A black background with white text

Description automatically generated with low confidence

MSc Data Science Project

7PAM2002-0509-2023

Department of Physics, Astronomy and Mathematics

**Data Science FINAL PROJECT REPORT**

Brain Tumor Prediction using MRI Images and Machine Learning.

**Student Name and SRN:**

Sana Ikhlaq: 22075278

Supervisor: Vandana Das

Date Submitted: 28-08-2024

Word Count: 5857

**DECLARATION STATEMENT**

This report is presented in partial satisfaction of the Master of Science in Data Science degree requirements at the University of Hertfordshire.

I have read the student guidance on academic integrity, illegal behavior, and copyright violation details at [Assessment and Educational Misconduct](https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct.pdf) and understand the University's process for dealing with suspected instances of academic misconduct, as well as the potential penalties, which include losing the project module or course.

I declare that the work presented is my own, as well as that any material drawn or quoted from the published or unreleased work of others has been properly recognized. (See UPR AS/C/6.1, Section 7 and the UPR AS/C/5, Section 3.6).

I did not conduct a poll or employ human participants for my MSc project.

I now allow the report to be published on module websites, as long as the source is mentioned.

Student Name printed: Sana Ikhlaq

Student Name Signature: Sana

Student SRN number: 22075278

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

**Acknowledgment**

As I am near my post-graduate studies, I reflect on this journey with immense gratitude for the invaluable learning experiences and the unwavering support I have received.

First and foremost, I am deeply thankful to Almighty God for His constant guidance and endless blessings and for instilling the confidence and courage to pursue this endeavor with determination and self-belief.

I want to thank my Supervisor, **Vandana Das**, whose continuous guidance and support have completed this project. Her patience and willingness to nurture my curiosity have been genuinely inspiring.

I am also grateful to all the professors at the University of Hertfordshire, whose expertise and instruction have profoundly enriched my understanding and knowledge throughout this course.

Lastly, I sincerely thank my parents, sister, and friends for their unwavering encouragement and support. Their belief in me has been a source of strength, and I could not have reached this milestone without them.

**Abstract**

Brain tumor classification is one of the critical objectives of medical imaging when an MRI is performed. MRI is non-invasive and provides high-resolution images. Accurate classification of tumors will help in effective diagnosis and treatment, directly affecting a patient's prognosis. In this study, we discussed the classification of brain tumors using deep learning(DL) models on MRI images and improved diagnosis accuracy, hence creating reliable decision-making tools in clinical settings. Mainly, it is guided by the research question of which DL models like VGG16, VGG19, InceptionV3, or a self-built CNN best classify the tumor. The methodology involved preprocessing a dataset of MRI images and applying models against them to classify tumors under four categories: glioma, meningioma, pituitary, and non-tumor. VGG19 performed best with a precision of 0.94, accuracy of 0.94, recall of 0.94, and an F1-score of 0.94. This paper concludes that VGG19 and VGG16 are the most efficient models for the classification of brain tumors using MRI images. VG19 is the model used in clinical settings to diagnose the tumor type. Future work will focus on enhancing these models further and looking into other DL architectures to achieve higher classification accuracy and overcome the limitations observed in this study.

**Keywords:** Brain tumor classification, VGG19, CNN, MRI Images

Contents

[1 Introduction 1](#_Toc175750480)

[1.1 Background of the Study 1](#_Toc175750481)

[1.2 Problem Statement 2](#_Toc175750482)

[1.3 Significance of the Study 3](#_Toc175750483)

[1.4 Scope of Study 3](#_Toc175750484)

[2 Literature Review 4](#_Toc175750485)

[3 Research Methodology 6](#_Toc175750486)

[3.1 Proposed Methodology 6](#_Toc175750487)

[3.2 Data Collection 6](#_Toc175750488)

[3.3 Data Analysis 6](#_Toc175750489)

[3.4 Ethical Issues 8](#_Toc175750490)

[3.5 Data Preprocessing 8](#_Toc175750491)

[3.6 Feature Engineering 9](#_Toc175750492)

[3.7 Model Selection 9](#_Toc175750493)

[3.7.1 VGG16 and VGG19 9](#_Toc175750494)

[3.7.2 InceptionV3 10](#_Toc175750495)

[3.7.3 Custom CNN 10](#_Toc175750496)

[3.8 Model Implementation 11](#_Toc175750497)

[3.9 Training and Evaluation 11](#_Toc175750498)

[3.10 Performance Metrics 11](#_Toc175750499)

[3.10.1 Precision 12](#_Toc175750500)

[3.10.2 Recall 12](#_Toc175750501)

[3.10.3 F1-Score 12](#_Toc175750502)

[3.11 Experimental Setup 13](#_Toc175750503)

[4 Results 14](#_Toc175750504)

[4.1 Class-wise results of Applied DL model 14](#_Toc175750505)

[4.2 Overall Results of Applied DL Models 15](#_Toc175750506)

[4.3 Confusion Matrix and History Graph 16](#_Toc175750507)

[5 Analysis and Discussion 19](#_Toc175750508)

[5.1 Interpreting the Result 19](#_Toc175750509)

[5.2 Comparative Model Performance 19](#_Toc175750510)

[5.3 Comparison with Existing Literature 19](#_Toc175750511)

[5.4 Study Limitations 19](#_Toc175750512)

[5.5 Relation to Project Objectives 20](#_Toc175750513)

[5.6 Practical Applicability of Models 20](#_Toc175750514)

[5.7 Future Work 20](#_Toc175750515)

[6 Conclusion 21](#_Toc175750516)

[7 Reference 22](#_Toc175750517)

[8 Appendices 23](#_Toc175750518)

# Introduction

## Background of the Study

Most of the patients suffering from brain tumor conditions die within five years after the detection of their tumors (Rosenfeld & Massey, 2019). Brain tumors are sometimes referred to as intracranial neoplasms, abnormal growths in the cranial cavity that originate from various types of cells like neurons, glia or support cells, blood vessels, lepto membranes, and even ectopic tissues. On the other hand, brain metastases are those tumors in the brain that have developed from cancer in different organs, like lung cancer or breast cancer. Brain tumors could either be benign or malignant, meaning non-life threatening or life-threatening, respectively. While benign brain tumors may exist forever without creating any problem for their owner other than psychological discomfort, malignant types still pose a severe risk to survival due to the aggressive nature in which they spread.

Brain tumors is most severe and deadly conditions that affect our central nervous systems alongside other system functions. These tumors have been known to have different behaviors that range from fast-growing and fatal ones to slow-growing and non-harmful ones, respectively. The classification of the brain neoplasms is crucial in deciding what treatment course is required, including surgery, irradiation, cytotoxic agents, combined chemotherapy and radiotherapy regimens, etc. The doctors specializing in this condition must make early diagnoses for the patients, as the type of the tumor must influence its treatment and prognosis.

Precise magnetic resonance imaging, is the latest non-invasive technique, helps guide the diagnosis of a brain tumor by painting a picture that lucidly depicts the degree of damage caused by such conditions (Işin et al., 2016; Wadhwa et al., 2019). Images produced by MRI give relevant information to different parts of the brain, enabling doctors to see something abnormal, for example, the existence of tumors in those areas, just before surgery. The MRI includes information on their location and size. However, at times, it can get quite challenging to interpret these scans manually due to the complexity involved when one looks at the brain structures together with subtle differences that exist between various tumor types, leading to the need for the development of automated systems aimed at achieving precise classifications of human brain cancer cells.

