN) - Your model correctly predicted the negative class
- False Positive (FP) - Your model incorrectly predicted the positive class
- False Negative (FN) - Your model incorrectly predicted the negative class

Based on these 4 outcomes, the performance of the classification model can be evaluated using their 2×2 formulated table called a binary confusion matrix. This matrix provides a clear and organized view of how well the model distinguishes between positive and negative cases, and it serves as the foundation for calculating key performance metrics such as accuracy, precision, etc.

	Predicted Positive	Predicted Negative
Actual Positive	True Positives (TP)	False Negatives (FN)
Actual Negative	False Positives (FP)	True Negatives (TN)

Table 2.1: Basic Structure of a Confusion Matrix

However, our case involves a multi-class problem, so we cannot apply the same theory used for binary classification directly. In a multi-class setting, there are no clear positive or negative classes, which complicates the identification of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). However, the solution is straightforward: we need to calculate these four categories (TP, TN, FP, FN) for each individual class.

True Class / Predicted Class	Dog	Cat	Mouse
Dog	50	5	3
Cat	4	45	6
Mouse	2	7	40

Table 2.2: Confusion Matrix for a Multi-Class Problem

For example, consider a dataset with three classes: Dog, Cat, and Mouse. The confusion matrix for such a multi-class problem will look different from the binary case, and we need to analyze each class separately to determine the TP, TN, FP, and FN values. We calculate:

- **True Positives (TP):** $50 + 45 + 40 = 135$
- **True Negatives (TN):**
 - For Dog: $45 + 40 + 7 + 6 = 98$
 - For Cat: $50 + 40 + 3 + 2 = 95$
 - For Mouse: $50 + 45 + 5 + 4 = 104$
- **False Positives (FP):**
 - For Dog: $4 + 2 = 6$
 - For Cat: $5 + 7 = 12$
 - For Mouse: $3 + 6 = 9$
- **False Negatives (FN):**
 - For Dog: $5 + 3 = 8$
 - For Cat: $4 + 6 = 10$
 - For Mouse: $2 + 7 = 9$

Chapter 3

BACKGROUND KNOWLEDGE

3.1 Brain tumors

3.1.1 Definition and location

A brain tumor is defined as an abnormal growth of cells occurring in the brain or its vicinity. These tumors can develop in various locations:

- Within the brain tissue itself
- Near the brain tissue, including:
 - Nerves
 - Pituitary gland
 - Pineal gland
 - Membranes covering the brain's surface

3.1.2 Nature of brain tumors

Brain tumors can be further categorized based on their cellular behavior:

- Noncancerous (Benign) Brain Tumors:
 - May grow over time
 - Can press on brain tissue
 - Generally do not invade surrounding tissue
- Cancerous (Malignant) Brain Tumors:
 - Tend to grow quickly
 - Can invade and destroy brain tissue

3.2 Types of brain tumors

There are many types of brain tumors, but in this project, I mainly focus on three types:

3.2.1 Glioma Brain Tumors

What is Glioma tumor

Glioma is a common type of tumor originating in the brain. About 33 percent of all brain tumors are gliomas, which originate in the glial cells that surround and support neurons in the brain, including astrocytes, oligodendrocytes, and ependymal cells. Gliomas are growths of cells that look like glial cells. The glial cells surround and support nerve cells in the brain tissue. Gliomas can be benign, but most are malignant. Glioblastoma is the most common type of malignant brain tumor.



Figure 3.1: Glioma Brain tumor

Types of Glioma tumor

- **Astrocytoma:** The most common type of glioma. It develops from a type of glial cell called an astrocyte.
- **Oligodendrogioma:** A rare type of glioma. They develop from glial cells called oligodendrocytes.
- **Glioblastoma:** A fast-growing (high-grade) glioma. In the past, it was also called glioblastoma multiforme or GBM.

Symptoms of Glioma tumor

- Headaches
- Seizures
- Personality changes
- Weakness in the arms, face, or legs
- Numbness
- Problems with speech

3.2.2 Meningiomas Brain tumors

What is Meningiomas tumors

A meningioma is a tumor that grows from the membranes that surround the brain and spinal cord, called the meninges. A meningioma is not a brain tumor, but it may press on the nearby brain, nerves and vessels. Most meningiomas grow very slowly. They can grow over many years without causing symptoms. But sometimes, their effects on nearby brain tissue, nerves or vessels may cause serious disability. Meningiomas are usually benign, but sometimes they can be malignant. Meningiomas are the most common type of benign brain tumor.

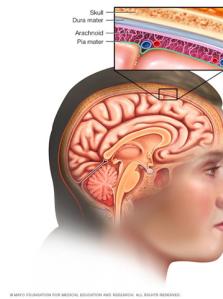


Figure 3.2: Meningiomas Brain tumor

Types of Meningioma tumor

- Convexity meningioma
- Falcine and parasagittal meningioma
- Intraventricular meningioma
- Posterior fossa / petrous meningioma
- Olfactory groove meningioma
- Sphenoid wing meningioma
- Petrous meningioma

Symptoms of Meningioma tumor

- Changes in vision, such as seeing double or blurring
- Headaches that are worse in the morning
- Memory loss
- Loss of smell
- Seizures
- Weakness in the arms or legs
- Problems with speech

3.2.3 Pituitary brain tumors

What is Pituitary tumors

Pituitary tumors are unusual growths that develop in the pituitary gland. This gland is an organ about the size of a pea. It's located behind the nose at the base of the brain. Some of these tumors cause the pituitary gland to make too much of certain hormones that control important body functions. Most tumors that happen in and around the pituitary gland are benign. Pituitary tumors can be treated in several ways. The tumor may be removed with surgery. Or its growth may be controlled with medications or radiation therapy. Sometimes, hormone levels are managed with medicine.

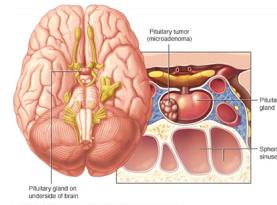


Figure 3.3: Pituitary Brain tumor

Types of Pituitary tumor

- **Nonfunctioning:** These adenomas don't make hormones.
- **Functioning:** These adenomas make hormones. They cause different symptoms depending on the kind of hormones they make:
 - **Adrenocorticotropic hormone:** This hormone also is known as ACTH
 - **Growth hormone:** These tumors are called somatotroph adenomas
 - **Luteinizing hormone and follicle-stimulating hormone:** These hormones also are known as gonadotropins
 - **Prolactin:** These tumors are called prolactinomas or lactotroph adenomas
 - **Thyroid-stimulating hormone:** These tumors are called thyrotroph adenomas.
- **Macroadenomas:** These are larger adenomas. They measure about 1 centimeter or more
- **Microadenomas:** These adenomas are smaller. They measure less than 1 centimeter

Symptoms of Meningioma tumor

Not all pituitary tumors cause symptoms. Sometimes these tumors are found during an imaging test, such as an MRI or a CT scan, that is done for another rea-

son. If they don't cause symptoms, pituitary tumors usually don't need treatment.

