Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Evaluation of Deep Convolutional Neural Network Architectures for Brain Tumor Classification

Evaluación de Arquitecturas de Redes Neuronales Convolucionales para la clasificación de tumores cerebrales

Dorcelus, S.[a], Del Razo-López, F.[a], Alejo-Eleuterio, R.[a], Villanueva-Vásquez, D.[a], Granda-Gutiérrez, E.E.[b]*

[a] División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México / Instituto Tecnológico de Toluca, CP 52149, Metepec, México (https://orcid.org/0009-0007-7877-1672, https://orcid.org/0000-0002-5241-5457, https://orcid.org/0000-0002-7580-3305, https://orcid.org/0000-0001-8512-7630).

[b] Centro Universitario UAEM Atlacomulco, Universidad Autónoma del Estado de México, CP 50450, Atlacomulco, México (https://orcid.org/0000-0002-9316-9627).

mm24280359@toluca.tecnm.mx; fdelrazol@toluca.tecnm.mx; ralejoe@toluca.tecnm.mx; daniel.vv@toluca.tecnm.mx; eegrandag@uaemex.mx*

Technological innovation: The main contribution of this work lies in the use of convolutional neural networks for the identification of brain tumors in magnetic resonance imaging.

Industrial application area: The results of this work can potentially be used in the training and calibration processes of magnetic resonance imaging instruments.

Received: December 2, 2025
Accepted: April 16, 2026

Resumen

La inteligencia artificial está ayudando cada vez más a los médicos a analizar imágenes médicas, como las resonancias magnéticas (MRI, por sus siglas en inglés), para detectar afecciones como tumores cerebrales. Sin embargo, no todos los modelos de inteligencia artificial funcionan igual de bien, y elegir el adecuado es crucial. En este estudio comparamos tres modelos ampliamente utilizados de aprendizaje profundo (VGG16, ResNet50 e InceptionV3) para determinar cuál ofrece el mejor rendimiento en la clasificación de tumores cerebrales a partir de imágenes de resonancia magnética. El dataset consta de 7023 imágenes MRI (4 clases: glioma, meningioma, pituitary, sin tumor) con división 70/15/15 para entrenamiento/validación/prueba. Al evaluar los tres modelos bajo las mismas condiciones, encontramos que InceptionV3 superó claramente a los demás, alcanzando un 94% de exactitud y obteniendo sistemáticamente mejores resultados en múltiples métricas de desempeño. Su diseño único, que captura detalles de la imagen a distintas escalas de forma simultánea, resulta especialmente adecuado para identificar patrones complejos presentes en los tumores cerebrales. Si bien VGG16 constituye una alternativa más sencilla con resultados

aceptables cuando los recursos computacionales son limitados, ResNet50, a pesar de su popularidad en otras áreas, mostró un rendimiento inferior en esta tarea específica. Nuestros hallazgos ofrecen una orientación práctica y basada en evidencia para investigadores y equipos de salud que buscan aplicar herramientas de inteligencia artificial en neuroimagen, destacando a InceptionV3 como la mejor opción para la clasificación de tumores cerebrales.

Palabras clave: Inteligencia artificial, Clasificación de tumores cerebrales, Redes convolucionales, Redes neuronales profundas.

Abstract

Artificial intelligence is increasingly helping doctors analyze medical images, such as magnetic resonance imaging (MRI) scans, to detect conditions such as brain tumors. However, not all AI models work equally well, and selecting an appropriate model is essential. In this study, we compare three widely used deep learning models (VGG16, ResNet50, and InceptionV3) to determine which performs best in classifying brain tumors from MRI images. Under identical evaluation conditions for all three models, we found that InceptionV3 clearly outperformed the others, achieving 94% accuracy and consistently higher scores across multiple performance metrics. Its unique design, which captures image details at multiple scales simultaneously, appears especially well-suited for the complex patterns seen in brain tumors. While VGG16 offers a simpler alternative with acceptable results when computing resources are limited, ResNet50, despite its popularity in other fields, showed weaker performance in this specific task. Our findings offer practical, evidence-based guidance for researchers and healthcare teams looking to apply AI tools in neuroimaging, highlighting InceptionV3 as the preferred choice for brain tumor classification.

Keywords: Artificial intelligence, Brain tumor classification, Convolutional networks, Deep neural networks.