Deep learning refers to the part of artificial intelligence concerned with constructing and training models to perform particular tasks, such as visual or speech recognition (Singhal et al., 2017). Thus, the latest development in AI was beneficial for automating brain tumor identification processes on MRI. When experts see humanity and AI together, it seems that all of them strive for excellence, but their methods differ considerably. I will examine some critical machine learning papers on brain tumor imaging by assessing DL models used in classification.

The current study applies DL models to categorize brain tumors using MRI images. It points to the efficiency of several existing DL architectures, including VGG19, CNN, Inception, and VGG16, when applied to identify brain tumors into four different classes such as glioma, meningioma, pituitary tumors, and no presence of a tumor. Different performance benchmarking on these models will be done to improve the reliability and accuracy of the tools developed in the future for the diagnosis of brain tumors and, hence, clinical outcomes for patients.

## Problem Statement

The objective of classification brain tumors from MRI images has been one of the most paramount challenges of medical diagnosis, where current traditional ways are often slow, subjective, and error-prone. Despite outstanding progress in deep learning, there remains a considerable gap in knowledge of the comparative effectiveness of different models within this domain. The available diagnostic methods, however, have not been very effective in the differentiation of the tumor types, which, in the end, could cause the making of false diagnoses and less-than-ideal treatment outcomes. The purpose of study is to indicate the way forward in improving the reliability and validity of brain tumor classification through an assessment of DL terminologies such as VGG19, CNN, Inception, and VGG16 to suggest the most productive way to achieve an improvement in diagnostic accuracy and quality in patient care.

**1.3 Study's Aims and Objectives**

The primary aim of the thesis is to construct a model based on DL for precise stem classification of brain tum using MRI images. Furthermore, this research should focus on the following objectives:

1. Collecting and preprocessing MRI images associated with brain tumors to guarantee that high-quality input data will be available for model training.

3. Recognize a brain tumor in an MRI image through training and testing various models such as CNNs, VGG19, Inception, and VGG16.

4. Compare all models concerning accuracy, precision, recall, and F1 score to determine which is most reliable regarding having an accurate classification of brain tumors.

5. Proposed a generalized DL model that can classify the brain tumor with high accuracy

**1.4 Research Questions**

The main research questions of this study are:

1. Will MRI images be classified as brain tumors using DL models?

2. How correct and reliable will brain tumor categorization from MRI images become after using the DL models?

3. How accurate, precise, and reliable are these DL models like VGG19, CNN, Inception, and VGG16 detecting brain tumors by their recall, accuracy, precision, and F1-score?

4. Which of the proposed deep models has the highest performance and thus could be recommended for accurate and robust classification of brain tumors?

## Significance of the Study

The importance of the present study is that DL models can be successfully used to advance the field of medical, in particular, the diagnosis of brain tumors by accurate and efficient classification using MRI images. Brain tumors, if detected early and classified accurately, would significantly help enhance treatment planning, patient outcomes, and overall survival rates. By comparing different DL models, this study will add to the present body of academic knowledge regarding model capabilities and provide practical insights for clinicians and radiologists in deciding which one is the best tool for diagnosing brain tumors. Such findings may result in a more reliable diagnosis, hence decreasing the misdiagnosis rate so that intervention for patients diagnosed with brain tumors is timely, with the avoidance of ineffective or harmful treatments.

## Scope of Study

It focuses on classifying brain tumors by DL models applied to MRI images. In particular, this study will focus on the collection, preprocessing, and feature engineering of MRI datasets and then use and evaluate four DL models such as VGG19, CNN, Inception, and VGG16. The study focuses on classifying four brain tumor types, including glioma, meningioma, no tumor, and pituitary tumors. It does not cover the other imaging modalities, such as CT scans, and neither has it talked about any other types of brain abnormalities and tumors except for the ones in the four groups. Clinical application of the models or the real-time implementation was also not part of the study; the comparison was only made in the dataset environment.

**1.7 Thesis Structure**

The thesis is divided into sections, representing the research timeline. Section 1 presents the background of the study and states the problem, objectives, research questions, significance, scope, and overview of the thesis structure. Literature Review Chapter 2 reviews the literature on brain tumor classification using DL models, showing the gaps this work is trying to fill. Specifically on the methodology, Chapter 3 elaborates on the research design, which further explains processes involved in data collection, preprocessing, feature engineering, and implementation of DL models. Chapter 4 presents the experimental results obtained from applying VGG19, CNN, Inception, and VGG16 models to the MRI dataset, majorly tending to metrics such as accuracy, F1 scores , recall, and precision. Chapter four also discusses Future Work, summarizing key research findings, contributions, and future investigation areas.

# Literature Review

It is a challenge to diagnose brain tumor types with common human mistakes accurately, and it often takes time, as stated in this analysis(Abiwinanda et al., 2019). An attempt to identify the three main types of brain tumors, Glioma Pituitary and Meningioma, is what this research emphasizes using CNN. It was structured in architecture with one convolution layer. The max pooling layer flattens out before entering a fully connected hidden layer. The training utilized publicly available 3064 T-1 weighted CE-MRI images from Figshare. Despite its simplicity, its achieved training accuracy stood at 98.51%, while that for validation was 84.19%. These results compete with region-based segmentation algorithms that are more complex and report accuracies ranging between 71.39% and 94.68% for the same dataset.

At advanced stages, brain tumors are the most violent diseases, hence making treatment planning essential to help improve patient's quality of life because they significantly reduce their lifespan. CIMGH|UScle1 is commonly utilized to evaluate different types of tumors within distinct organs. This kind of research (Seetha & Selvakumar Raja, 2018)centered on detecting brain tumors via MRI images, after which numerous issues emanate from the gigantic amounts of data realized through MRI scans.

Magnetic resonance imaging (MRI) is applied in detecting brain tumors, though manual analysis of big data and the diversity of tumor types are time-consuming and lead to errors. The aim of the current research work (Ayadi et al., 2021) requires an automated computer-aided diagnosis application. However, There have been vital improvements in recent image classification, notably with deep CNNs. This paper presents the new CNN model, dealing with brain tumor classification with the essential support of deep layers in classifying MRI brain images. They have tested their model on three datasets, and from the experimental results, it can be concluded that their approach outperforms many existing methods.

This paper (Aamir et al., 2022)utilizes the automated procedure for classifying brain tumors in MRI proposed. The MRI image is preprocessed for better visualization. Two types of pre-trained feature extraction models are applied to the images. Using a feature extraction process, the extracted features produced a hybrid vector using partial least squares (PLS). Lastly, agglomerative clustering identifies the top locations that contain a tumor. These locations are later resized based on this resizing; the final classification uses a head network. The classification accuracy obtained on the dataset from the proposed method was 98.95%.