Pituitary tumor symptoms may be caused by a tumor putting pressure on the brain or other parts of the body nearby. Symptoms also can be caused by a hormone imbalance. Hormone levels can rise when a pituitary tumor makes too much of one or more hormones. A large tumor that disrupts the way the pituitary gland works may cause hormone levels to fall.

3.3 Magnetic resonance imaging (MRI)

3.3.1 What is MRI

MRI is a medical imaging technique that uses a magnetic field and computer-generated radio waves to create detailed images of the organs and tissues in your body. This non-invasive method allows healthcare professionals to closely examine internal structures such as organs, tissues, and the skeletal system. It produces high-resolution images of the inside of the body that help diagnose a variety of conditions.

MRI is highly effective and widely used for detecting brain tumors because it produces detailed images of brain structures. Additionally, MRI provides exquisite detail of brain, spinal cord and vascular anatomy, and has the advantage of being able to visualize anatomy in all three planes: axial, sagittal and coronal as the figure below.

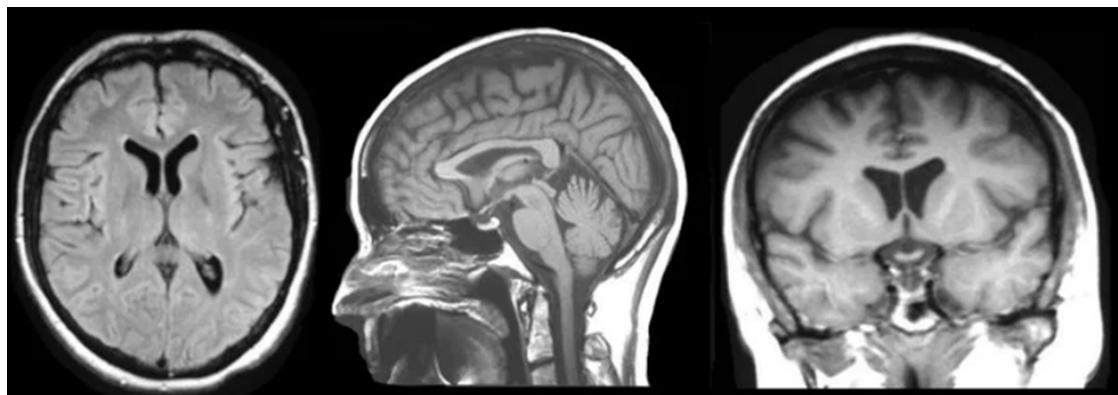


Figure 3.4: MRI sample picture

MRI has an advantage over CT in being able to detect flowing blood and cryptic vascular malformations. It can also detect demyelinating disease, and has no beam-hardening artifacts such as can be seen with CT. Thus, the posterior fossa

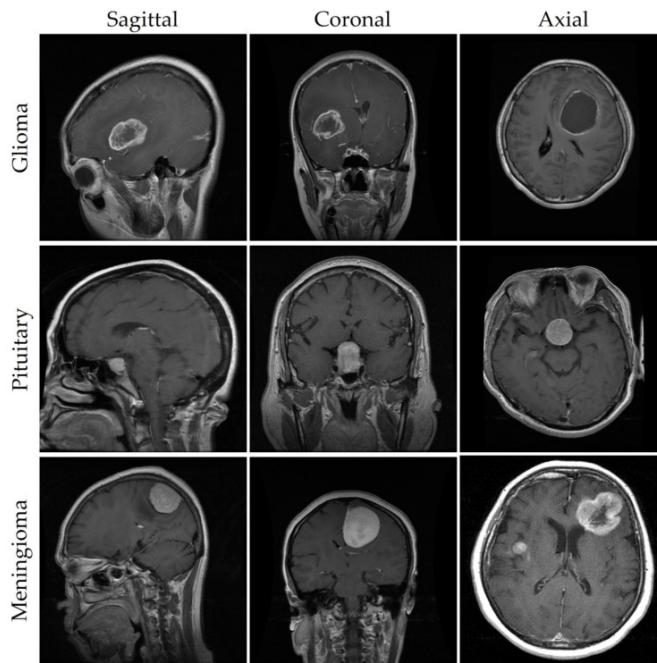


Figure 3.5: Example MRI images of 3 brain tumors types

is more easily visualized on MRI than CT. Imaging is also performed without any ionizing radiation.

3.3.2 Basic principles

MRI uses strong magnetic fields and radio waves to image body tissues. Protons in water molecules are aligned by a magnetic field, then disturbed by radio frequency (RF) pulses. As protons return to alignment, they emit RF signals which are measured and transformed into images. Different pulse sequences produce various image types. Key parameters include Repetition Time (TR) and Time to Echo (TE). Tissue characteristics are determined by T1 (longitudinal) and T2 (transverse) relaxation times, which measure how quickly protons realign with the magnetic field and lose phase coherence, respectively.

3.3.3 MRI imaging sequences

The most common MRI sequences are T1-weighted and T2-weighted scans. T1-weighted images are produced by using short TE and TR times. The contrast and brightness of the image are predominately determined by the T1 properties of tissue. Conversely, T2-weighted images are produced by using longer TE and TR

times. In these images, the contrast and brightness are predominately determined by the T2 properties of tissue.

A third commonly used sequence is the Fluid Attenuated Inversion Recovery (Flair). The Flair sequence is similar to a T2-weighted image except that the TE and TR times are very long.

In general, T1- and T2-weighted images can be easily differentiated by looking at the CSF. CSF is dark on T1-weighted imaging and bright on T2-weighted imaging.

	TR (msec)	TE (msec)
T1-Weighted (Short TR and TE)	500	14
T2-Weighted (Long TR and TE)	4000	900
Flair (Very long TR and TE)	9000	114

Table 3.1: Most common MRI Sequences and their Approximate TR and TE times

3.3.4 Properties of MRI images

MRI images have plenty of properties:

- High Resolution images.
- Multiplanar Imaging: MRI can produce images in multiple planes (axial, sagittal, coronal) without moving the patient.
- Non-Ionizing Radiation: Unlike X-rays and CT scans, MRI does not use ionizing radiation, making it safer for repeated use.
- MRI images demonstrate superior soft-tissue contrast as compared to CT scans and plain radiographs.

3.3.5 Limitation of MRI images

- Relatively long scan times compared to CT or X-ray.
- High cost of equipment and maintenance.
- Limited availability in some areas.
- Not suitable for patients with certain implants or severe claustrophobia.
- Noise: loud noise commonly referred to as clicking and beeping.
- People with implants, particularly those containing iron can't take MRI scans.

