Inteligencia Artificial 28(75) (2025), 81-100
doi: 10.4114/intartif.vol28iss75pp81-100

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

# An Efficient Deep Learning Technique for Brain Abnormality Detection Using MRI

Shilpa Mahajan[1], Aryan Dahiya[2], Anuradha Dhull[3]

[1] Computer Science Engineering (CSE) The NorthCap University, Gurugram, HR – 122017
[2] Computer Science Engineering (CSE) The NorthCap University, Gurugram, HR – 122017
[3] Computer Science Engineering (CSE) The NorthCap University, Gurugram, HR – 122017
[1] shilpa@ncuindia.edu
[2] aryandahiya15@gmail.com
[3] anuradha@ncuindia.edu

**Abstract** This research proposes an effective and reliable deep learning method for detecting brain abnormalities via magnetic resonance imaging (MRI). The technique consists of two primary stages: first, a binary classifier that divides pictures into "Brain" and "Non-Brain" categories; second, multi-class classifiers that explicitly recognise categories such pituitary adenomas, gliomas, and meningiomas. The labelled and preprocessed data were taken from a collection of 7,753 pictures provided by Qhills Technologies Pvt. Ltd. Additional data from the Brain Tumour MRI collection was also incorporated to improve the model's generalisation skills. VGG-16 outperforms the other machine learning models, with an accuracy rate of 96.4%, when compared to ANN, CNN, VGG-16, and AlexNet. A thorough model evaluation and hyperparameter tweaking process was conducted using the accuracy, precision, recall F1-score. The findings of this study point to the potential of deep learning techniques in identifying brain disorders fast and precisely, opening the door to more precise diagnosis in clinical settings.

**Keywords**. Deep learning, brain abnormality detection, MRI images, VGG-16, multi-class classification, medical image analysis, hyperparameter tuning, dataset diversity

## 1 Introduction

The human brain, an intricate masterpiece of nature, orchestrates the symphony of our thoughts, emotions, and actions. Its complexity is what makes it fascinating and challenging at the same time, especially when trying to comprehend and cure problems that impact it. Brain tumours have a prominent place among them. Numerous neurological disorders can result from this delicate equilibrium being upset by these aberrant cell growths in the brain. Accurate diagnostic instruments are necessary to treat brain tumours successfully, and this is where medical imaging more specifically, Magnetic Resonance Imaging (MRI) comes into play. [1]

An aberrant mass of tissue in the brain that can impair normal brain function is called a brain tumor. Based on the type of cell that gave rise to the tumor, its location within the brain, and its growth pattern, brain tumors can be categorized. As seen in Figure 1, gliomas, meningiomas, and pituitary tumors are the three most prevalent forms of brain tumors. Tumors called gliomas develop from the glial cells in the brain that sustain and shield the neurons. The most prevalent kind of adult brain tumor is glioma. Tumors can be categorized as high-grade or low-grade. Slow-growing low-grade gliomas usually have a more favourable prognosis than high-grade gliomas. Tumors that originate from the meninges, the membranes around the brain and spinal cord, are known as meningiomas. Most meningiomas are slow-growing, benign tumors. Some meningiomas, however, might be

**ISSN: 1137-3601 (print), 1988-3064 (on-line)**

aggressive and necessitate surgery. Pituitary tumours are tumours that start in the pituitary gland, a small organ located close to the base of the brain.

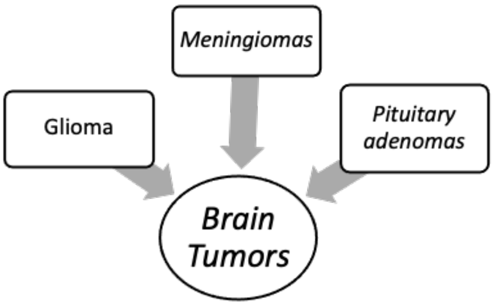


Figure 1. Types of brain tumors

It is imperative to have accurate imaging methods to identify brain tumors. Aside from aiding in tumor location, medical imaging also offers important details regarding the size, shape, and effect of the tumor on neighbouring brain tissue. Magnetic Resonance Imaging (MRI) is a standout imaging technique among the others. Rich images of the structures of the brain are produced by MRI using radio waves and strong magnets. Physicians can differentiate between various kinds of brain tissue and anomalies thanks to the superior soft tissue contrast provided by this non-invasive tech.

Brain tumor detection requires the use of magnetic resonance imaging (MRI) technology because of its versatility and impact on patient outcomes. 1.4% of all new instances of cancer, including brain tumors, involve the nervous system. This information comes from data provided by the American Cancer Society. Brain tumors can be accurately identified by MRI, which provides important information for medical professionals to assess their characteristics and create treatment options. It guides surgeons throughout surgeries, helps oncologists monitor treatment outcomes, and aids in distinguishing benign from malignant tumors. Two brain MRI scans are side-by-side contrasted and the patient has an aberrant brain tissue, which is seen as a white highlight in the scan's center in figure 2. In contrast to the patient scan on the left, which shows no obvious abnormalities, the MRI on the right side of the patient reveals a brain abnormality.

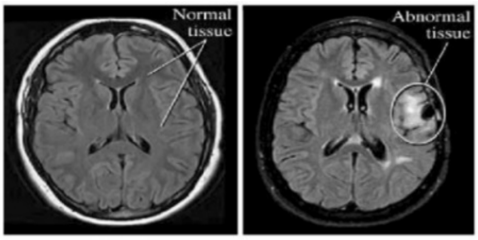


Figure 2. Magnetic Resonance Image of patient highlighting normal tissue and abnormal tissue

These distinctive patterns can be utilized to identify any brain tumors, further assisting us in determining whether the brain is abnormal or not. Deep learning algorithms have advanced the field of medical picture analysis. Traditional approaches frequently used hand-crafted algorithms and manual feature extraction. Convolutional Neural Networks (CNNs), in particular, have proven to have an extraordinary ability to automatically learn pertinent information from photos. This adjustment led to a paradigm shift, which considerably increased diagnostic precision.

Studies contrasting deep learning methods with more conventional machine learning methods have demonstrated the latter's advantage in the detection of brain tumors. Deep learning models routinely achieve accuracy levels exceeding 90%, significantly improving sensitivity and specificity rates. This seismic shift has sped up the adoption of contemporary technology in healthcare, allowing for speedier and more precise diagnosis.

Two sources of images of the brain have been used in this research to improve the MRI dataset and the deep learning model for the classification of brain tumors. The first data set contains 720 images which are offered by

Qhills Technologies Pvt. Ltd, Maharashtra, India. The type of tissues consists of the brain, spine & gallbladder out of which, 500 are of the brain only. To augment this, the Brain Tumor MRI Dataset [2], which contains 7023 Brain MRI images which are classified into 4 classes: glioma - meningioma - no tumor, and pituitary. The merging of these datasets amplified the total amount of images tremendously, as it offered a much richer dataset for training and testing. Thus, the proposed DL approach was used in this study based on the combined dataset to enhance brain tumor classification. Subsequently, the performance of the model was evaluated by Accuracy, Precision, Recall, F1 score, Confusion Matrix in the test set, and RMSE, MAE, and MSE to gauge the effectiveness of this study's proposed deep learning model. The rest of the paper is organized as follows:

- **Section 2** A summary of relevant research on deep learning-based brain tumour diagnostics
- **Section 3** describes in detail about experimental setup, and algorithms used.
- **Section 4** talks about the research's methodology.
- **Section 5** revolves around performance assessment
- **Section 6** discusses the challenges and issues identified in the research
- **Section 7** talks about the future implications of the research

## 2 Literature Review

Ayadi et al. (2021) developed a CNN-based CAD system for brain tumor classification. The study used the 18-layer CNN model, which was trained on a dataset of 3064 brain MRI images. The study achieved an accuracy of 94.74% for brain tumor-type classification and 90.35% for tumor grading. [3]

Brain tumor segmentation and classification using deep learning method was proposed by Wang et al. in 2021. This study used a deep learning model to segment brain tumors in MRI images and then classify the segmented tumors. The study achieved an accuracy of 96.6% in tumor segmentation and 93.7% in tumor classification. [4]

In another study used a deep learning model to classify brain tumors using multimodal MRI images. The study achieved an accuracy of 97.4% in classifying brain tumors. [5]. One of the proposed work used an attention mechanism in a deep learning model to improve the classification of brain tumors. The study achieved an accuracy of 97.7% in classifying brain tumors. [6]

"Brain Tumor Classification Using Deep Learning with Uncertainty Estimation" by Liu et al. (2023). This study used a deep learning model to classify brain tumors and estimate the uncertainty of the classification. The study achieved an accuracy of 96.8% in classifying brain tumors and an average uncertainty of 0.15. [7]. Other research work used self-supervised learning to train a deep-learning model to classify brain tumors. Self-supervised learning is a technique where the model learns to extract features from images without being explicitly labeled. The study achieved an accuracy of 97.2% in classifying brain tumors. [8]. Table 1 provides a concise overview of the previous research in relation to the methods, information, assessment, findings, and their importance.