This work (Saleh et al., 2020) enhances the classification of brain tumors from MRI by AI algorithms, particularly by CNN and deep learning. This approach has trained a brain tumor dataset with five fine-tuned models: Xception, InceptionV3, ResNet50, MobileNet, and VGG16. The highest F1-score reached was 98.75% with the Xception model. These high accuracies increase early tumor detection significantly and help in effective treatment planning, thus preventing severe physical side effects.

DeepTumorNet(Amran et al., 2022) is introduced as a hybrid DL model for classifying three types of BTs: glioma tumor, meningioma tumor, and pituitary tumor. It is based on the GoogLeNet CNN architecture, modified to drop the last five layers before adding fifteen new layers. Leaky ReLU was added as an activation function to improve feature extraction. The model went well on a freely available dataset: its accuracy stood at 99.67%, precision at 99.6%, and recall at 100%, with an F1-score of 99.66%. These results show improved performance over the rest of the state-of-the-art models, including AlexNet, ResNet50, Darknet53 model, GoogLeNet model, ShuffleNet, SqueezeNet model, ResNet101 model, XceptionNet model, and mobileNetv2.

# Research Methodology

## Proposed Methodology

I designed this research to classify brain tumors using DL models on Magnetic Resource Imaging images. This study comprises the collection and preprocessing of MRI datasets, development, training, and finally, evaluation of these four DL models: VGG16 model, VGG19 model, CNN model, and InceptionV3 model. These models are trained against a labeled dataset to validate their accuracy in classifying four classes of brain tumors: glioma tumor, meningioma tumor, no tumor, and pituitary tumor. This means that the general objective of this work is to establish the most accurate model of brain tumor classification, as presented in Figure 3.1.

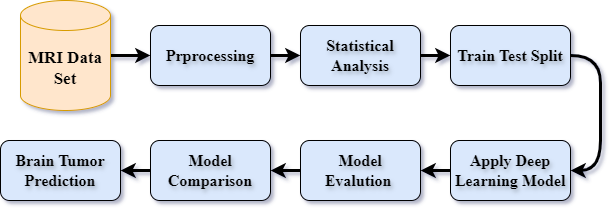


Figure 3.1 Diagram of our proposed methodology

## Data Collection

In this work, the dataset used is the "Brain Tumor MRI Dataset" from Kaggle (*Brain Tumor MRI Dataset*, n.d.).The dataset is split into training and testing directories, with images categorized into four classes such as glioma tumor, meningioma tumor, no tumor, and pituitary tumor.

## Data Analysis

This section thoroughly analyzed the dataset to show how images were distributed across the training and test sets. The training set consists of 5,712 images, while the evaluation set has 1,311. A class-wise distribution was also performed to understand the representation of each tumor type. The glioma class consists of 1,621 images, the pituitary tumor class has 1,757 photos, and the meningioma class includes 1,645 and belongs to nontumor, which consists of 2,000 pieces. This analysis showcases how balanced representations among different tumor types are essential. Figure 3.2 shows image distributions among other classes that emphasize shapes within the dataset and accentuate the need for balanced training to burn stamina into the machine learning performance of deep convolutional networks.

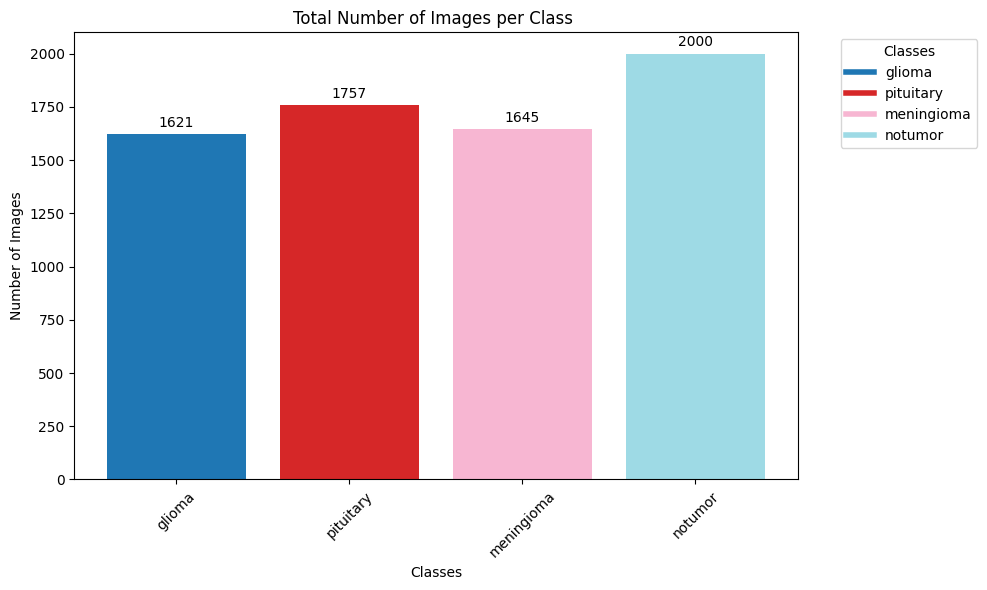


Figure 3.2 Data Distribution across Each class

Selective samples were drawn from each class to clarify the MRI images better and then examined. This process has been vital to dispel any confusion regarding early picture identification characteristics of various types of brain tumors and the nontumor category. By plotting these images, researchers can observe the visual distinctions between glioma, pituitary tumors, meningioma, and nontumor MRI scans, as shown in Figure 3.3. Consequently, this step not only makes one with the dataset but also plays a critical role in the initial assessment of the classification models' challenges. Understanding the images' visual features helps refine preprocessing steps and model design to better capture the nuances in the data and for more accurate classification outcomes.