1. Introduction

The healthcare sector has experienced a profound transformation in recent years, driven significantly by rapid technological advancements, particularly within Artificial Intelligence (AI) [1, 2]. AI is widely recognized as a key driver in this evolution, showing significant potential to improve global health outcomes [3].

Cancer remains a leading cause of mortality worldwide [4, 5]. Brain tumors, specifically, represent a critical threat due to their location within the central nervous system and inherent diagnostic complexities [6, 7]. Therefore, early and precise detection is paramount for effective treatment planning and improved prognoses [8].

In response, deep learning for automated medical image analysis has seen substantial adoption [9]. These techniques often surpass traditional methods in diagnostic accuracy, especially for complex modalities such as MRI (Magnetic Resonance Imaging) [10]. The past decade has witnessed remarkable evolution in Convolutional Neural Network (CNN) architectures, with significant performance gains from foundational models (e.g., LeNet [11]) to state-of-the-art designs employing neural architecture search and attention mechanisms (e.g., EfficientNet

[12], ConvNeXt [13]).

This study evaluates three established CNN models for brain tumor classification: VGG16 [14], InceptionV3 [15], and ResNet50 [16]. After identifying the best architecture, its design principles will be examined to guide future development of efficient diagnostic tools. Although newer architectures are available, these three remain common benchmarks and are often referenced in review papers, performance analyses, and as baselines in new research.

2. Related Works

The rise of machine learning has catalyzed a revolution across numerous sectors, including industry, sports, and automotive technology. Its impact on the medical field has been particularly profound, providing clinicians with advanced tools to process complex patient data (from symptomatic histories to visual diagnostics) to aid in accurate diagnosis. This section explores how machine learning is used in neuroimaging, reviewing key and recent studies on brain tumor detection and classification. It traces the development of techniques, from feature-based models to deep learning architectures, and examines their performance metrics.

Zacharaki et al. [17] pioneered a computer-aided diagnosis (CAD) system for grading gliomas, the most common primary brain tumors. Their methodology went beyond simple binary classification (high-grade vs. low-grade) to distinguish between different glioma grades. They extracted morphological, intensity-based, and texture features from MRI scans. These features were then fed into two classifiers: a Support Vector Machine (SVM) with a radial basis function (RBF) kernel and a k-Nearest Neighbors (KNN) algorithm. Their system achieved 85% accuracy, demonstrating the potential of machine learning to differentiate complex tumor grades.

Huang et al. (2012) [18] focused on a Content-Based Image Retrieval (CBIR) system using a Bag-of-Visual-Words (BoVW) model tailored to specific regions. Their innovation lay in using two types of information: the absolute intensity of the lesion core and the relative intensity change at the lesion's periphery and surrounding environment. They also emphasized that preprocessing steps such as intensity normalization and whitening were critical for boosting retrieval performance. Their model achieved a high average accuracy of 91.0%. The study's major limitation, noted by the authors themselves, was the reliance on manual segmentation for tumor delineation in the MRIs. This process is time-consuming, subjective, and introduces bias. They also observed that the model's performance degraded when the initial segmentation contour was inaccurate, as it failed to capture the full intensity variation of the tumor.

Yang et al. (2012) [19] also developed a CBIR framework for navigating large image databases. Their system treated a query tumor region as an input to find other tumors with analogous pathological properties (a useful tool for comparative diagnosis). A novel feature, the Margin Information Descriptor (MID), was introduced to characterize tumors by encoding information about their surrounding tissue, which is often crucial for diagnosis. The proposed system reached an accuracy of 89.3%. However, its main drawback was the use of a globally linear distance metric, which was insufficient for capturing complex, non-linear relationships in local regions of the image. Like Huang's work, this system was also built upon manually segmented tumor data.

Everingham et al. (2015) [20] marked a shift towards deeper architectures. They applied Convolutional Neural Networks (CNNs) to the task, utilizing the publicly available Figshare dataset. A key aspect of their preprocessing was an adaptive spatial division algorithm designed to automatically identify and amplify tumor regions as “regions of interest” (ROIs). These ROIs were further partitioned to extract robust features, including intensity histograms and textural information from the Gray-Level Co-Occurrence Matrix (GLCM). They also employed a Bag-of-Words (BoW) model for feature encoding. This multi-faceted approach yielded impressive accuracy rates of 87.54%, 89.72%, and 91.28% on their different test configurations, significantly outperforming earlier methods.