Chapter 4

Results and Discussions

4.1 Experiment Setup

4.1.1 Data

To solve the problem of insufficient data volume, I have merged data from two folders, training and testing, each containing 4 labels, into one folder with 4 labels. Then I split the data into three parts: Training set accounts for 80% total of the dataset, Testing set accounts for 10% and Validation for 10%. Then I perform various augmentation as mentioned in *Section 2.1.2*.

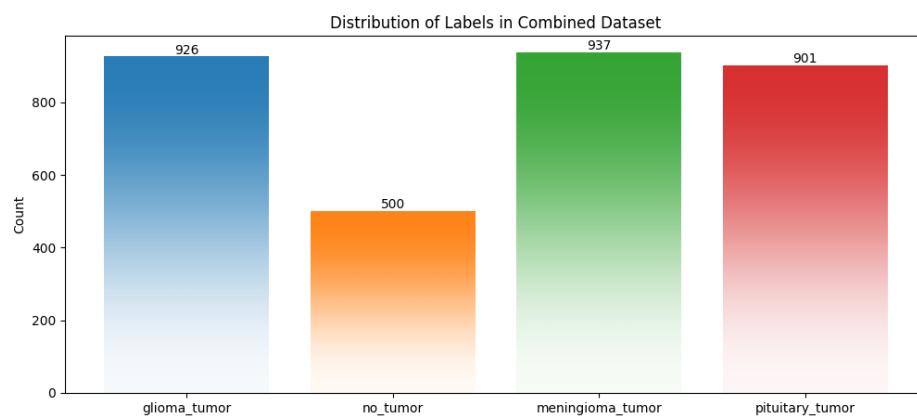


Figure 4.1: Combined Dataset

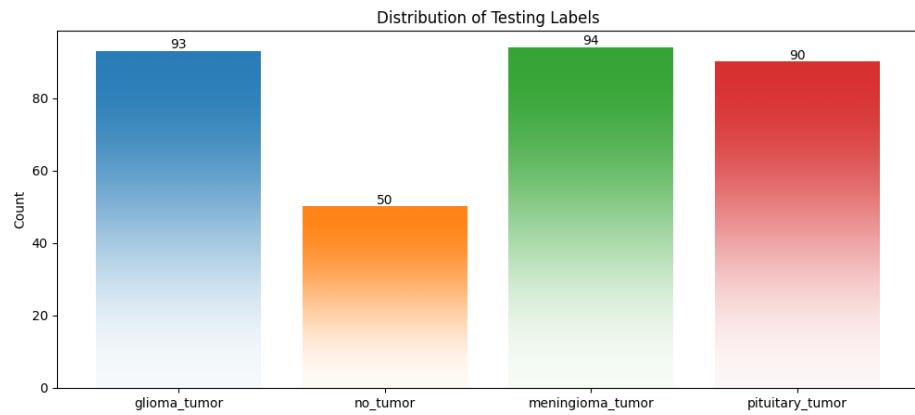


Figure 4.2: Testing Labels After Combined

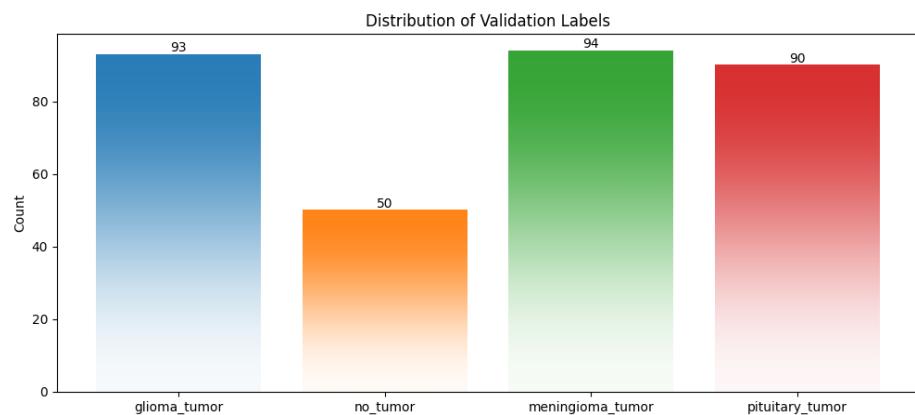


Figure 4.3: Validation Labels After Combined

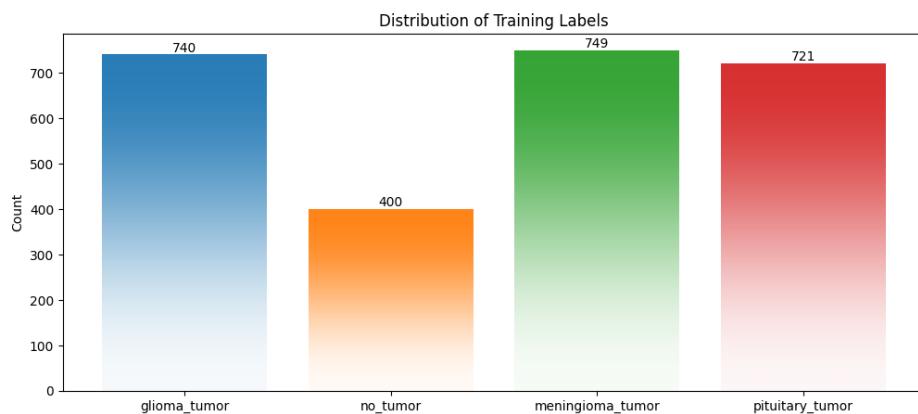


Figure 4.4: Training Labels After Combined

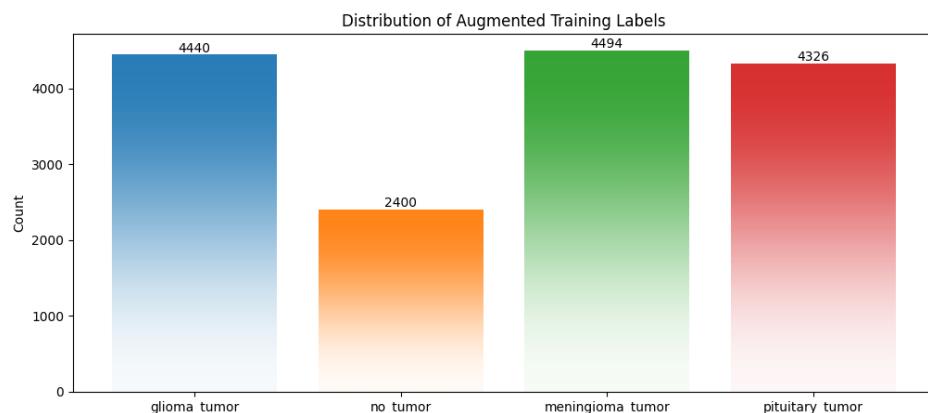


Figure 4.5: Training Labels After Augmented

4.1.2 Framework

I choose Keras as the deep learning framework because it offers a high-level, user-friendly interface for building and training deep learning models. Keras simplifies the process of designing complex neural networks, such as Convolutional Neural Networks (CNNs), which are crucial for image classification tasks like brain tumor detection. Its flexibility allows me to customize models easily, experiment with different architectures, and integrate advanced features like transfer learning. Not only that, Keras contains a group of different pre-trained models on the ImageNet dataset that helps us build the model quickly due to its fast convergence.