Table 1. Comparison with others' existing work on brain tumor classification

| **S.No** | **Author** | **Models used** | **Accuracy** | **Paper name** |
|---|---|---|---|---|
| **1** | Ayesha Younis et al. | CNN, VGG-16 , Ensemble Model | CNN 96%<br>VGG-16 98.5%<br>Ensemble Model | Brain Tumor Analysis Using Deep Learning and VGG-16 Assembling Learning Approaches [9] |

| | | | | |
|---|---|---|---|---|
| | | | (98.14%) | |
| **2** | Anushka Singh1 et al. | VGG-16 | VGG-16 (93%) | Brain Tumor Classification Using CNN And Vgg16 Model [10] |
| **3** | P Gayathri et al. | VGG-16 | VGG-16 (94%) | Exploring the Potential of VGG-16 Architecture for Accurate Brain Tumor Detection Using Deep Learning [11] |
| **4** | P Gokila Brindha et al. | ANN & CNN | ANN (80.77%)<br>CNN (89%) | Brain tumor detection from MRI images using deep learning techniques [12] |
| **5** | Arkapravo Chatto padhyay et al. | CNN | CNN (99.7%) | MRI-based brain tumor image detection using CNN-based deep learning method [13] |
| **6** | Aboli Kapadnis et al. | AlexNet | AlexNet (98%) | Brain Tumor Detection using Transfer Learning Technique with AlexNet and CNN [14] |
| **7** | Rehman et al. | VGG-16 | VGG-16 (98.69%) | Brain Tumor Classification Using Deep Learning [15] |
| **8** | Mehrotra et al. | CNN | CNN (94.8%) | A transfer learning approach for AI-based classification of brain tumors [16] |
| **9** | Mzoughi et al. | CNN | CNN (96.49%) | Deep Multi-scale 3D CNN for Brain Tumor Grading from Volumetric MRI Images [17] |
| **10** | S Chatterjee et al. | CNN | CNN (96.5%) | Classification of Brain Tumours in MR Images Using Deep Spatiospatial CNNs [18] |

# 3 Background

Table 2 provides a comprehensive overview of the diverse array of tools that have been meticulously selected and employed for the seamless implementation of our work. Each tool was thoughtfully chosen to address specific aspects of the project's requirements, ensuring a synergistic approach to attaining our goals.

Table 2: Implementation Requirements

| **S.No.** | **Requirements type** | **Tools Used** |
|---|---|---|
| **1.** | Third-party software | Windows 10, Google Collab, VScode |
| **2.** | Coding Language | Python |
| **3.** | Library Used | Pigeon, TensorFlow, Numpy, glob, matplotlib, PIL, sklearn, |

### 3.1 Artificial Neural Network

A popular training architecture for the classification of photographs is the artificial neural network. The image is first transformed into a grid of pixels with values ranging from 0-255 in the classic ANN method. The needed object has values that are only slightly different from those of other components, creating the scope for identifying the difference. A series of neurons in the network's initial layer requires one-dimensional input. Grids must be flattened (they are typically multidimensional, such as 2*2 or 7*7). The several hidden layers work to improve learning. Neurons in every buried or dense layer are connected to every neuron in the layer below them. The weights assigned and the activation function determine the value of the connections. Every iteration calculates the discrepancy between the result that was expected and what was produced. Following the results, the backpropagation begins, and the weights are adjusted appropriately. They continue to alter until the maximum precision is achieved shown in Figure 3. This paradigm does, however, have a few shortcomings like a complicated image of a huge size may produce millions of neurons in the input layer, which is computationally inefficient.

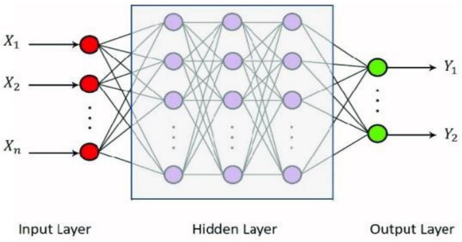


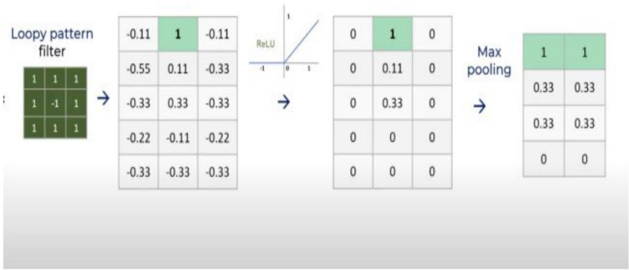
Figure 3. Pictorial View of ANN

## 3.2 Convolution Neural Network

In terms of picture partitioning, the Convolutional Neural Network (CNN) varies from the ANN. CNN learns individual pixels, but the ANN learns the entire image at once. CNN is not impacted by the image's location adjustment. Over ANN, it has a substantial advantage. The three main steps in CNN's operation are as follows:

- Convolutional Operation or Filter
- Activation Function
- Pooling
- Dropout Layer
-

The image grid is multiplied by a smaller matrix in the Convolutional filter. For instance, if the image grid is 5*7, it would be multiplied by a 3*3 matrix of 3*3 grid samples. It produces a feature map with greater values at the cells that contain our target object. The data becomes nonlinear due to the activation function. The network is comparable to linear regression because there isn't an activation function. This method assumes that dependent and independent variables have a linear relationship but in the real world, hardly such a linear relationship exists. A RELU function, for instance, changes negative numbers to zero. It does away with linearity. The pooling layers are in charge of shrinking the object. It guarantees the smaller, information-containing grid is created by downsizing the larger grid. While average pooling chooses the average of the subset values, max pooling selects the highest value from the subset of the grid as defined in Figure 4.



| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -1 | 1 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| -0.11 | 1 | -0.11 |
| -0.55 | 0.11 | -0.33 |
| -0.33 | 0.33 | -0.33 |
| -0.22 | -0.11 | -0.22 |
| -0.33 | -0.33 | -0.33 |

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 0.11 | 0 |
| 0 | 0.33 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

| | |
|---|---|
| 1 | 1 |
| 0.33 | 0.33 |
| 0.33 | 0.33 |
| 0 | 0 |

Figure. 4. CNN model

A dropout layer was also included. Dropout operates by setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase. For all varieties of deep neural networks, this method offers a computationally straightforward and incredibly effective regularisation solution to reduce overfitting and improve generalization error. Through the combination of the aforementioned layers, the various components of the images are learned. When an aggregated version of previously learned partitions crosses the layers once more, the model learns the whole object image. Convolutional filters, activation functions, and pooling layers come in many forms in the CNN model as shown in Figure 5.

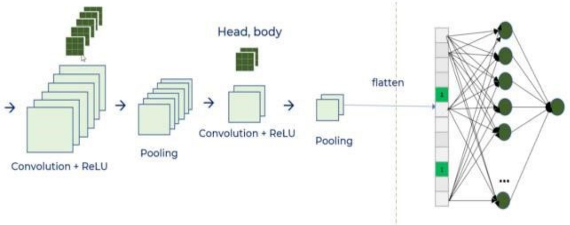


Figure. 5. CNN Model working

### 3.3 Visual Geometric Group

The deep convolutional neural network (CNN) architecture known as VGG-16, or Visual Geometry Group 16, is generally employed for image categorization applications. VGG-16, created by the Visual Geometry Group at the University of Oxford, is distinguished by its depth and simplicity. It contains 3 fully connected layers and 13 convolution layers As such, it has 16 weight layers. It has a well-defined pattern of 2x2 max pooling, and the output is processed using 3x3 convolution layers all through the layers of VGG-16. This homogeneity makes model design less complicated as well as makes training less of a challenge.

VGG-16 is a deep architecture that can describe precise details in photos and does very well in pattern recognition and object detection on a vast array of visualization inputs. In the areas of deep learning and computer vision, it evolved to be famous for a variety of applications that essentially need categorized picture recognition with utmost precision. Due to its efficiency in performing different image-related functions, VGG-16 is normally employed by researchers for transfer learning on several tasks.