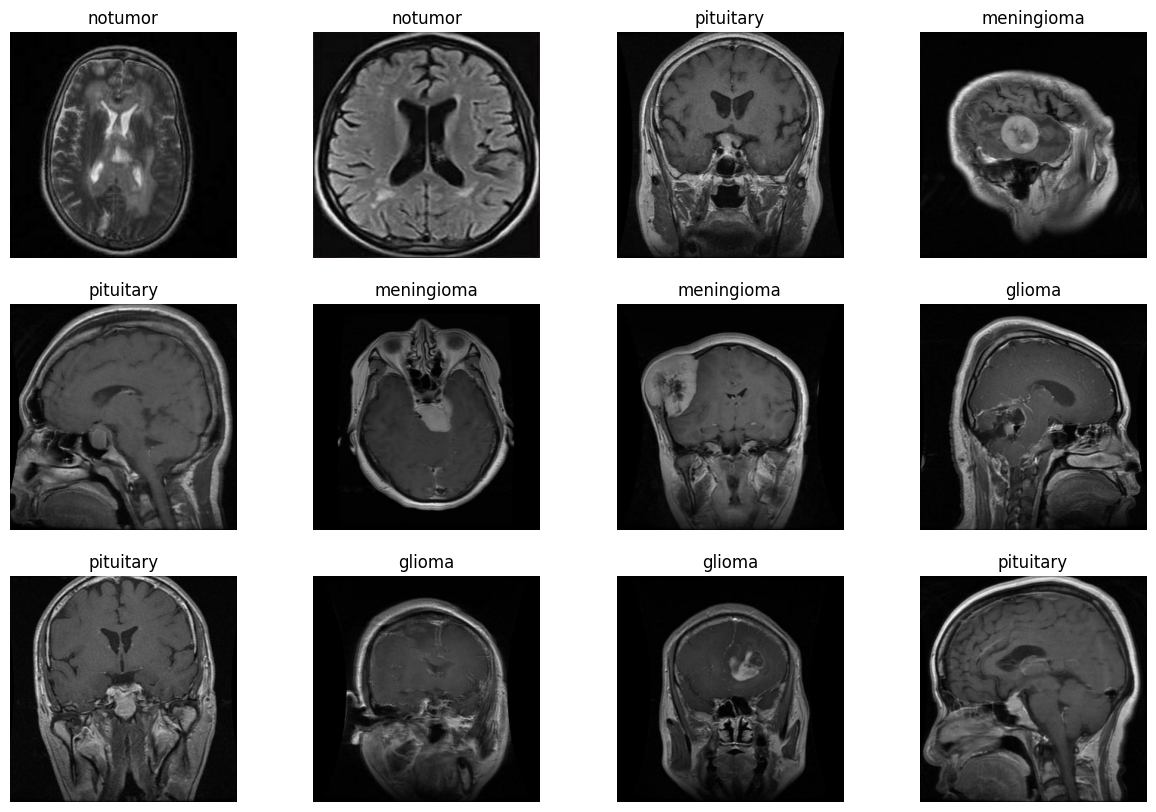


Figure 3.3 Selected sample of MRI images

## Ethical Issues

From an ethical point of view, we note that the Brain Tumor MRI Kaggle dataset does not contain any personal data and is not subject to GDPR. The data is anonymized, and no identifiable information is connected with individual patients. Moreover, it is openly available under a Creative Commons license and thus offers the opportunity to be used for research without further permissions. However, ethical issues regarding data collection must be checked to make sure that a well-recognized and highly-reputed organization has collected it. Since the data was hosted in Kaggle, an established platform for data science competitions and datasets, there is an assumption that the data had been ethically sourced. Since no new data is collected in this project, and since no questionnaires will be administered, UH ethical approval will not be required.

## Data Preprocessing

The preprocessing step is crucial to properly format the MRI images and prepare them for input into the model. All the photos were resized to 224×224 pixels to be consistent across all models. I applied data augmentation techniques using the ImageDataGenerator class to artificially increase the training dataset's size by generating image variants. The augmentation process involved horizontal flipping, random rotation, shearing, height, width shifts, and zooming. These techniques help prevent overfitting and provide better generalization ability in models.

## Feature Engineering

In this work, Feature Engineering was conducted using pre-trained models like VGG16, VGG19, and InceptionV3 for brain tumor classification. These models were used as feature extractors, wherein the convolutional layers were frozen, and new fully connected layers were stacked to fine-tune the models for brain tumor classification.

## Model Selection

Four DL models, a VGG16 model, a VGG19 model, an InceptionV3 model, and a custom-built CNN are built in this study. These models have been chosen for the study based on prior knowledge about their performance concerning image classification problems, especially in medical imaging. Each model has its strengths, and all their architectures are designed to capture the minute features present in the MRI images of brain tumors.

### VGG16 and VGG19

VGG16 and VGG19 are very famous models in the Visual Geometry Group. Characterized deep architectures have multiple convolutional layers combined with fully connected layers. The most important innovation of the VGG model is the tiny filters across the network. It will then capture the fine details of images while maintaining computational efficiency per requirements. Below is the general structure of a VGG model represented by the following equations:

**Convolutional Layers:** Each convolutional Layer applies a filter.

Where 𝑏 is the bias term, ∗ denotes the convolution operation, and 𝑓 is the RELU activation function.

**Max Pooling:** is used to reduce the spatial dimensions:

Where represents the local region in the feature map.

The flattened output from the final convolutional Layer undergoes one or more fully connected layers:

where and It is the fully connected layer's parameters, weights, and biases.

**Softmax Output:** The softmax function at the last output layer produces class probabilities.

VGG16 includes 16 layers, featuring 13 convolutional layers and three fully connected layers, whereas VGG19 has 19 layers, consisting of 16 convolutional layers and three fully connected layers. Pre-training is done for each model on the ImageNet dataset and fine-tuned on the Brain Tumor MRI data set to borrow advantageous features using transfer learning.

### InceptionV3

InceptionV3 is a deep convolutional network belonging to the family of Inception. The model uses an Inception module, which applies different sizes of convolution filters on the image in parallel and concatenates their outputs to capture the multi-scale features inherent in the image efficiently.

Mathematically, an Inception module can be explained as follows:

Parallel Convolutions: Several convolution operations are applied parallel to the same input.

where ​ are filters of sizes 1x1, 3x3, and 5x5, and are the corresponding biases.

Concatenation: These parallel convolutions' feature maps are concatenated:

Reduction Layer: The concatenated output may be followed by a pooling or dimensionality reduction layer to control the computational cost.

InceptionV3 introduced several optimizations, including factorized convolutions, which go a long way in further lightening the computational load but maintaining high accuracy. The model has a more complicated architecture than VGG but captures diverse features from MRI images.

### Custom CNN

The custom CNN will serve as a baseline to which the more complex models like VGG and Inception can be compared. It consists of convolutional layers followed by pooling layers, then fully connected layers, and ends with a softmax output layer. This custom CNN can be represented mathematically through the following equations:

Convolutional Layer: the convolution is performed as:

Pooling Layer: A max-pooling layer decreases the size of the feature map

Dense Layer: The pooled feature map is flattened, and the flattened feature maps pass through dense layers.

Output Layer: The last dense layer applies the softmax function to give the predicted probabilities for four tumor classes.

This simple model still has a good way of capturing the most crucial features and acts as a baseline to check the performance of more advanced models.

## Model Implementation

All the models were implemented using TensorFlow and Keras. Each model from VGG16 and VGG19 was made where pre-trained weights were loaded while the convolutional layers were frozen to retain the learned features. A custom fully connected layer with 256 units was introduced, followed by a dropout layer to mitigate overfitting. It was trained using a CNN built from the ground up, featuring three convolutional layers, followed by max-pooling layers, and concluding with a dense layer for classification. The fourth model utilized was InceptionV3, which retained the pre-trained base, with only the top layers replaced by a global average pooling layer, followed by a dense layer and a dropout layer.

Each model was compiled using the Adam optimizer with a learning rate of 0.0001 and trained with categorical cross-entropy loss. They were trained for 50 epochs with batch size 32, and their performance was validated using a separate validation set.

## Training and Evaluation

The trained models used the augmented training dataset, and the performance was tested on the test dataset. During training, performance was monitored, including training and validation accuracy and loss. The performance of the trained models was finally assessed in terms of precision, recall, the F1-score, and the confusion matrix for their classification performance.

Finally, in the process, some predictions over the test dataset will be generated and compared against the hold-out actual labels. This would give a detailed account of how the models perform through a confusion matrix and a classification report across various classes.

## Performance Metrics

We employed several vital metrics to evaluate the performance of DL models. Precision, Recall, Accuracy, and F1-Score. These metrics are crucial for assessing the effectiveness of the models in classifying brain tumor images into the correct categories.