4.1.3 Model

I am using the InceptionV3 and ResNet101 models for my brain tumor classification project due to their proven performance in image classification tasks. InceptionV3 is well-known for its ability to handle complex images efficiently by using multiple convolutional filter sizes within the same module, which helps capture different levels of image detail. ResNet101, on the other hand, uses deep residual learning, which allows for extremely deep architectures without the risk of vanishing gradients, improving model accuracy. The parameter to set for the 2 models:

ResNet101

- **Batch Size:** Set to 32 for both training and validation. The number of iterations per epoch is the number of training samples divided by the batch size.
- **Model Architecture:** The top layers of the ResNet101 model (fully connected layers) are removed, retaining the convolutional base.
- **Dropout:** A dropout rate of 0.5 is applied to prevent overfitting.
- **Retraining:** The entire model, including the pre-trained layers, is retrained on the brain tumor dataset.
- **Optimizer:** The Adam optimizer is used with a starting learning rate of 0.001.
- **Activation Function:** Softmax is used as the activation function in the output layer for multi-class classification.
- **Loss Function:** Categorical cross-entropy is used for multi-class classification, with metrics set to categorical accuracy.
- **Model Checkpoint:** The ModelCheckpoint callback saves the model's weights when the validation accuracy is highest.
- **Learning Rate Reduction:** The ReduceLROnPlateau callback reduces the learning rate by a factor of 0.3 after 2 epochs if the validation accuracy stops improving, with a minimum delta of 0.001.
- **Training Duration:** The model is trained for 15 epochs.

InceptionV3

- **Batch Size:** Set to 32 for training. The number of iterations per epoch is determined by dividing the number of training samples by the batch size.
- **Model Architecture:** InceptionV3 is used as the base model with the top layers removed, keeping the convolutional layers. A `GlobalAveragePooling2D` layer is added, followed by a `Dropout` layer with a rate of 0.5 to reduce overfitting, and a final dense layer with a softmax activation for multi-class classification.
- **Dropout:** A dropout rate of 0.5 is applied after the pooling layer to prevent overfitting.
- **Retraining:** The entire model, including the pre-trained InceptionV3 layers, is retrained on the brain tumor dataset.
- **Optimizer:** Adam optimizer is used with default settings (learning rate of 0.001).
- **Activation Function:** Softmax is applied in the output layer to perform multi-class classification, since there are four classes (glioma tumor, no tumor, meningioma tumor, and pituitary tumor).
- **Loss Function:** Categorical cross-entropy is used for multi-class classification, with accuracy as the metric for evaluation.
- **Model Checkpoint:** The `ModelCheckpoint` callback is used to save the best-performing model based on validation accuracy, ensuring that only the best model is stored.
- **Learning Rate Reduction:** The `ReduceLROnPlateau` callback decreases the learning rate by a factor of 0.3 if the validation accuracy does not improve after 2 epochs, with a minimum change (delta) of 0.001 required for improvement detection.
- **Early Stopping:** The `EarlyStopping` callback stops training if the validation accuracy does not improve for 5 consecutive epochs, and it restores the best weights from the training process.
- **Training Duration:** The model is trained for a total of 15 epochs using data augmentation to enhance the dataset variability.

Custom CNN

- **Batch Size:** Set to 32 for both training and validation. The number of iterations per epoch is determined by the number of training samples divided by the batch size.
- **Epochs:** The model is trained for 25 epochs.
- **Model Architecture:**
 - **Conv2D Layers:** Three convolutional layers with 32, 64, and 64 filters, respectively, each with a kernel size of (3, 3) and ReLU activation.
 - **MaxPooling2D Layers:** Three max-pooling layers with pool size (2, 2).
 - **Flatten Layer:** Flattens the 3D output from the convolutional layers to 1D.
 - **Dense Layers:** A fully connected layer with 64 units and ReLU activation, followed by a dropout layer with a dropout rate of 0.5, and an output dense layer with softmax activation.
- **Optimizer:** Adam optimizer with a learning rate of 0.001.
- **Activation Function:** Softmax is used as the activation function in the output layer for multi-class classification.
- **Loss Function:** Categorical cross-entropy is used for multi-class classification, with metrics set to categorical accuracy.

4.2 Result

4.2.1 ResNet101

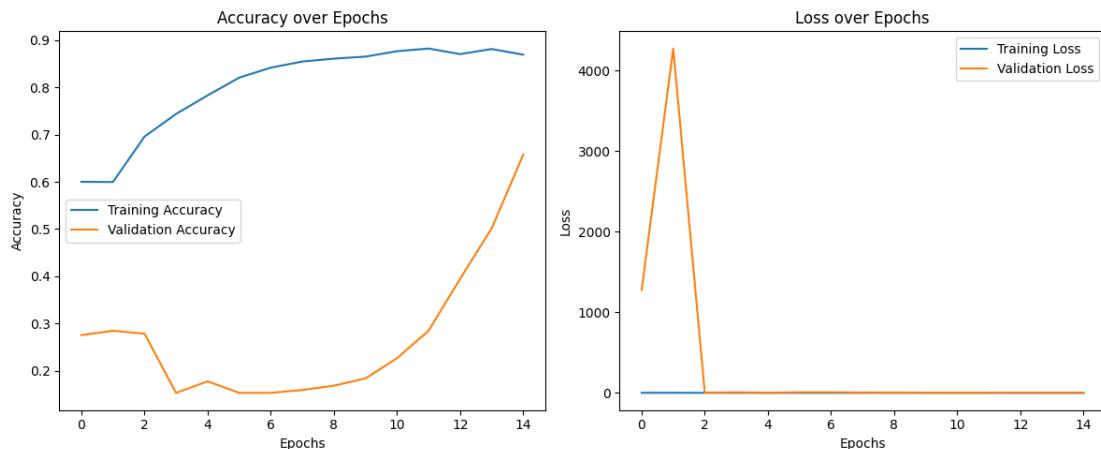


Figure 4.6: Training - Validation Accuracy and Loss of ResNet101

Model performance Looking at the plotting curves and the model's performance, we are encountering a clear issue of overfitting. The overfitting in the InceptionV3 model is evident from the significant gap between training and validation accuracies and losses. Despite using Dropout layers to reduce complexity during training, the discrepancy between training and validation arises due to Dropout being inactive during validation, leading to better performance in validation predictions. Additionally, the training loss is computed after each batch, while validation loss is calculated at the end of each epoch, which can cause validation loss to be higher initially. This model was trained using ResNet101, a complex architecture with 101 layers. Given the size and complexity of the dataset, the model might be prone to overfitting due to its high capacity, despite the use of data augmentation and regularization strategies. This is reflected in the curve patterns, where training accuracy continues to improve steadily, but validation accuracy diverges, indicating poor generalization.