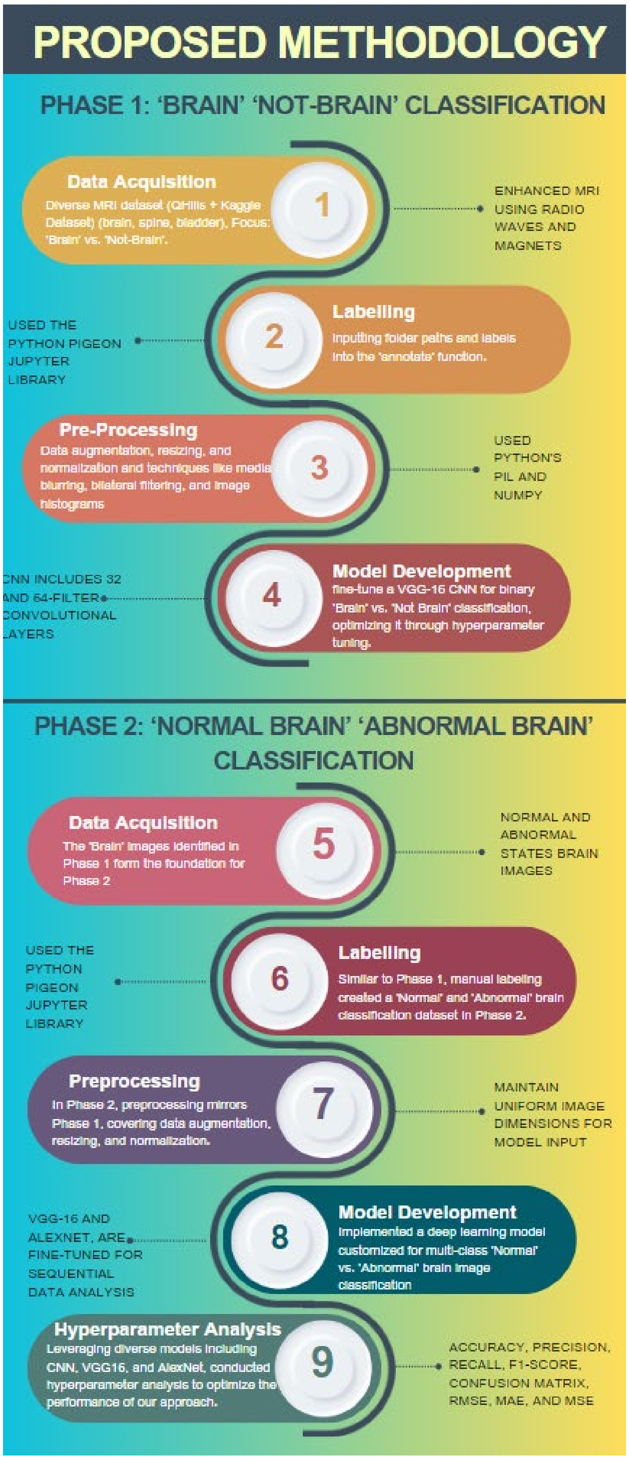
### 3.3 AlexNet

The architecture based on the deep convolutional neural network (CNN) which is known as AlexNet, created by Alex Krizhevsky became a turning point in the field of computer vision. It gained popularity because of its outstanding performance in the 2012 ImageNet Large Scale Visual Recognition Challenge which enhanced the state of the art in image categorization significantly. AlexNet is different from the others by its deeper depth design containing eight layers in total, five of which are convolutional and three – fully linked layers. Some of these include the use of rectified linear units (ReLU) in its activation functions, which reduced the training time by preventing the vanishing gradient problem.

Also, the overfitting was reduced through what was attributed to dropout and data augmentation in the AlexNet. The current model architecture is designed to include two sets of convolutional paths that run in parallel, yielding the appreciation of the fine as well as the coarser details of the images. The current success of AlexNet paved the way for new developments of deep-learning in computer vision and its effect is still a crucial role in the formation of CNN architectures for image elucidation applications.

## 4 Proposed Methodology

The proposed methodology focused on the detection of brain abnormalities from MRI images is divided into two phases: They include Phase I and Phase II which deal with two major stages of the detection process. The first envisioned phase involves sorting the images as either “Brain” or “Not Brain. ” The second phase involves sorting the images as “Normal” and “Abnormal” Brain images. If the brain image is found to be abnormal, the method then classifies the type of brain tumor into three categories: It may be used for conditions such as glioma, meningioma, and pituitary adenoma. This additional classification step improves the model’s diagnostic capacity by not only identifying the presence of an abnormality but also its type. These two phases collectively form a strong pattern for establishing an accurate diagnosis. Each phase consists of the following steps: In the first step, the data is collected during the second step the data is stratified and labelled, in the third step the data is preprocessed and in the final step the data is used to develop the model and this is depicted in figure 6.



PROPOSED METHODOLOGY

PHASE 1: ‘BRAIN’ ‘NOT-BRAIN’ CLASSIFICATION

1 Data Acquisition
Diverse MRI dataset (QHills + Kaggle Dataset) (brain, spine, bladder), Focus: 'Brain' vs. 'Not-Brain'.
ENHANCED MRI USING RADIO WAVES AND MAGNETS

2 Labelling
Inputting folder paths and labels into the 'annotate' function.
USED THE PYTHON PIGEON JUPYTER LIBRARY

3 Pre-Processing
Data augmentation, resizing, and normalization and techniques like media blurring, bilateral filtering, and image histograms
USED PYTHON'S PIL AND NUMPY

4 Model Development
fine-tune a VGG-16 CNN for binary 'Brain' vs. 'Not Brain' classification, optimizing it through hyperparameter tuning.
CNN INCLUDES 32 AND 64-FILTER CONVOLUTIONAL LAYERS

PHASE 2: ‘NORMAL BRAIN’ ‘ABNORMAL BRAIN’ CLASSIFICATION

5 Data Acquisition
The 'Brain' images identified in Phase 1 form the foundation for Phase 2
NORMAL AND ABNORMAL STATES BRAIN IMAGES

6 Labelling
Similar to Phase 1, manual labeling created a 'Normal' and 'Abnormal' brain classification dataset in Phase 2.
USED THE PYTHON PIGEON JUPYTER LIBRARY

7 Preprocessing
In Phase 2, preprocessing mirrors Phase 1, covering data augmentation, resizing, and normalization.
MAINTAIN UNIFORM IMAGE DIMENSIONS FOR MODEL INPUT

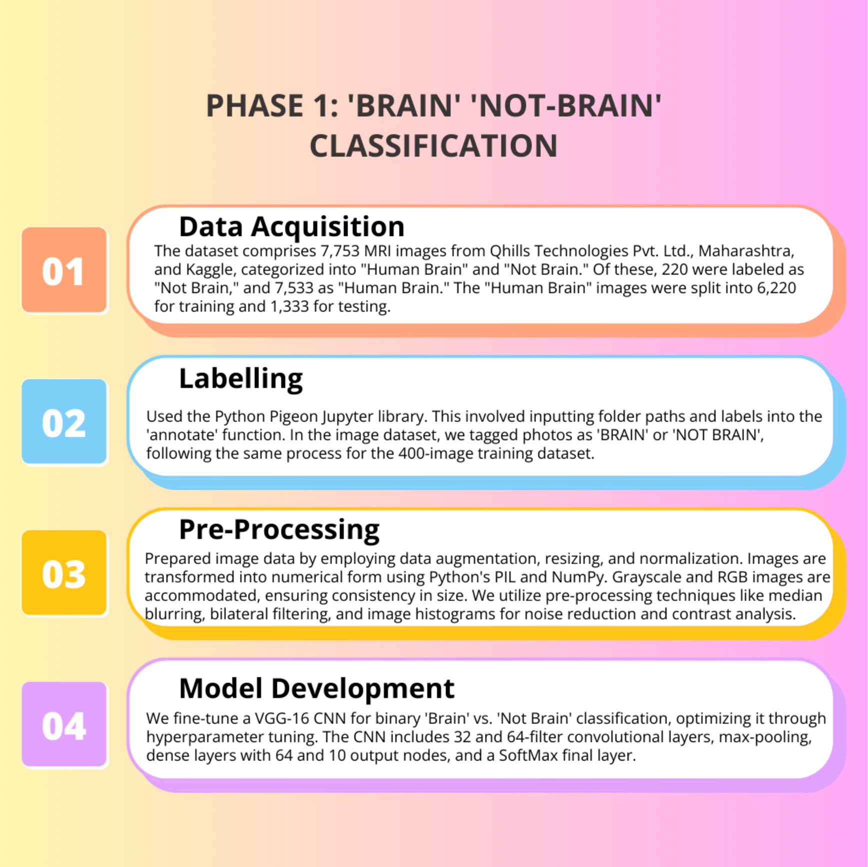
8 Model Development
Implemented a deep learning model customized for multi-class 'Normal' vs. 'Abnormal' brain image classification
VGG-16 AND ALEXNET, ARE FINE-TUNED FOR SEQUENTIAL DATA ANALYSIS

9 Hyperparameter Analysis
Leveraging diverse models including CNN, VGG16, and AlexNet, conducted hyperparameter analysis to optimize the performance of our approach.
ACCURACY, PRECISION, RECALL, F1-SCORE, CONFUSION MATRIX, RMSE, MAE, AND MSE

Figure 6. Proposed Methodology

## 4.1 Phase 1:Brain-Not Brain Classification

In Phase 1 of our proposed methodology for efficient brain abnormality detection using MRI images, our primary phase is to distinguish between images that contain brain structures ("Brain") and those that do not ("Not Brain"). Each step performed in Phase 1 is shown in Figure 7. This phase serves as the foundational step in our two-phase approach.