### Precision

**It measures the accuracy of the optimistic predictions made by the model. It also indicates how many instances are predicted as positive by the model are positive. Precision is precious when high costs are associated with false positives.**

Mathematically, precision is defined as:

​

Where:

* TP (True Positives) measures how many positive instances were predicted correctly.
* Falses positives refer specifically to situations where the computer says it's a cat but not a cat.

### Recall

Recall, or Sensitivity or True Positive Rate, measures how well the model can find all the relevant instances with a positive value. The ratio of the true positives to the actual number of positives in the dataset is what it is about.

Recall is defined as:

Where:

* TP (True Positives) measures how many positive instances were predicted correctly.
* FNs (false negatives) refer to the actual positive cases the model wrongly predicted to be negative.

A high recall signifies that the model effectively detects the majority of positive instances, resulting in few false negatives.

### F1-Score

it is a measure that computes the harmonic mean of precision and recall to achieve a balance. 1-Score is very important when there is unbalanced class distribution or when precision and recall are critical. The F1-Score also considers false positives and false negatives in such cases that one should not be biased towards precision or recall in model performance.

The F1-Score is defined as:

F1-Score ranges between 0 and 1, with a value of 1 representing perfect recall and precision. A high F1-Score demonstrates that the model performs well in recall and precision.

## Experimental Setup

Experiments were conducted using a Google collaboration with enough power, and all models were implemented in TensorFlow and Keras libraries using Python. Seaborn and Matplotlib are used for data visualization and analysis.

# Results

In this section, I shall delve into the findings of DL models used in the study. Each model has been subjected to extensive evaluations through plotting training histories and confusion matrices. Insights into different aspects of how accurate they were or not at various training and testing stages are provided in these visualizations. These results will be explored more deeply to give a critical analysis that would enable us to examine how well each setup can classify brain tumor images accurately. Such analysis will reveal where there are strengths and weaknesses and, therefore, facilitate future advancements.

## Class-wise results of Applied DL model

Different performance outcomes were noted among the four models developed to categorize brain cancers. The VGG19 model had an exceptional overall performance, especially in classifying no-tumor cases, which were rewarded with a precision score 0.99, and F1-score 0.99 and recall score 0.99. It was also good at identifying pituitary tumors precision score of 0.94, recall score 0.99, and F1 score 0.96. VGG19 produced precise glioma measurements with 0.92 precision, 0.89 recall, and F1-score rated as 0.91, respectively. The meningioma category had precision rates of 0.90, with recalls accounting for about 87%, resulting in an F1-score of 0.89, as presented in Table 4.1.

In addition to that, the VGG16 model exhibited high levels of performance in precision and recall, especially when distinguishing no tumor cases at all, which earned it 0.98 precision, 1% less than the first number but almost equal to 99% observed in both recall rate and F-1 score as well respectively. Compared with other models, such as in diagnosing pituitary tumors category, this engine retained good characteristics such as countenance pointing to values around 0.91 for maximum correctness and even metrics represented through recall system equivalents.

In general, overall effectiveness-wise, the best performance are displayed by the VGG19 model mostly when identifying gliomas where correctness was measured at best at 0.92, and misrepresentation rate was seen at its greatest extent as 0.89.

Table 4.1 Class-wise result of DL model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Target Class** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| VGG19 | glioma | 92 | 89 | 91 |
| meningioma | 90 | 87 | 89 |
| notumar | 99 | 99 | 99 |
| pituitary | 94 | 99 | 96 |
| CNN | glioma | 98 | 69 | 81 |
| meningioma | 79 | 57 | 66 |
| notumar | 90 | 96 | 93 |
| pituitary | 67 | 100 | 80 |
| Inception Model | glioma | 81 | 39 | 52 |
| meningioma | 56 | 25 | 34 |
| notumar | 72 | 80 | 76 |
| pituitary | 47 | 92 | 62 |
| VGG16 | glioma | 97 | 83 | 89 |
| meningioma | 84 | 88 | 86 |
| notumar | 98 | 99 | 98 |
| pituitary | 91 | 99 | 95 |

## Overall Results of Applied DL Models

The overall performance of the DL models in classifying brain tumors reveals their effectiveness variances. VGG19 attained the highest accuracy at 0.94, with precision, recall, and F1-score being 0.94. This consistency across metrics indicates VGG19's robustness and dependability for classifying brain tumors.

The VGG16 model also attained an accuracy of 0.93 and precision, recall and F1-score of 0.92 indicating strong predictive ability as shown in Table 4.2.

The custom CNN model was less effective, with test accuracy at 0.81 precisely and 0.83 in the share of true favorable rates, while its sensitivity and F1-score were both 0.80. It showed some reasonable reliability despite having lower performance, as shown in Figure 4.1.

In contrast, Inception had the poorest results. It achieves an accuracy of 0.60, precision of 0.64, sensitivity of 0.59, and F1-score of 0.56. The implication is that it has severe limitations in diagnosis, indicating a high likelihood of missing a diagnosis altogether.

The VGG19 and VGG16 outperform others in terms of performance. They become choices when it comes to effective brain tumor classification

Table 4.2 Overall result of DL models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **Accuracy** | **Recall** | **F1-score** | **Precision** |
| VGG19 | 0.94 | 0.94 | 0.94 | 0.94 |
| CNN | 0.81 | 0.80 | 0.80 | 0.83 |
| Inception Model | 0.60 | 0.59 | 0.56 | 0.64 |
| VGG16 | 0.93 | 0.92 | 0.92 | 0.92 |

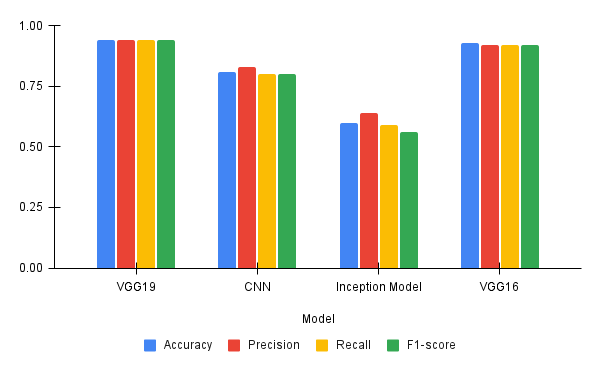
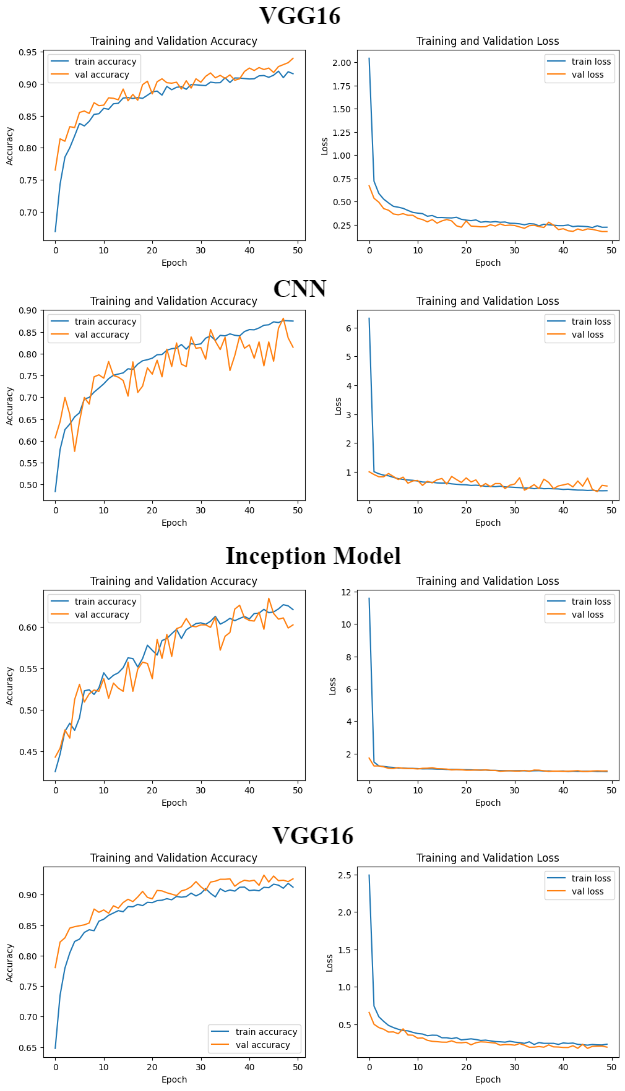