Result After testing the model on the testing dataset, the outcome was not as expected because of the overfitting problem. The model *Accuracy* is 65.44%, furthermore, the *Precision*, *Sensitivity*, *F1 score* is 81.84%, 65.44%, 65.92%. Looking at the result, we can see the Sensitivity and F1 score are way too low, and this is unacceptable. The evaluation result is shown as follows:

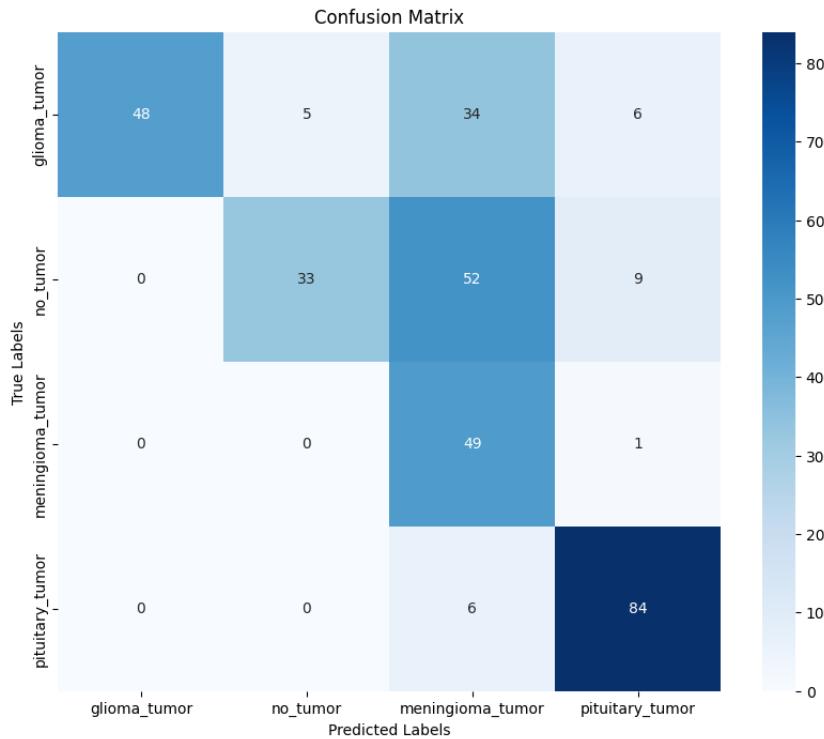


Figure 4.7: Confusion matrix of ResNet101

Label	Accuracy	Precision	Sensitivity (Recall)	Specificity	F1 Score	AUC-ROC
glioma-tumor	0.654434	1.000000	0.516129	1.000000	0.680851	0.950418
no-tumor	0.654434	0.868421	0.351064	0.978541	0.500000	0.894211
meningioma-tumor	0.654434	0.347518	0.980000	0.667870	0.513089	0.968159
pituitary-tumor	0.654434	0.840000	0.933333	0.932489	0.884211	0.977356
Average	0.654434	0.763985	0.695132	0.894725	0.644538	0.947536

Table 4.1: Classification Metrics of ResNet101

Based on these consequences, even though some metrics are significant like *Specificity* and *AUC-ROC* are quite significant, other metrics remain not impressive. For instance, the model struggles particularly with the prediction of certain classes, such as glioma, where Precision is at 1.0, but Recall is only 0.516, indicating that while the model is highly confident in its predictions for glioma tumors, it misses a significant portion of true positive cases. Meningioma shows the opposite issue, with high Recall (0.98) but low Precision (0.35), indicating many false positives. This

imbalance lowers the F1-score (0.51) for meningioma. This imbalance negatively impacts the F1-score (0.51), reflecting a significant trade-off between Precision and Recall for this class. These discrepancies likely stem from the limited sample size in some classes, leading to poor generalization. Variability in tumor shapes and colors, particularly for meningioma, adds further complexity.

ROC Curves The ResNet101 shows good overall performance, with AUC values ranging from 0.8942 to 0.9774. It excels at identifying pituitary and meningioma tumors but shows lower performance for the "no tumor" class. This variability suggests the model learned features more distinctive for certain tumor types. The spread between best and worst performing classes is notable, indicating potential for improvement in consistency across tumor types. This suggests that the model may have learned features that are more distinctive for certain tumor types (like pituitary and meningioma) than for others. The lower performance on the "no tumor" class could potentially lead to false positives in a clinical setting, indicating a need for careful threshold setting or additional validation steps.