PHASE 1: 'BRAIN' 'NOT-BRAIN' CLASSIFICATION

01 **Data Acquisition**

The dataset comprises 7,753 MRI images from Qhills Technologies Pvt. Ltd., Maharashtra, and Kaggle, categorized into "Human Brain" and "Not Brain." Of these, 220 were labeled as "Not Brain," and 7,533 as "Human Brain." The "Human Brain" images were split into 6,220 for training and 1,333 for testing.

02 **Labelling**

Used the Python Pigeon Jupyter library. This involved inputting folder paths and labels into the 'annotate' function. In the image dataset, we tagged photos as 'BRAIN' or 'NOT BRAIN', following the same process for the 400-image training dataset.

03 **Pre-Processing**

Prepared image data by employing data augmentation, resizing, and normalization. Images are transformed into numerical form using Python's PIL and NumPy. Grayscale and RGB images are accommodated, ensuring consistency in size. We utilize pre-processing techniques like median blurring, bilateral filtering, and image histograms for noise reduction and contrast analysis.

04 **Model Development**

We fine-tune a VGG-16 CNN for binary 'Brain' vs. 'Not Brain' classification, optimizing it through hyperparameter tuning. The CNN includes 32 and 64-filter convolutional layers, max-pooling, dense layers with 64 and 10 output nodes, and a SoftMax final layer.

Figure 7. Phase 1: ‘Brain’ ‘Not-Brain’ Classification

### 4.1.1 Data Acquisition

The foundation of Phase 1 is to obtain a set of MRI images of various patients with different types of tumors. The first image database was collected from Qhills Technologies Pvt. Ltd., Maharashtra, and contains MRI images of the brain, spine, and gallbladder in ‘dicom.’ Adding in non-brain images helps to maintain the model’s ability to distinguish target structures from other regions of the brain. This dataset encompasses approximately 720 images with a focus on the spine, gallbladder, and brain (T1W, T2W, Flair). The three subparts of the brain, T1W, T2W, and Flair are synchronized with the visuals as highlighted in Figure 8.

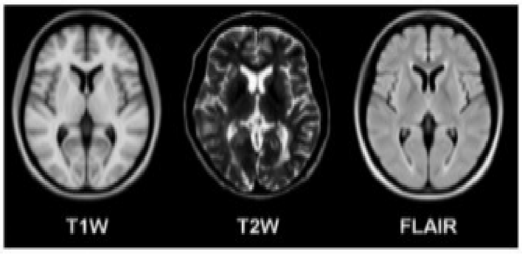


Figure 8. Magnetic Resonance Image of the brain

Similarly, to increase the reliability of the proposed model, an extra dataset namely the Brain Tumor MRI Dataset [2]. The dataset hence contained 7753 Brain MRI images, divided into three folders. Combining these two datasets allowed for a larger and more varied set of pictures for training and testing. The dataset of 7,753 MRI images were categorized into "Human Brain" and "Not brain" images. Out of these, 220 images were labeled as "Not brain," while the remaining 7,553 images were categorized as "Human Brain." The "Human Brain" images were further divided into training and testing sets, with 6,220 images allocated for training and 1,333 images for testing. The training set was comprised of 1,448 glioma images, 1,466 meningioma images, 1,584 pituitary tumor images, and 1,722 images showing no tumors, representing healthy brain tissue. Similarly, the testing set included 354 glioma images, 410 meningioma images, 354 pituitary tumor images, and 215 images of healthy brain tissue. This structured distribution of images across different tumor types and healthy cases ensured a comprehensive evaluation of the model's performance in detecting various brain abnormalities. A quick overview of the images for the training and testing stage is depicted in figure 9.

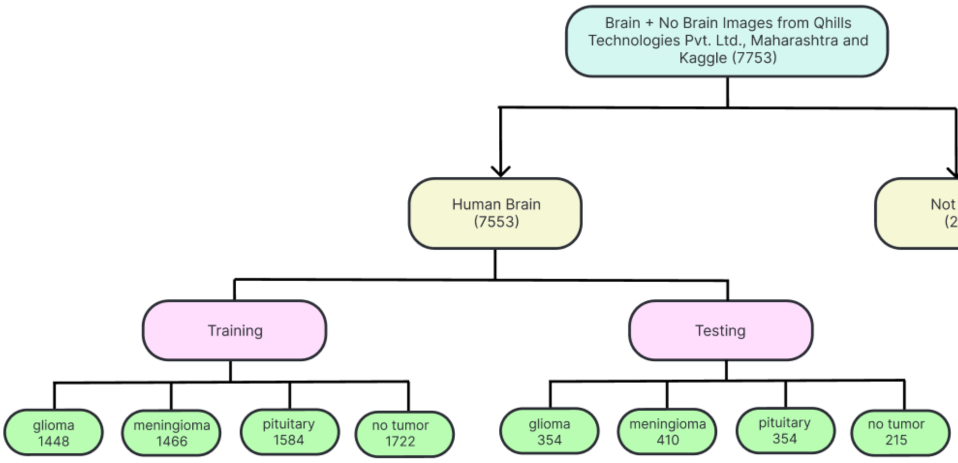
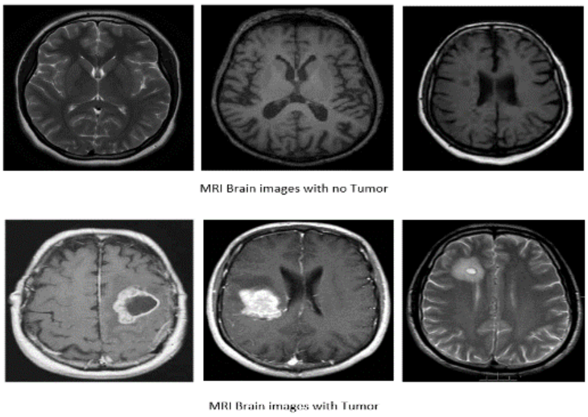


Figure 9. Images taken for the training and testing stage

T1W pictures are predominantly bright with dark ventricles. The flair has a little brighter grey section with dark ventricles while the T2W is black with bright ventricles. A specific category must have at least one other distinction made against it to be classified. T1W, T2W, and flair photos from the two categories were required for the classification (ABNORMAL and NORMAL). Radio waves and magnets (magnetic fields) are used in the MRI imaging procedure to produce images of the body's internal organs. A CT performs the same task using an X-ray. A few examples from the Brain MRI dataset with brain tumors and non-brain tumors are shown in Figure10.


MRI Brain images with no Tumor

MRI Brain images with Tumor

Figure 10. Magnetic Resonance Image of healthy brains and brains with tumor

### 4.1.2 Labelling

The Python Pigeon Jupyter library is used for labelling of images. The folder path and the labels to be applied are entered into the annotate function found under the library. Accurate labelling is fundamental to the model's learning process, enabling it to recognize and distinguish brain-related structures from other anatomical features. Every image is explored, and the user is prompted to assign the appropriate category. The 7753 photos in the dataset were tagged as BRAIN and NOT BRAIN using the annotate tool. The function returns a NumPy array at the end that contains the picture paths and labels.

### 4.1.3 Pre-Processing

Data transformations play a critical role to ensure the image data is appropriately fed into the model. In this phase, a variety of methods was used such as data augmentation, image resizing and image normalization. Data augmentation helps to bring variety to our dataset and to make it more robust, we apply some operators like rotation, scaling, and flipping to the images. These transformations enhance the stability of our model since it is exposed to different data set. It also helps eliminate images that are either too large or too small, which can make a difference when using the selected model architecture. Normalisation adds an additional layer of normality where pixel values of images fall under normalized ranges thus increasing the ease at which the model learns and converges.

The image dataset was presented as labels and image paths. As string paths or images are not understood by the categorization model, so, the transformation into numerical form is required. The Python PIL library's IMAGE function aids in opening the image. It offers file read and write operations. They are further transformed into arrays via the ASARRAY function from the NumPy library. The information utilized for both classifications was presented in various formats. It was a haphazard assortment of grayscale and RGB channel images. The photo' NumPy arrays' form is where the numerical variation can be found. The shape of a grayscale image array would be -> (256,300). This indicates that the array is two-dimensional, has 256 outer blocks, and each block has values that are 300 pixels long. The brightness of each pixel is represented by its value, which varies from 0 to 255, with zero denoting the darkest value and 255 denoting the brightest.