Figure 4.1 Visual Presentation of the Results

## Confusion Matrix and History Graph

In this section, I will analyze the confusion matrix and the training history graphs generated from the results of the DL models. These insights into model performance help us understand how well they can tell classes apart and track their learning journey over time.

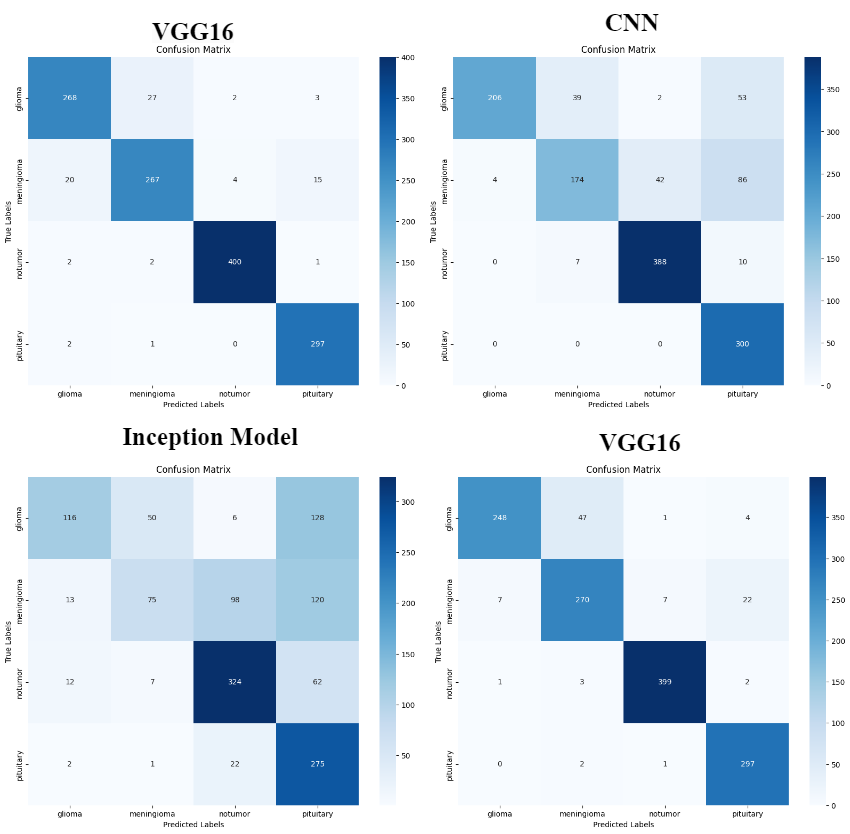
The VGG16 model's training history graph Displays a consistent rise in both training and validation accuracy, with minor fluctuations in the curves. This smooth trajectory indicates that the model is effectively generalizing to the data. In parallel, the validation loss and training loss curves decrease gradually, reinforcing the model's strong generalization capability. In contrast, other DL models in this study present a more erratic pattern on their history curves, characterized by wild swings from one extreme to another, as shown in Figure 4.2. These fluctuations suggest that these models had difficulties achieving consistent performance and did not reach the same generalization level as VGG16.



**VGG19**

Figure 4.2 History graph of DL models

The confusion matrix is valuable for assessing the performance of DL models by providing detailed information on model predictions for various classes. This includes each class's true positives, true negatives, and false positives, summarizing the model's accuracy. For example, it can be observed that the VGG19 model has more true positives than its false negatives, which indicates high predictive abilities. On the other hand, the Inception model has higher rates of false positives and negatives, especially in challenging classes, indicating poor performance and misclassification problems. In conclusion, the confusion matrix helps evaluate the strengths and weaknesses of different models in brain tumor classification.



**VGG19**

Figure 4.3 Confusion Matrix of DL Models

# Analysis and Discussion

## Interpreting the Result

The results obtained from the four DL models, the VGG16 model, the VGG19 model, InceptionV3, and a convolutional neural network developed in-house, show an essential trend in classifying brain tumors from MRI images. The VGG19 was more accurate, with an overall accuracy of 94% in classifying precision, F1-score, and recall, which was also consistently high in every class of tumors. VGG16 wasn't far behind, with an accuracy of 93%. On the other hand, the custom CNN performed poorly but still gave good results, while InceptionV3 gave the poorest results with an accuracy of only 60%.

## Comparative Model Performance

The VGG19 model outperformed the rest, mainly due to the deep architecture and small convolution filters, which could capture intricate details in MRI images. High accuracies and F1 scores achieved in different classes proved that the network was robust enough to cope with the complexity of brain tumor classification. VGG16 performed just as well but with slightly less strength, presumably because it has a shallower architecture compared to VGG19.

The CNN was designed with a good baseline despite the network being less deep and complex. Poor performance could be attributed to the low power of the simpler architecture of InceptionV3 for acceptable feature extraction purposes from MRI images. InceptionV3 performs poorly in the present analysis because of the complex architecture for the betterment of certain tasks. Still, it does not suit the specifics of the brain tumor dataset used well.

## Comparison with Existing Literature

The comparison of these results with the existing literature reveals that the performance of VGG19 is at par with other studies claiming this model is very effective for medical image classification tasks. Results obtained for InceptionV3 in this study are comparatively lower than those reported in other research, but it was mostly mentioned for its efficiency and accuracy. This discrepancy could be due to differences in preprocessing, training, or dataset characteristics. While inferior to the VGG models, it was expected that the performance of the custom CNN would be because simpler models tend to suffer from the complexity of medical image data.

## Study Limitations

Several limitations exist in this study that must be addressed. First of all, the dataset's size and variety may affect the models' generalization capabilities. Additionally, VGG19 and VGG16, although good, tend to have a lot of model size and computational complexity and may become problematic in practical deployability, particularly on lower-end hardware. Regarding how the InceptionV3 model fared relatively poorly, questions of applicability will be channeled to this specific task, which indicates a possible mismatch in the model architecture and the data characteristics.