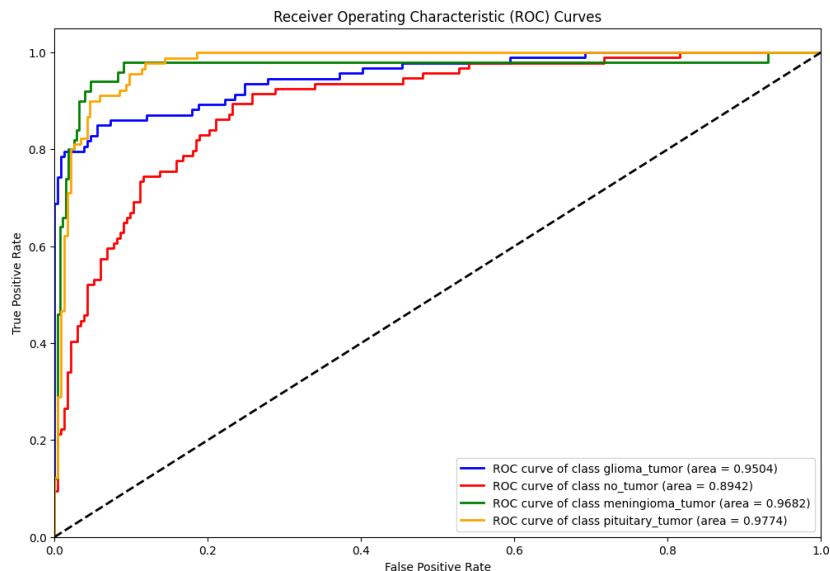


Figure 4.8: ResNet101 ROC Curves

4.2.2 InceptionV3

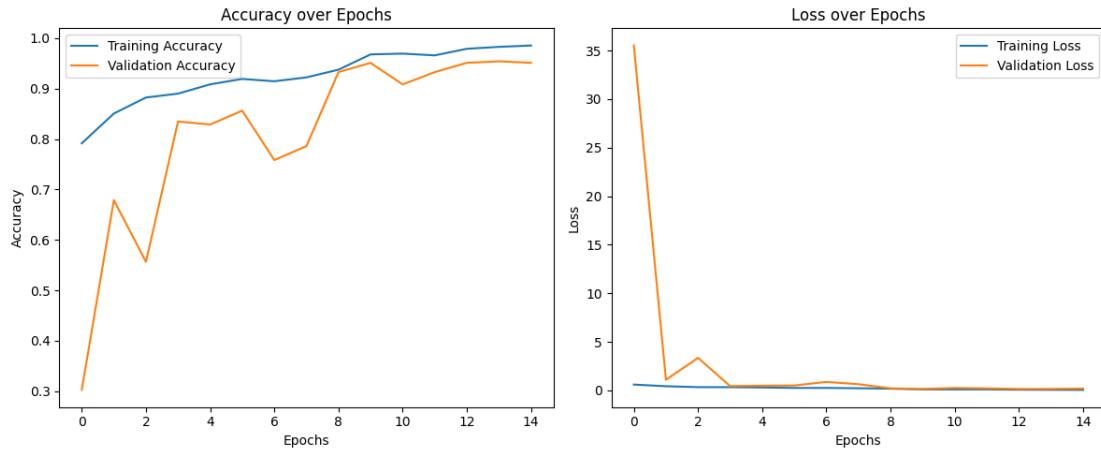


Figure 4.9: Training - Validation Accuracy and Loss of InceptionV3

Model performance Looking at the accuracy and loss curves, the model shows strong performance early in the training process. Both training and validation accuracy improve quickly and converge around 90-95% by the 10th epoch. The gap between training and validation accuracy remains small, indicating that the model is generalizing well and there is no immediate sign of overfitting. However, in the loss plot, the validation loss starts very high in the first epoch, dropping dramatically afterward, and stabilizes at a low value from epoch 2 onward. This initial spike in validation loss could be due to how the model is learning and adjusting early on. After this, both training and validation losses stabilize close to zero, further indicating that the model fits the data well without much variance between the training and validation sets.

Result The model InceptionV3 achieves high *Accuracy* up to 97.25% and the other metric like *Precision*, *Sensitivity*, *F1 score* remain high results about 97.29%, 97.25% and 97.23%. Compared to the model ResNet101, these results show noticeable improvement, highlighting InceptionV3's enhanced performance in brain tumor classification.

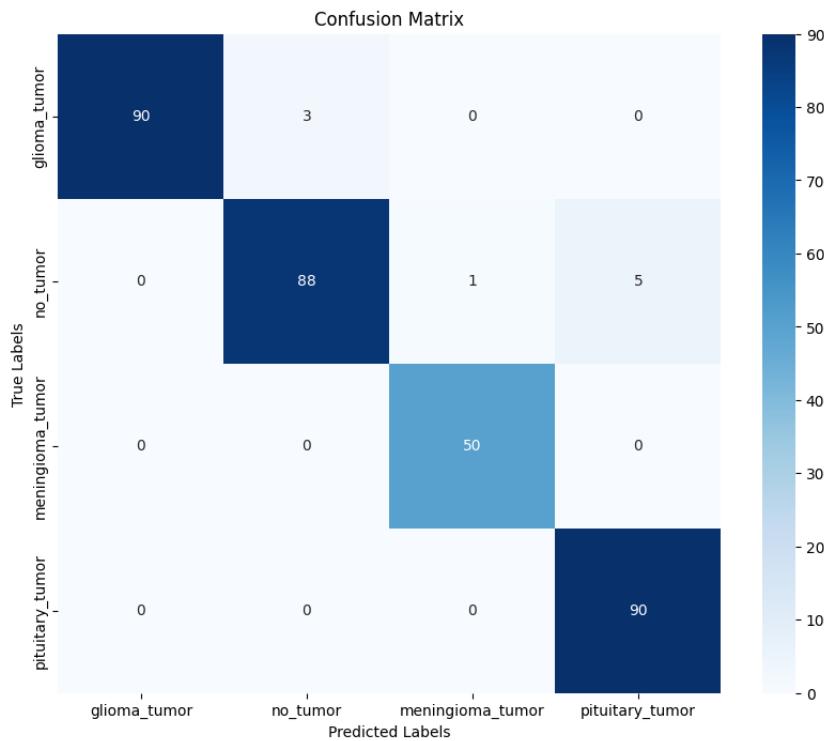


Figure 4.10: Confusion matrix of InceptionV3

Label	Accuracy	Precision	Sensitivity (Recall)	Specificity	F1 Score	AUC-ROC
	0.9725	0.9900	0.9200	0.9894	0.9600	0.9984
glioma_tumor	0.9725	0.9500	0.9800	0.9700	0.9600	0.9993
no_tumor	0.9725	0.9800	1.0000	0.9650	0.9900	0.9996
meningioma_tumor	0.9725	0.9800	1.0000	0.9600	0.9900	1.0000
pituitary_tumor	0.9725	0.9750	0.9750	0.9711	0.9750	0.9993
Average	0.9725	0.9750	0.9750	0.9711	0.9750	0.9993

Table 4.2: Performance metrics of InceptionV3

The model shows strong overall performance, with high specificity and AUC-ROC scores across all classes. However, there are some areas for improvement in precision and recall. The "glioma tumor" class has high precision (0.99) but slightly lower recall (0.92), indicating some missed true positives. For the "no tumor" class, precision is strong (0.95) but recall is a bit lower (0.98), suggesting occasional oversight of true negatives. The "meningioma tumor" class has perfect recall (1.00) but slightly lower precision (0.98), indicating some misclassification of other tumors.

The "pituitary tumor" class performs very well with perfect recall (1.00) and high F1 score (0.99). Overall, the model is effective but could benefit from balancing precision and recall more evenly across classes.

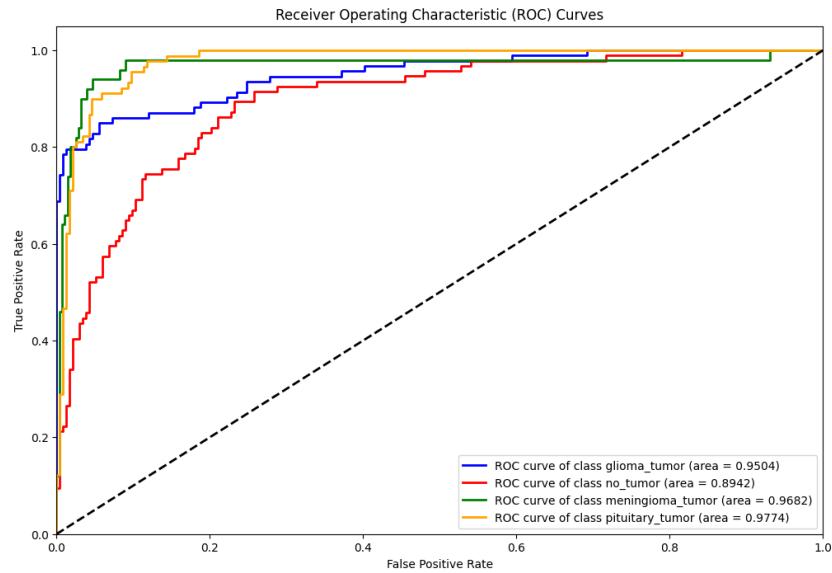


Figure 4.11: InceptionV3 ROC Curves

ROC Curves InceptionV3 achieves near-perfect classification across all tumor types, with AUC values from 0.9975 to 1.0000. The ROC curves are almost indistinguishable, showing exceptional consistency across classes. This suggests the model learned features equally effective for all tumor types, including previously challenging classifications. The near-vertical rise of all curves indicates minimal false positives, crucial for clinical reliability. However, such high-performance warrants extensive validation to ensure generalizability.