In the case of an RGB picture array would be-> (256,300,3). It denotes a three-dimensional array with a 256-bit outer size. The second layer is 300 pixels wide, with three pixels per block. The pixel values show the RGB dispersion (Red, Green, Blue). These pictures were available in several sizes like (256,330) or (400,1000). However, the input size must be taken into account while defining the neural network's design. The picture arrays must be resized consistently because the various sizes in each iteration may result in issues. This can be done by the "resize" function from the CV2 library or the "np. resize" function from the NumPy library. These methods can be used for processing the picture.

**a. Median blurring:** This pre-processing step takes the median of all the pixels in the kernel area and replaces the core element with the median value before applying convolution to the entire image. This method does a great job of eliminating the salt and paper noise that can be present in MRI scans.

**b. Bilateral Filtering:** This method is typically used to eliminate noise while maintaining edges. It has the following advantages over other blurring algorithms added capability of determining pixel density as defined in equation #. It has two noteworthy characteristics:

1. **Gaussian function of space**: Only closely spaced pixels are looked at. The pixels with comparable intensities are analyzed using the Gaussian function of intensity.
2. It guarantees that while retaining crisp intensity changes, only pixels with intensity values that match the core pixel are examined for blurring.

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma s}\left(||p - q||\right) G_{\sigma r}\left(|I_p - I_q|\right) I_q \qquad \text{..................} \; (1)$$

Where:

- $BF[I]_p$: This represents the filtered intensity value at pixel p after applying the bilateral filter. It's the result we want to compute.
- $W_p$: This is the normalization factor, ensuring that the filter output is properly scaled. It's computed as the sum of the spatial and range weights for pixel p.

- $\Sigma q \in S$: This signifies a summation over all pixels q in a local neighborhood S of pixel p. The neighborhood S typically includes pixels within a certain spatial distance from p.
- $G_{\sigma s}$ (||p - q||): This is the spatial Gaussian function, which depends on the spatial distance between pixel p and pixel q (||p - q||). It's a measure of how close or far the pixels are in the image.
- $G_{\sigma r}$ ($|I_p - I_q|$): This is the range Gaussian function, which depends on the difference in intensity values between pixel p and pixel q ($|I_p - I_q|$). It's a measure of how similar or different the intensity values of the pixels are.
- $I_q$: This represents the intensity value at pixel q, which is a neighbouring pixel to p. It's the value that contributes to the filtering process.

**c. Image Histograms:** The image represented in figure 11 presents a series of brain MRI scans alongside their corresponding histograms, which illustrate the distribution of pixel intensities within each scan. The MRI images, displayed on the left, represent different brain sections, while the histograms on the right show how pixel intensities are distributed across each image. The X-axis of each histogram represents pixel intensity values, ranging from 0 to 255, and the Y-axis shows the frequency of these intensities. The shape of each histogram provides insights into the image's contrast and the distribution of light and dark areas. For instance, a narrow histogram suggests low contrast, while a wider histogram indicates higher contrast, reflecting the variation in tissue densities within the brain. Together, these pairs of images and histograms are useful for analyzing image quality, tissue differentiation, and identifying potential abnormalities, such as tumors, which could be further enhanced during the pre-processing steps in a deep learning model.

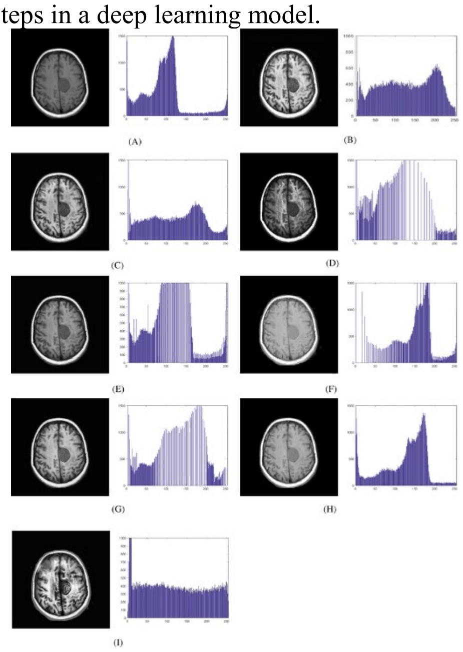


Figure 11. Comparison of brain images and histograms

### 4.1.4 Model Development

With our pre-processed and labeled dataset, a Convolutional Neural Network (CNN) model optimized for binary classification—specifically, classifying images as 'Brain' or 'Not Brain was developed and leveraged transfer learning by fine-tuning a pre-trained CNN model, such as VGG-16 on our specialized dataset. This process allows our model to inherit knowledge learned from a broader dataset, adapting it to the nuances of brain and non-brain image classification. The dataset was split into 80% training, 10% validation, and 10% testing. This split ensured a balanced approach to training and validating the model's performance. Hyperparameter tuning was conducted to optimize model performance, making sure that it can accurately distinguish between the two classes.

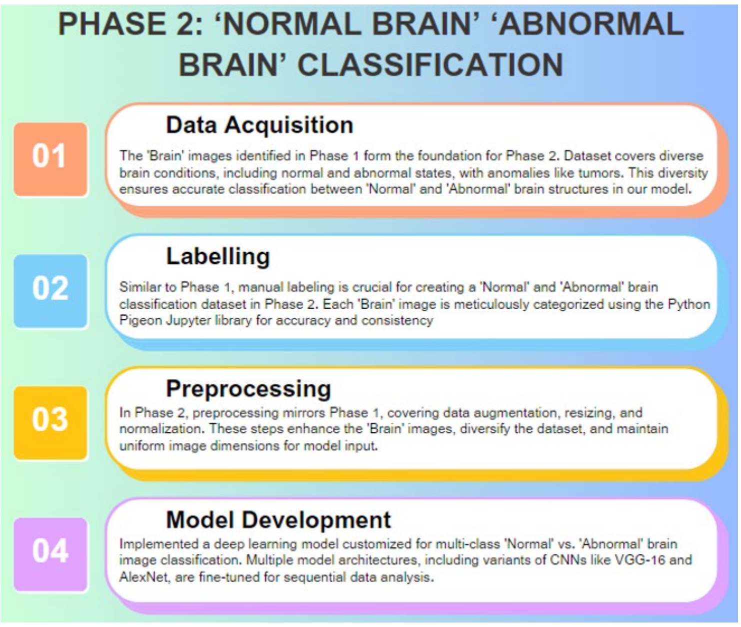
**A) CNN Model**

This model will have a convolution neural network design with dense, max pooling, and Convolutional layers. These convolutional layers are a first layer with a filter of 32, kernel size of 3*3 and Relu entry function. The next layer to be implemented will be a maximum pooling layer with a pooling size of 2*2. Second, another

Convolutional layer with 64 filters and 3*3 kernel size and Relu activation function will be used. There is then another 2 * 2 maximum pooling layer to reduce the size of the output volume an extra stage. To a fully connected dense network, the flattened feature values will be sent. After flattening, there will be two thick layers with 64 and 10 output nodes respectively or Relu activation. The last one shall be a dense layer with two outputs nodes – SoftMax activation – and the capacity to recognize if the input was a brain or not.

## 4.2 Phase 2: Abnormal-Normal Classification

Building upon the foundation laid in Phase 1, Phase 2 focuses on further classifying 'Brain' images into 'Normal' and 'Abnormal' categories, and results in the precise detection of brain abnormalities in MRI scans as shown in Figure 12.



PHASE 2: ‘NORMAL BRAIN’ ‘ABNORMAL BRAIN’ CLASSIFICATION

01 Data Acquisition

The 'Brain' images identified in Phase 1 form the foundation for Phase 2. Dataset covers diverse brain conditions, including normal and abnormal states, with anomalies like tumors. This diversity ensures accurate classification between 'Normal' and 'Abnormal' brain structures in our model.

02 Labelling

Similar to Phase 1, manual labeling is crucial for creating a 'Normal' and 'Abnormal' brain classification dataset in Phase 2. Each 'Brain' image is meticulously categorized using the Python Pigeon Jupyter library for accuracy and consistency

03 Preprocessing

In Phase 2, preprocessing mirrors Phase 1, covering data augmentation, resizing, and normalization. These steps enhance the 'Brain' images, diversify the dataset, and maintain uniform image dimensions for model input.

04 Model Development

Implemented a deep learning model customized for multi-class 'Normal' vs. 'Abnormal' brain image classification. Multiple model architectures, including variants of CNNs like VGG-16 and AlexNet, are fine-tuned for sequential data analysis.