## Relation to Project Objectives

This study's results align with this project's purpose, especially in finding the model that can perform the best classification of brain tumors. Whence the success of VGG19 and VGG16 in this mission, the first hypothesis of the introduced DL model in making an essential leap towards improved performance in tumor classification, the results also showed that the model should be chosen and adjusted appropriately according to the features and settings of the specific data and application.

## Practical Applicability of Models

Among those tested, VGG19 and VGG16 are some models that can be readily applied using medical diagnosis. They are higher-accuracy models used for deployment in real-life situations, thereby assisting radiologists in brain tumor classification. However, the computational demand is high, which makes them suitable candidates for deployment in setups with limited resources.

Therefore, this confirms a significant research question on whether MRI images could effectively be used for brain tumor classification using DL models, a question presumed up to this point for granted. From the results, DL models, especially VGG19 and VGG16, have shown high accuracy and reliability for brain tumor classification. However, this point does not minimize the careful selection of models already discussed and the potential for further optimization for practical implementation.

## Future Work

Future research should address the limitations identified by this study. Among these, the dataset should be expanded so that the model's generalization can also be boosted. In further research, transfer learning and data augmentation are necessary to exploit and maximize the model's performance. Further optimization of VGG models for deployment in resource-constrained environments will significantly improve the practical value of the models. Please search for other hybrid approaches for creating DL architectures and their betterment in accuracy and efficiency in the classification of brain tumors.

# Conclusion

In this research, four DL models—VGG19, VGG16, InceptionV3, and a custom CNN—are applied and comparatively evaluated for classifying brain tumors against MRI images. The results show that VGG19 and VGG16 performed better, wherein the former fares are the best regarding accuracy, precision, F1-score, and recall of all classes of studied brain tumors. These results show the potential of DL models in classifying brain tumors with accuracy and high reliability, mostly deep architectures providing significant support for medical diagnostics.

These models designed in this work can be directly used in the clinical field to aid radiologists in diagnosing brain tumors, increasing speed and accuracy. The study also exposed several limitations, such as computational complexity, presumably to be used in models deployed in resource-constrained environments. Future work should, therefore, aim to optimize the models for use in such environments, extend the dataset to improve model generalization, and further explore DL architectures that might balance out accuracy and computational efficiency. Moreover, the critical steps in translating this research into application would be embedding these models into clinical workflows and validating them with real-world data.

# Reference

Aamir, M., Rahman, Z., Dayo, Z. A., Abro, W. A., Uddin, M. I., Khan, I., Imran, A. S., Ali, Z., Ishfaq, M., Guan, Y., & Hu, Z. (2022). A DL approach for brain tumor classification using MRI images. Computers and Electrical Engineering, 101, 108105. https://doi.org/10.1016/J.COMPELECENG.2022.108105

Abiwinanda, N., Hanif, M., Hesaputra, S. T., Handayani, A., & Mengko, T. R. (2019). Brain Tumor Classification Using Convolutional Neural Network. IFMBE Proceedings, 68(1), 183–189. https://doi.org/10.1007/978-981-10-9035-6\_33

Amran, G. A., Alsharam, M. S., Blajam, A. O. A., Hasan, A. A., Alfaifi, M. Y., Amran, M. H., Gumaei, A., & Eldin, S. M. (2022). Brain Tumor Classification and Detection Using Hybrid Deep Tumor Network. Electronics 2022, Vol. 11, Page 3457, 11(21), 3457. https://doi.org/10.3390/ELECTRONICS11213457

Ayadi, W., Elhamzi, W., Charfi, I., & Atri, M. (2021). Deep CNN for Brain Tumor Classification. Neural Processing Letters, 53(1), 671–700. https://doi.org/10.1007/S11063-020-10398-2/TABLES/29

Brain Tumor MRI Dataset. (n.d.). Retrieved August 10, 2024, from https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

Işin, A., Direkoǧlu, C., & Şah, M. (2016). Review of MRI-based Brain Tumor Image Segmentation Using DL Methods. Procedia Computer Science, 102, 317–324. https://doi.org/10.1016/J.PROCS.2016.09.407

Rosenfeld, S. S., & Massey, E. W. (2019). Epidemiology of Primary Brain Tumor. Neuroepidemiology, 121–144. https://doi.org/10.1201/9780429277276-8

Saleh, A., Sukaik, R., & Abu-Naser, S. S. (2020). Brain tumor classification using deep learning. Proceedings - 2020 International Conference on Assistive and Rehabilitation Technologies, ICareTech 2020, 131–136. https://doi.org/10.1109/ICARETECH49914.2020.00032

Seetha, J., & Selvakumar Raja, S. (2018). Brain Tumor Classification Using Convolutional Neural Networks. Biomedical & Pharmacology Journal, 11(3), 1457–1461. https://doi.org/10.13005/bpj/1511

Singhal, A., Sinha, P., & Pant, R. (2017). Use of DL in Modern Recommendation System: A Summary of Recent Works. International Journal of Computer Applications, 180(7), 17–22. https://doi.org/10.5120/ijca2017916055

Wadhwa, A., Bhardwaj, A., & Singh Verma, V. (2019). A review on brain tumor segmentation of MRI images. Magnetic Resonance Imaging, 61, 247–259. https://doi.org/10.1016/J.MRI.2019.05.043

# Appendices

Install kaggle library to download the MRI data in google collab

!pip install kaggle# install kaggle library

!mkdir -p ~/.kaggle

!cp kaggle.json ~/.kaggle/# import jason file

!chmod 600 ~/.kaggle/kaggle.json

Download MRI data

!kaggle datasets download -d masoudnickparvar/brain-tumor-mri-dataset

Unzip Data

!unzip brain-tumor-mri-dataset.zip

Showing the statics how many images we have for training and for testing

import tensorflow as tf # import the tensorflow library

from tensorflow.keras.preprocessing import image\_dataset\_from\_directory

trainDire = '/content/Training'

testDire= '/content/Testing'

batchSize = 32 # setting the batch

imgheight = 300 # setting the height and width

imgwidth = 300

trainDataset = imageDatasetFromDirectory(

trainDire,

validation\_split=0.2,

subset="training",

seed=13,

imageSize=(imgheight, imgWidth),

batchSize=batchSize

)

validationDataset = imageDatasetFromDirectory(

trainDire,

validationSplit=0.2,

subset="validation",

seed=3,

image\_size=(imgHeight, imgWidth),

batch\_size=batch\_size

)

testDataset = imageDatasetFromDirectory(

testDire,

imageSize=(imgHeight, imgWidth),

batchSize=batchSize

)

classNames = trainDataset.class\_names

trainDataset = trainDataset.cache().shuffle(1024).prefetch(buffer\_size=tf.data.AUTOTUNE)

validationDataset = validationDataset.cache().prefetch(buffer\_size=tf.data.AUTOTUNE)

testDataset = testDdataset.cache().prefetch(buffer\_size=tf.data.AUTOTUNE)

import matplotlib.pyplot as pltt # for plot graphs

import seaborn as sns # for plot graphs

import warnings # for ignoring the warning in the code

warnings.filterwarnings('ignore')

pltt.figure(figsize=(15, 10))

for img, labels in trainDataset.take(1):

for x in range(12):

ax = pltt.subplot(3, 4, x + 1)

pltt.imshow(images[i].numpy().astype("uint8"))

pltt.title(class\_names[labels[i]]) # plot the title on the bar

pltt.axis("off")

pltt.show()