4.2.3 Custom CNN

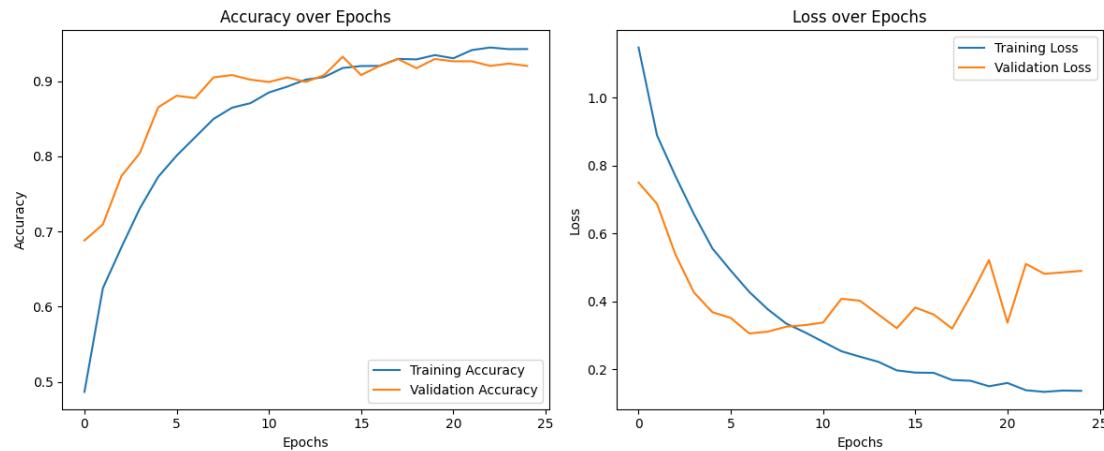


Figure 4.12: Training - Validation Accuracy and Loss of Custom CNN

Model performance Looking at the plotting curves and the model's performance, can easily see there are no immediate signs of severe overfitting in this case. In the accuracy plot, training and validation accuracies increase steadily and remain close, indicating good generalization. However, in the loss plot, slight fluctuations in validation loss after epoch 15 suggest minor overfitting or instability, while training loss continues to decrease. Overall, the model performs well, but further regularization or early stopping could help stabilize the validation loss in later epochs.

The model used appears to have a good learning curve up until the later epochs, where it might benefit from early stopping or further regularization techniques (e.g., increased Dropout, L2 regularization) to improve stability. This custom CNN for brain tumor classification shows promising results but might need fine-tuning to handle potential minor overfitting in the later stages of training.

Result After testing the model on the testing dataset, the model achieves *Accuracy* is 91.93%, *Precision*, *Recall* and *F1 score* by roster are 91.39%, 91.13% and 91.03%, respectively. These results demonstrate strong and balanced performance across the evaluation metrics, indicating the model's effectiveness in brain tumor classification.

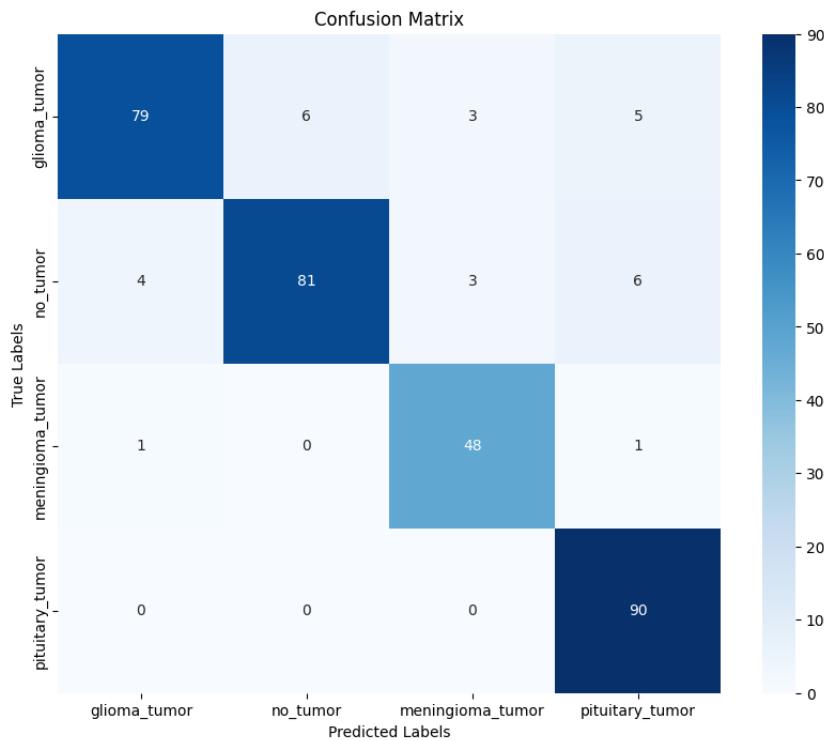


Figure 4.13: Confusion matrix of Custom CNN

Label	Accuracy	Precision	Sensitivity (Recall)	Specificity	F1 Score	AUC-ROC
glioma-tumor	0.911315	0.940476	0.849462	0.978632	0.892655	0.966823
no-tumor	0.911315	0.931034	0.861702	0.974249	0.895028	0.982330
meningioma-tumor	0.911315	0.888889	0.960000	0.978339	0.923077	0.990036
pituitary-tumor	0.911315	0.882353	1.000000	0.949367	0.937500	0.995874
Average	0.911315	0.910688	0.917791	0.970147	0.912065	0.983766

Table 4.3: Classification Metrics of Custom CNN

Based on these results, the model demonstrates strong performance overall, particularly in metrics like specificity and AUC-ROC, which are consistently high across all classes. However, some challenges remain in achieving balanced precision and recall for specific classes. For instance, the "glioma tumor" class shows a reasonable precision (0.94), but recall is lower at 0.85, indicating that the model correctly identifies most glioma cases but still misses a portion of true positives.

Similarly, for the "no tumor" class, precision (0.93) is strong, but recall is

slightly lower at 0.86, suggesting that while the model is confident in its predictions, it occasionally overlooks true negative cases. The "meningioma tumor" class performs well in recall (0.96), but precision (0.88) is slightly lower, indicating some misclassification of other tumor types as meningioma. The "pituitary tumor" class achieves the highest recall (1.0) and a strong F1 score (0.94), showing that the model handles this class effectively.