Figure 12. Phase 2: ‘NORMAL Brain’ ‘ABNORMAL Brain’ Classification

### 4.2.1 Data Acquisition

The 'Brain' images identified in Phase 1 serve as the basis for Phase 2. These images encompass diverse brain conditions, including both normal and abnormal states. Our dataset includes a wide range of anomalies such as tumors, lesions, and other pathological conditions, which ensures that our model can accurately differentiate between the 'Normal' and 'Abnormal' brain structures.

### 4.2.2 Labelling

Just as in phase 1, labeling is an important component in the subsequent formation of a labeled dataset for classification between ‘Normal’ and ‘Abnormal’ brains. All the ‘Brain’ images have the ground truth as each of the images is well sorted according to the condition it possesses. At least, this is what the Python Pigeon Jupyter library does to maintain accuracy and consistency in this labelling process.

### 4.2.3 Preprocessing

Preprocessing in Phase 2 is almost similar to what was done in Phase 1; data augmentation, resizing, and min-max normalization. These steps pre-process the ‘Brain’ images for model input, expand the plurality and verity of the dataset, and regulate the size of the images.

### 4.2.4 Model Development

In Phase 2, a deep learning model with high accuracy in multi-class classification algorithms for distinguishing among the types of brain tumor images was used. The following model types are analysed and adjusted, including several kinds of CNNs such as VGG-16, and AlexNet for sequential data processing. The latter remains important for increasing the model’s efficiency for identifying minor pathologies in MRI scans.

### A) CNN Model

Convolution, pooling, dropout, and thick layers make up the architecture of a convolutional neural network. The first two layers will both be convolutional layers with 32 and 64 filters, a kernel size of 3*3, and Relu as activation, respectively. In the third layer, max pooling with a pool size of 2*2 will be used, and the fourth layer will be a dropout layer with 25% of the total nodes deleted. A convolutional layer with 64 filters, a 3*3 kernel size, and Relu activation will be the fifth layer. A maximum pooling layer with a pool size of 2*2 will be the sixth layer, and a dropout layer with a 20% dropout rate will be the seventh layer.

After that Another Convolutional layer with 128 filters, 3*3 kernel size of Relu activation should be included next. Next, there is a dropout of 25% implemented and there is a Max Pooling layer of size 2*2 as well. To transmit each of the individual values as an independent node in a highly connected neural network, all the matrices will go through a flattened layer. Following flattening, a dense layer with 64 output nodes and Relu activation will be the next layer, which will again be followed by a dropout layer with a 20% dropout rate. Due to the architecture's prediction of two classes, the final layer will be another dense layer with a SoftMax activation of 2 output nodes.

### B) VGG16

The model is built on top of VGG16, which is a pre-trained convolutional neural network (CNN) for image classification. First, the VGG16 model is loaded and the input shape is set to match the size of the images in the dataset, which is 224 x 224 pixels. The 'include_top' parameter is set to False, which means that the final fully connected layers of VGG16 that perform the classification will not be included. The weights parameter is set to 'imagenet' which means that the model will be pre-trained with a dataset of 1.4 million images called imagenet

Next, the for layer in the base model. layers: loop is used to set all layers of the base model (VGG16) to non-trainable so that the weights of these layers will not be updated during training. Then, the last three layers of the VGG16 model are set to trainable. After that, a Sequential model is created and the VGG16 model is added to it.

Next, a Flatten layer is added to the model which reshapes the output of the VGG16 model from a 3D tensor to a 1D tensor, so that it can be processed by the next layers of the model. Then, a Dropout layer is added which is used to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training time.

After that, a dense layer is added with 128 neurons, and relu activation function is added. Next, another Dropout layer is added. Finally, the output dense layer is added with a number of neurons equal to the number of units equal to 2, and the 'SoftMax' activation function is added. The 'SoftMax' activation function is used to give a probability distribution over the possible classes.

### C) AlexNet

Eight layers make up the AlexNet architecture, including three FC levels and five Convolutional layers. Due to the 60 million parameters in this architecture, overlapping is a problem that is avoided. Methods applied in this model

**I. Relu Nonlinearity:** Relu is used in place of the previously popular tanh function. It has been demonstrated that when utilizing the CIFAR-10 dataset for training, CNN models using Relu activation can achieve a 25% error rate six times faster than CNN models using tanh.

**II. Overlapping Pooling:** CNNs typically combine the outputs of nearby linked neurons. However, adding overlap resulted in a 0.5 percent decrease in error, indicating that overlapping pooling models are harder to overfit.

**III. Standardization (Local Response Normalization):** Only a portion of the image was standardized, as opposed to the complete image. The overall effectiveness of the model is improved by this method.

The following equation determines the output image size for the next layer:

$$n_{out} = \left\lceil \frac{n_{in}+2p-k}{s} \right\rceil + 1 \qquad \text{.......} (2)$$

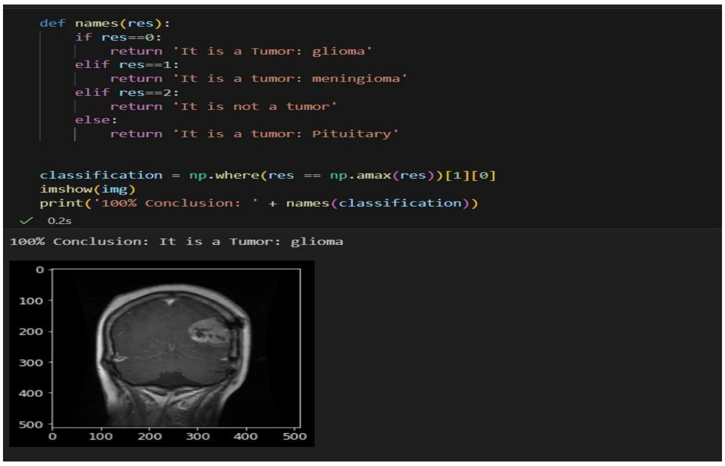
Where:

- $n_{in}$: Number of input features
- $n_{out}$: Number of output features
- k: Convolutional kernel size
- p: Convolutional padding size
- s: Convolutional stride size

Hence in phase 2, The model was trained to make predictions on the test set, comprising the test_paths and test_labels. The datagen() function was employed to generate batches of images and corresponding labels. For each batch, the model.predict() method was utilized to make predictions on the images. The predicted labels were initially encoded and decoded using the decode_label() function, with the decoded values stored in y_pred. The actual labels were stored in y_true. To provide a visual representation of the progress, the tqdm library was used to display a progress bar during the execution of the loop.

To represent the training history of the model, there is a use of a graph that shows the accuracy and the loss of the model in different epochs. The horizontal axis denotes the number of epochs while the vertical axis denotes the accuracy and loss for each of the epochs. This graph is generated using the matplotlib library and comprises two lines: It has one key for accuracy and another one for loss. Thus, by analyzing such a plot, one can get an idea about the model's learning dynamics and achievements during the training process. Thus, it is a good means of checking whether a model is overfitting or underfitting, that is, whether it is more specialized to the training data set or is not fitting enough to the data.

In figure 13, the model's functioning is depicted, showing its ability to detect the presence of a tumor and its type. By leveraging its underlying algorithms and trained patterns, the model is capable of discerning whether a tumor exists or not, and if it does, it can further classify it into specific types based on the provided data. This visualization provides insights into the operational mechanism of the model in tumor detection and classification tasks.



```
def names(res):
    if res==0:
        return 'It is a Tumor: glioma'
    elif res==1:
        return 'It is a tumor: meningioma'
    elif res==2:
        return 'It is not a tumor'
    else:
        return 'It is a tumor: Pituitary'

classification = np.where(res == np.amax(res))[1][0]
imshow(img)
print('100% Conclusion: ' + names(classification))
```

✓ 0.2s

```
100% Conclusion: It is a Tumor: glioma
```

Figure 13. Working of the model with an example

## 5 Result and Analysis

The authors had to analyze using classification measures because identifying brain abnormalities is a classification task. A confusion matrix is typically the first step in categorization evaluations. The confusion matrix for all the models (CNN, AlexNet, VGG16) is shown in Figures 14,15,16 respectively. This visualization reflects the performance of the models on the dataset, showing the number of correct and incorrect classifications for each class (Glioma, Meningioma, Pituitary, and No Tumor).