**Import necessary librarires**

import numpy as np # for arthimatic operation

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications.vgg16 import preprocess\_input as vgg\_preprocess

from tensorflow.keras.applications.inception\_v3 import preprocess\_input as inception\_preprocess

from sklearn.metrics import classification\_report, confusion\_matrix

**Import the train test dataset**

# Directories

trainDire = '/content/Training'

testDire= '/content/Testing'

# Data generators

train\_datagen = ImageDataGenerator(

preprocessing\_function=vgg\_preprocess,

rotationange=40,

width\_shift\_range=0.2

shear\_range=0.2,

fill\_mode='nearest'

height\_shift\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

)

test\_datagen = ImageDataGenerator(preprocessing\_function=vgg\_preprocess)

train\_generator = train\_datagen.flow\_from\_directory(

train\_dir,

target\_size=(224, 224),

batch\_size=32,

class\_mode='categorical'

)

testGenerator = test\_datagen.flow\_from\_directory(

testDire,

targetSize=(224, 224),

class\_mode='categorical',

batchSize=32,

shuffle=False

)

**Import library for model building**

# Model building functions

from tensorflow.keras.applications import VGG16, VGG19, InceptionV3

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Flatten, Conv2D, Dense, MaxPooling2D, GlobalAveragePooling2D, Dropout,

from tensorflow.keras.optimizers import Adam

**Define function for model building**

def build\_vgg16\_model():

baseModel = VGG16(weights='imagenet', input\_shape=(224, 224, 3), include\_top=False)

model = Sequential()

model.add(base\_model)

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(4, activation='softmax'))

baseModel.trainable = False

model.compile( loss='categorical\_crossentropy', optimizer=Adam(learning\_rate=0.0001), metrics=['accuracy'])

return model

def build\_vgg19\_model():

baseModel = VGG19(weights='imagenet', include\_top=False, input\_shape=(224, 224, 3))

model = Sequential()

model.add(base\_model)

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(4, activation='softmax'))

base\_model.trainable = False

model.compile( loss='categorical\_crossentropy', optimizer=Adam(learning\_rate=0.0001), metrics=['accuracy'])

return model

def build\_cnnModel():

model = Sequential()

model.add(Conv2D(32, (3, 3), input\_shape=(224, 224, 3)), activation='relu')

# adding the maxpoling layer

model.add(MaxPooling2D((2, 2)))

# another conv2d layer

model.add(Conv2D(64, (3, 3), activation='relu'))

# adding the maxpolling layer

model.add(MaxPooling2D((2, 2)))

# another conv2d layer

model.add(Conv2D(128, (3, 3), activation='relu'))

# adding the maxpolling layer

model.add(MaxPooling2D((2, 2)))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(4, activation='softmax'))

model.compile(loss='categorical\_crossentropy',metrics=['accuracy'], optimizer=Adam(learning\_rate=0.0001))

return model

def build\_inception\_model():

baseModel = InceptionV3(input\_shape=(224, 224, 3) , weights='imagenet', include\_top=False)

model = Sequential()

model.add(base\_model)

model.add(GlobalAveragePooling2D())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(4, activation='softmax'))

base\_model.trainable = False

model.compile(loss='categorical\_crossentropy', optimizer=Adam(learning\_rate=0.0001), metrics=['accuracy'])

return model

**Evaluation and plotting functions**

# Evaluation and plotting functions

def evaluateModel (model, testGenerator):

testGenerator.reset()

predictions = model.predict(testGenerator)

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

trueClasses = testGenerator.classes

classLabels = list(testGenerator.class\_indices.keys())

report = classificationReport(trueClasses, predictedClasses, targetNames=classLabels)

print(report)

cm = confusion\_matrix(trueClasses, predictedClasses)

print(cm)

return report, cm

def plotConfusionMatrix cm, class\_labels):

pltt.figure(figsize=(10, 8))

sns.heatmap(cm, fmt='d', xticklabels=class\_labels, cmap='Blues', annot=True, yticklabels=class\_labels)

pltt.ylabel('True Labels')

pltt.xlabel('Predicted Labels')

pltt.title('Confusion Matrix')

pltt.show()

def plot\_training\_history(modelhistory):

pltt.figure(figsize=(12, 4))

pltt.subplot(1, 2, 1)

pltt.plot(modelhistory.history['accuracy'], label='train accuracy')

pltt.plot(modelhistory.history['val\_accuracy'], label='val accuracy')

pltt.ylabel('Accuracy')

pltt.xlabel('Epoch')

pltt.legend()

pltt.title('Training and Validation Accuracy')

pltt.subplot(1, 2, 2)

pltt.plot(modelhistory.history['loss'], label='train loss')

pltt.plot(modelhistory.history['val\_loss'], label='val loss')

pltt.ylabel('Loss')

pltt.xlabel('Epoch')

pltt.legend()

pltt.title('Training and Validation Loss')

pltt.show()

vgg19\_model = build\_vgg19Mmodel()

history = vgg19\_model.fit(

trainGenerator,

epochs=50,

validationData=testGenerator)

**vgg19**

# Evaluate the model

report, cm = evaluateModel(vgg19Mmodel, testGgenerator)

# Plot the confusion matrix

class\_labels = list(test\_generator.class\_indices.keys())

plotConfusionMatrix(cm, class\_labels)

# Plot the training history

plot\_training\_history(history)

# Example usage

cnn\_model = build\_cnn\_model()

history = cnn\_model.fit(

train\_generator,

epochs=50,

validation\_data=test\_generator

)

**CNN**

# Evaluate the model

report, cm = evaluateModel(cnn\_model, test\_generator)

# Plot the confusion matrix

classLabels = list(test\_generator.class\_indices.keys())

plotConfusionMatrix(cm, classLabels)

# Plot the training history

plot\_training\_history(history)

# Example usage

inception\_model = build\_inception\_model()

history = inception\_model.fit(

train\_generator,

epochs=50,

validation\_data=test\_generator

)

**inception**

# Evaluate the model

report, cm = evaluateModel(inception\_model, testGenerator)

# Plot the confusion matrix

class\_labels = list(test\_generator.class\_indices.keys())

plotConfusionMatrix (cm, class\_labels)

# Plot the training history

plot\_training\_history(history)

# Example usage

vgg16\_model = build\_vgg16\_model()

history = vgg16\_model.fit(

train\_generator,

epochs=50,

validation\_data=test\_generator

)

# Evaluate the model

report, cm = evaluateModel(vgg16\_model, test\_generator)

# Plot the confusion matrix

class\_labels = list(test\_generator.class\_indices.keys())

plotConfusionMatrix (cm, class\_labels)

# Plot the training history

plot\_training\_history(history)