ROC Curves The custom CNN demonstrates significant improvement over ResNet101, with AUC values from 0.9668 to 0.9959. Performance is more consistent across classes, with a notable improvement in the "no tumor" classification. The curves show steeper initial rises, indicating high true positive rates at low false positive rates. The overall shape of the ROC curves for the custom CNN shows a very steep initial rise across all classes. This indicates that the model achieves high true positive rates with very low false positive rates, a crucial characteristic for clinical applications. The improved and more consistent performance across classes suggests that the custom CNN has learned a more generalized set of features that work well for all tumor types, making it a more reliable model overall.

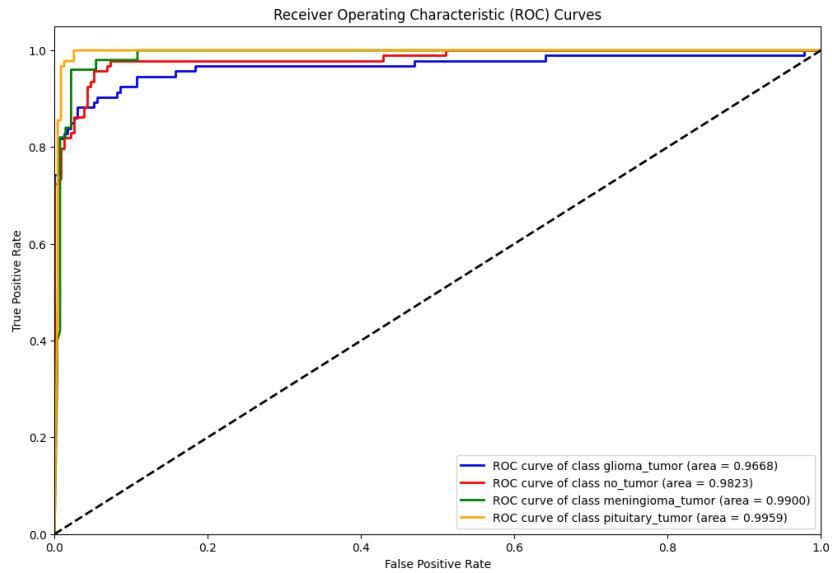


Figure 4.14: Custom CNN ROC Curves

4.3 Discussion

Metric	ResNet101	InceptionV3	Custom CNN
Accuracy	0.6544	0.9725	0.9113
Precision	0.7640	0.9750	0.9107
Recall/Sensitivity	0.6951	0.9750	0.9178
Specificity	0.8947	0.9711	0.9701
F1 Score	0.6445	0.9750	0.9121
AUC-ROC	0.9475	0.9993	0.9838

Table 4.4: Comparison of Classification Metrics

InceptionV3 achieves the highest performance across all metrics due to its rich feature extraction capability, thanks to being pre-trained on a large dataset (ImageNet). The transfer learning approach allows InceptionV3 to leverage learned features, enabling superior generalization even when fine-tuned for brain tumor classification. ResNet101, despite being pre-trained, underperforms compared to InceptionV3 and even Custom CNN, likely due to overfitting or architectural complexity.

Despite being built from scratch, Custom CNN performs impressively well with an *Accuracy* of 91.13%, *Precision* of 91.07%, and *AUC-ROC* of 0.9838. While it doesn't outperform the pre-trained InceptionV3, the fact that a custom architecture is able to match and exceed ResNet101 in key metrics such as Recall (0.9178) and F1 Score (0.9121) is noticeable. The Custom CNN demonstrates that with careful design and tuning, a model built from scratch can yield competitive results, especially when computational resources or dataset-specific adaptations are prioritized. It also highlights the potential to fine-tune a model for more specific feature extraction compared to a generalized pre-trained model.

The results clearly show that InceptionV3 is the most effective model for brain tumor classification, achieving high scores across all metrics. The Custom CNN offers a reasonable trade-off, performing well, but it does not reach the same level of accuracy or precision as InceptionV3. ResNet101, while still functional, is the least effective model and may not be suitable for high-stakes tasks like medical image classification, where precision and recall are critical.

In practical terms, InceptionV3 would be the best choice for deployment in a real-world setting due to its superior performance. However, if computational resources are limited, the Custom CNN could be a viable alternative, offering solid performance at potentially lower computational cost.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this project, we explored brain tumor classification using three different models: ResNet101, InceptionV3, and a Custom CNN. We addressed the challenges of medical image classification and applied techniques such as transfer learning to improve accuracy. The models were developed in Python using the Keras package with TensorFlow as the backend. Specifically, ResNet101 and InceptionV3, both pre-trained on large-scale image datasets, were fine-tuned to adapt to the specific domain of brain MRI scans. The custom CNN, on the other hand, was designed and trained from scratch to learn the features specific to brain tumor classification.

After the process of training and testing with 3 models, InceptionV3 is the most effective model with the highest *Accuracy* of 97.25%. It outperformed both ResNet101 and the custom CNN model, delivering superior accuracy and performance across several evaluation metrics. This highlights the strength of pre-trained models in leveraging prior knowledge for more accurate brain tumor classification.

5.2 Future work

In the future, I would like to use other pre-trained models to determine which model can further improve the accuracy and performance of brain tumor classification. Additionally, I plan to explore more advanced techniques like ensemble learning and model stacking to combine the strengths of multiple models. Moreover, as additional data is collected from various sources, the results are expected to improve, facilitating the development of an application with enhanced capabilities.

Bibliography

- [1] Global Cancer Observatory. Brain and central nervous system cancer fact sheet, 2022.
- [2] Sundeep Deorah, Charles F Lynch, Zita A Sibenaller, and Timothy C Ryken. Trends in brain cancer incidence and survival in the united states: Surveillance, epidemiology, and end results program, 1973 to 2001. *Neurosurgical focus*, 20(4):E1, 2006.
- [3] Mahmoud Khaled Abd-Ellah, Ali Ismail Awad, Ashraf AM Khalaf, and Hesham FA Hamed. A review on brain tumor diagnosis from mri images: Practical implications, key achievements, and lessons learned. *Magnetic resonance imaging*, 61:300–318, 2019.
- [4] Justin Ker, Lipo Wang, Jai Rao, and Tchoyoson Lim. Deep learning applications in medical image analysis. *Ieee Access*, 6:9375–9389, 2017.
- [5] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.
- [6] Chuen-Kai Shie, Chung-Hisang Chuang, Chun-Nan Chou, Meng-Hsi Wu, and Edward Y Chang. Transfer representation learning for medical image analysis. In *2015 37th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 711–714. IEEE, 2015.
- [7] Brain tumor dataset. [/kaggle/input/brain-tumor-classification-mri](https://www.kaggle.com/c/brain-tumor-classification-mri).
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.
- [9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dmitry Erhan, Sergey Ioffe, Christian Szegedy, and Alex Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. IEEE, 2015.