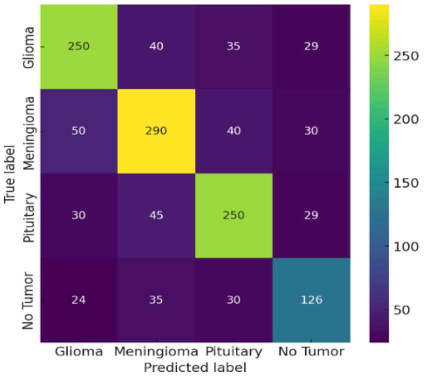


Figure. 14. Confusion Matrix for CNN

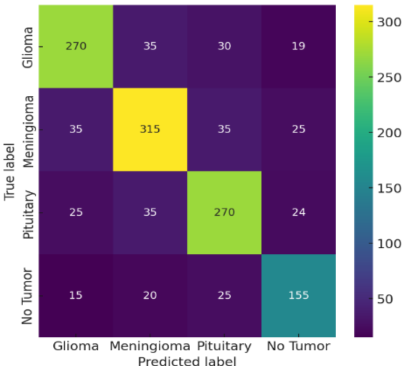


Figure. 15. Confusion Matrix for AlexNet

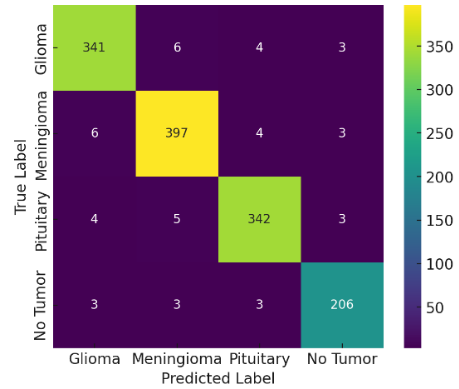


Figure 16. Confusion Matrix for VGG16

Most values, as seen by the confusion matrix above, fall into the genuine positive. The majority of projected values typically tend to be close to the actual value, according to a true positive percentage. This suggests that our VGG16 model is effective. This assertion is supported by the fact that true negatives make up the second-highest majority, indicating that the majority of values that are genuinely "not abnormal" are also projected to be "not abnormal."

Some more evaluation indicators after developing the confusion matrix:

1. **Accuracy Score:** The accuracy score is a measure of the fraction of samples that were predicted correctly. It is given by:

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

2. **Precision Score:** The precision score calculates the percentage of positive results that were anticipated to be positive.

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

3. **Recall Score:** Recall score is a measurement of the proportion of positive outcomes that the model accurately predicted:

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

Where;

- ***TP (True Positive):*** *The number of instances where the model correctly predicts the positive class (i.e., the model predicts "positive" and the actual class is "positive").*
- ***TN (True Negative):*** *The number of instances where the model correctly predicts the negative class (i.e., the model predicts "negative" and the actual class is "negative").*
- ***FP (False Positive):*** *The number of instances where the model incorrectly predicts the positive class (i.e., the model predicts "positive" but the actual class is "negative"). This is also known as a Type I error.*
- ***FN (False Negative):*** *The number of instances where the model incorrectly predicts the negative class (i.e., the model predicts "negative" but the actual class is "positive"). This is also known as a Type II error.*

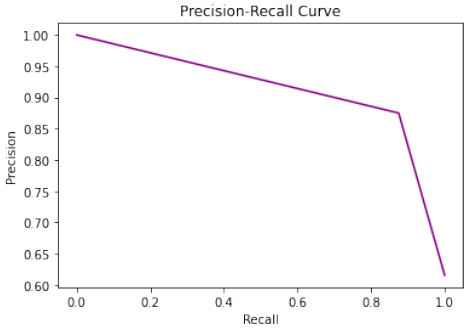


Figure 17. Precision-Recall Curve of Phase 1

In this instance, recall scores are far greater than the precision ratings as shown in Figure 17. These data indicate that our model is producing the majority of the pertinent outcomes and forecasts. The authors used a ROC curve to map the false positive rate with the genuine positive rate in order to make a more logical comparison.

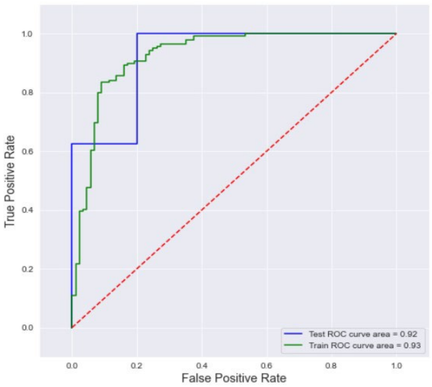


Figure 18. ROC Curve of the Phase 1

It can be seen in Figure 18 that the genuine positive rate rises exponentially over time but the false positive rate doesn't significantly increase over time.

4. **F1 Score:** This metric provides the average of recall and precision scores, with higher values indicating a more accurate model

$$F1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

5. **Jaccard Score:** The Jaccard score provides insight into the degree of similarity between the predicted and observed datasets. The formula used to calculate a Jaccard score is as follows

$$Jaccard\ Score = \frac{A \cap B}{A \cup B}$$

*Where*

- *A represents the set of actual positive instances (i.e., the true positive cases where the actual label is positive).*
- *B represents the set of predicted positive instances (i.e., the cases where the model predicts a positive label).*

6. **Hamming Loss:** Hamming loss counts the labels that were predicted inaccurately across the dataset. All evaluation indicators for all the models are given below in the Table 3:

Table 3: Evaluation Indicators in the proposed methodology

| | **Accuracy Score** | **Precision** | **Recall Score** | **F1_score** | **Hamming Loss** | **Jaccard Score** |
|---|---|---|---|---|---|---|
| **CNN** | 0.731 | 0.687 | 0.686 | 0.683 | 0.312 | 0.521 |
| **VGG16** | 0.964 | 0.964 | 0.951 | 0.962 | 0.0353 | 0.931 |
| **AlexNet** | 0.769 | 0.758 | 0.757 | 0.757 | 0.242 | 0.610 |

Multiple models have to be deployed and trained to assess performance and select the best fit for our task along with the pre-processing using data augmentation to enhance the brain scans to broaden the range of training data and training techniques. An ANN has a few dense layers and uses the activation functions like Relu and Sigmoid. The model evaluation after training and testing showed that the ANN model had an accuracy of 53.8%. A loss of 46.2% suggested inefficiency. Next, CNN was implemented, where we could execute convolution and pooling more quickly than with an ANN. The accuracy of the CNN model across our training and testing data was 73.1% and loss was 31.28%. Then, VGG16 was implemented which has an already-built architecture that extracts features on a need basis that provided us with an accuracy of 96.4% and a loss of 3.53%. then at last, the AlexNet model was used which also has a pre-build model and gave us an accuracy of 76.9% and a loss of 24.23%. All in all, the VGG16-based model had the highest accuracy, scoring 94.4% as shown in Table 4.

Table 4: Different Models and their accuracies and losses in the proposed methodology

| | **ACCURACY** | **LOSS** |
|---|---|---|
| **CNN** | 73.1% | 31.28% |
| **VGG16** | 96.4% | 3.53% |
| **AlexNet** | 76.9% | 24.23% |

The culmination of our research presents compelling outcomes, showcasing the potential of our deep learning-driven brain tumor detection methodology. Leveraging diverse models including CNN, VGG16, and AlexNet, a hyperparameter analysis to optimize the performance of our approach was conducted. Through a comprehensive array of performance metrics such as Accuracy, Precision, Recall, F1-score, Confusion Matrix, RMSE, MAE, and MSE, our model's effectiveness was evaluated. The results underscore the model's ability to discern abnormalities in brain MRI images, demonstrating its superiority over traditional methods.

Relatively to the research phases, the material comprises a contrast of the results of the research with other authors' findings, as depicted in table 5. This also makes it possible to compare the outcomes and efficiencies of the suggested approach with the related results and studies. The table proves very useful in this where the user gets the difference and summary of the findings within the shortest time possible. Thus, the inclusion of this comparison increases the reliability of the current study and the accumulated understanding of the topic.

Table 5: Comparison between proposed work and existing literature

| **S.NO** | **Author** | **Models Used** | **Accuracy** | **Paper Name** |
|---|---|---|---|---|
| **1** | PGokila Brindha et al. | ANN & CNN | ANN (80.77%)<br>CNN (89%) | Brain tumor detection from MRI images using deep learning techniques [12] |
| **2** | Anushka Singh1 et al. | VGG-16 | VGG-16 (93%) | Brain Tumor Classification Using CNN And Vgg16 Model [10] |
| **3** | P Gayathri et al. | VGG-16 | VGG-16 (94%) | Exploring the Potential of VGG-16 Architecture for Accurate Brain Tumor Detection Using Deep Learning [11] |
| **4** | Mehrotra et al. | CNN | CNN (94.8%) | A transfer learning approach for AI-based classification of brain tumors [16] |
| **5** | Mzoughi et al. | CNN | CNN (96.49%) | Deep Multi-scale 3D CNN for Brain Tumor Grading from Volumetric MRI Images [17] |

| | | | | |
|---|---|---|---|---|
| **6** | S Chatterjee e al. | CNN | CNN (96.5%) | Classification of Brain Tumours in MR Images Using Deep Statistical CNNs [18] |
| **7** | Proposed Model | VGG-16 | VGG-16 (96.4) | An Efficient Deep Learning Technique for Brain Abnormality Detection Using MRI Images (Proposed Model) |

In our case, our complication of distinguishing between a normal state of the brain and an abnormal one has been resolved successfully. The authors were able to design a model that had the classification capability integrated into it with superior accuracy and performance levels built on the VGG 16 structure. The preprocessing methods allowed us to label data properly and perform data augmentation that provided the model with the necessary properties. The highest accuracy recorded was of the VGG16 model which is 96.4%.

# 6 Challenges and Issues Identified

**6.1 Image Quality:** MRI scans can be affected by motion artifacts, image distortions, and low signal-to-noise ratio, which can create problems in tumor identification. These factors affect the clarity and accuracy of the images, which makes it difficult to differentiate between different tumor types.

**6.2 Variability in Imaging Protocols:** Based on imaging protocols, including different sequences and parameters, MRI scans can vary. These variations result in differences in image appearance and can affect the interpretation and comparability of MRI images, which creates challenges in tumor identification and classification across different cases and healthcare institutions.

**6.3 Computational Challenges:** With reference to tumor identification and its classifications, processing and analyzing big MRI datasets demands lots of computational power. This data is complex, and for training deep learning models like VGG-16 on this kind of data we would need quite some computational power which may constrain the hardware available or even more specifically computing infrastructure itself. These computational challenges could be overcome using efficient processing techniques and hardware acceleration.

# 7 Future Work

Further work in this area can concentrate on various facets which augment the proposed model and make it relevant for clinical application.

Firstly, we could use more images to train our model (expand the dataset) which would already improve performance. The model will therefore be trained on a broader dataset of medical imaging, providing greater numbers of the types of tumors and representations to allow it to create more out-of-reach scenarios. As the authors themselves pointed out, in addition to those things, and avoid biasing towards common tumors or easier-to-separate-treat images by including an equal representation of tumor types within their dataset.

Secondly, exploring the potential of transfer learning can be beneficial. Instead of starting from scratch, using pre-trained models on a large-scale dataset, such as ImageNet, and fine-tuning them specifically for tumor detection can enhance the model's learning process and will improve its performance. This approach can also help in the integration of new imaging modalities or adaptability to different clinical settings.

Additionally, optimizing pre-processing steps, labeling techniques, and data augmentation can also help to improve the model's robustness and performance. We must preprocess data like normalization, image registration, and noise reduction carefully to have high-quality input into the model. Furthermore, advanced data augmentation methods like geometric transformations can help create even more of the dataset diversity required to handle changes in tumor appearance.

To evaluate the generalizability and performance of this model in real-world clinical practice, it is crucial to validate on different datasets with independent characteristics. Also, it may bolster reproducibility and reduce bias by collaborating with healthcare institutions for capturing data across various imagery devices, clinical protocols, or populations.

Finally, integrating the model into an existing clinical workflow and assessing its influence on patient outcomes and clinical decisions is essential. Future prospective clinical studies employing real-time comparisons with human experts could illustrate this model's constructive practical utility. Addressing these areas for future work will serve in the continued optimization, validation, and clinical integration of this model to improve the early detection of tumors and reliable diagnoses, leading to improved patient outcomes on all fronts.

# 8 Conclusion

In the proposed model, the authors were able to successfully distinguish between a normal brain and an abnormal brain. In this paper, the authors built a system with classification capabilities with excellent accuracy and performance using a VGG16-based model. Advanced pre-processing methods, labeling, and data augmentation have also been used to provide the model with the necessary properties. The accuracy of the final model was 96.4%

This was followed by the introduction of a model designed on the VGG16 architecture specifically for imaging medical data to detect different types of tumors. The model was trained across a variety of images including glioma, meningioma, and pituitary adenomas. The model showed a great ability to correctly discriminate different tumor types based on our study results, demonstrating the potential of using this approach in clinical cancer detection and diagnosis. Theoretically, with more refinement of the proposed classifiers followed by validation and integration into clinical workflows (which itself takes time), this model can be a useful tool for early detection of these tumors which would translate to better patient outcomes; whereby potentially lives could be saved.

Thus, this paper can enhance the knowledge in the areas of medical imaging and deep learning since it explores the success of the VGG-16 model in diagnosing brain tumors from MRI scans as well as exploring the model’s capability in diagnosing other diseases. The outcomes can greatly help healthcare learners and practitioners regarding the performance of deep learning algorithms for diagnosing various diseases. Since the proposed model tends to provide better identification of diseases, better patient outcomes can be expected. The proposed work can assist in better utilization of resources available in the healthcare systems. This paper also supports the necessity of new investigations and advancements in medical imaging and deep learning to enhance disease identification and intervention in clinical practice.

# 9 References

[1] Roy, S., & Bandyopadhyay, S. K. (2012). Detection and Quantification of Brain Tumor from MRI of Brain and it’s Symmetric Analysis. International Journal of Information and Communication Technology Research, 2(6).

[2] https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset/data

[3] Ayadi, W., Elhamzi, W., Charfi, I., & Atri, M. (2021). Deep CNN for brain tumor classification. Neural processing letters, 53, 671-700.

[4] Santhi, P., & Anuaparna, P. (2021). Brain Tumor Segmentation and Classification Using Deep Learning Algorithm. Annals of the Romanian Society for Cell Biology, 16243-16250.

[5] Khan, M. A., Ashraf, I., Alhaisoni, M., Damaševičius, R., Scherer, R., Rehman, A., & Bukhari, S. A. C. (2020). Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists. Diagnostics, 10(8), 565.

[6] Li, Z. C., Yan, J., Zhang, S., Liang, C., Lv, X., Zou, Y., ... & Chen, Y. (2022). Glioma survival prediction from whole-brain MRI without tumor segmentation using deep attention network: a multicenter study. European Radiology, 32(8), 5719-5729.

[7] Liu, Z., Tong, L., Chen, L., Jiang, Z., Zhou, F., Zhang, Q., ... & Zhou, H. (2023). Deep learning based brain tumor segmentation: a survey. Complex & intelligent systems, 9(1), 1001-1026.

[8] Fang, F., Yao, Y., Zhou, T., Xie, G., & Lu, J. (2021). Self-supervised multi-modal hybrid fusion network for

brain tumor segmentation. IEEE Journal of Biomedical and Health Informatics, 26(11), 5310-5320.

[9] Younis, A., Qiang, L., Nyatega, C. O., Adamu, M. J., & Kawuwa, H. B. (2022). Brain tumor analysis using deep learning and VGG-16 ensembling learning approaches. Applied Sciences, 12(14), 7282.

[10] Singh, A., Deshmukh, R., Jha, R., Shahare, N., Verma, S., & Nilawar, A. (2020). Brain tumor classification using CNN and VGG16 model. Int J Adv Res Innov Ideas Educ, 6(2), 1331-1336.

[11] Gayathri, P., Dhavileswarapu, A., Ibrahim, S., Paul, R., & Gupta, R. (2023). Exploring the Potential of VGG-16 Architecture for Accurate Brain Tumor Detection Using Deep Learning. Journal of Computers, Mechanical and Management, 2(2), 23056-23056.

[12] Brindha, P. G., Kavinraj, M., Manivasakam, P., & Prasanth, P. (2021, February). Brain tumor detection from MRI images using deep learning techniques. In IOP conference series: materials science and engineering (Vol. 1055, No. 1, p. 012115). IOP Publishing.

[13] Chattopadhyay, A., & Maitra, M. (2022). MRI-based brain tumour image detection using CNN based deep learning method. Neuroscience informatics, 2(4), 100060.

[14] Kapadnis, A. (2021). Brain Tumor Detection using Transfer Learning Technique with AlexNet and CNN (Doctoral dissertation, Dublin, National College of Ireland).

[15] Khan, M. A., Ashraf, I., Alhaisoni, M., Damaševičius, R., Scherer, R., Rehman, A., & Bukhari, S. A. C. (2020). Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists. Diagnostics, 10(8), 565.

[16] Mehrotra, R., Ansari, M. A., Agrawal, R., & Anand, R. S. (2020). A transfer learning approach for AI-based classification of brain tumors. Machine Learning with Applications, 2, 100003.

[17] Mzoughi, H., Njeh, I., Wali, A., Slima, M. B., BenHamida, A., Mhiri, C., & Mahfoudhe, K. B. (2020). Deep multi-scale 3D convolutional neural network (CNN) for MRI gliomas brain tumor classification. Journal of Digital Imaging, 33, 903-915.

[18] Chatterjee, S., Nizamani, F. A., Nürnberger, A., & Speck, O. (2022). Classification of brain tumours in MR images using deep spatiospatial models. Scientific Reports, 12(1), 1505.