Ismael et al. (2018) [21] conducted experiments to detect and classify three prevalent tumor types: meningioma, glioma, and pituitary tumors. Their work highlighted the challenges of class imbalance and complex feature extraction, as evidenced by the significantly varying accuracy rates they reported: 45% for meningioma, and 15% for both glioma and pituitary tumors. These results underscore the difficulty of the task with the methods and dataset used at the time.

Shakeel et al. (2019) [22] addressed the challenge of feature extraction by applying advanced mathematical techniques, including fractional and multifractal dimensions. These methods are adept at quantifying the complex, irregular texture and structure of tumors in MRI scans. They paired these powerful features with a machine learning classifier trained with backpropagation, reporting substantial improvements in detection performance compared to traditional feature extraction methods.

Deepak and Ameer (2019) [23] focused explicitly on classifying the three main tumor types: glioma, meningioma, and pituitary. Their main contribution was leveraging transfer learning. Instead of training a model from scratch, they fine-tuned the pre-trained GoogleNet architecture, a powerful CNN model initially trained on millions of natural images (ImageNet). This approach allowed them to benefit from the robust feature detectors already present in GoogleNet, adapting them for the medical imaging domain.

Sultan et al. (2019) [24] conducted a comprehensive study using two public datasets. They employed a two-model deep learning approach: the first model classified tumors into the three major types, while a second, more specialized model graded gliomas into Grade II, III, or IV—a critical distinction for treatment planning. Their first model achieved an excellent 96.13% accuracy, and the second, a 16-layer custom CNN, achieved a remarkable 98.7% accuracy on the glioma grading task. This high performance was achieved even on a relatively small dataset of Contrast-Enhanced MRI (CE-MRI) images, demonstrating the efficiency of their architecture.

Sajjad et al. (2019) [25] performed an extensive comparative analysis of several leading pre-trained deep learning models: VGG16, VGG19, InceptionV3, ResNet50, InceptionResNetV2, and Xception. They then proposed a novel, hybrid methodology that used a fine-tuned InceptionV3 network as a feature extractor. The deep features from this model were used to train a Support Vector Machine (SVM) classifier, creating a powerful CNN-SVM ensemble. This strategy effectively combined the strengths of deep feature extraction and

robust classification, yielding a top-tier accuracy of 96.97% for multi-grade brain tumor classification.

Tiwari et al. (2022) [26] developed a sophisticated CNN-based model for four-class classification: glioma, meningioma, pituitary tumor, and “no tumor”. Their model architecture was likely designed to overcome challenges like overfitting and feature selection. The result was a state-of-the-art accuracy of 99%, showcasing the potential of well-designed CNNs to achieve near-perfect classification on a balanced and well-preprocessed dataset.

Srinivasan et al. (2024) [27] tackled an even more complex five-class classification problem, adding “metastatic” tumors (which originate from cancers elsewhere in the body) to the standard classes, including "normal”. They systematically optimized three different CNN models using a grid search technique to find the best hyperparameters. The best-performing model among them achieved a robust accuracy of 93.81%, a commendable result given the increased difficulty of distinguishing between five classes.

Hossain et al. (2023) [28] conducted a comparative analysis of modern deep learning architectures, primarily focusing on Vision Transformers (ViTs) like ViT-B/16 and ViT-L/16, alongside CNNs like EfficientNet. They proposed a novel ensemble model named EVoM that combined the top-performing Vision Transformer (ViT-L/16) with the top-performing CNN (EfficientNetB3) to create a robust feature extractor. This hybrid strategy effectively leveraged the complementary strengths of both architectural types, yielding a top-tier accuracy of 96.94% for multiclass brain tumor classification. Furthermore, they utilized Explainable AI (XAI) techniques to provide visual explanations for the model's predictions.

The selection of models for this study (VGG16, ResNet50, Inception V3) was influenced by their proven efficacy in related work, such as the brain tumor classification study by Pillai et al. [29]. It is noteworthy, however, that our findings did not align perfectly. Contrary to their result identifying VGG16 (91.58% accuracy) as optimal, our comprehensive evaluation found that the Inception V3 model achieved the highest classification accuracy on our data. This variance in optimal model performance motivates a detailed examination of the core architectural features of VGG16, ResNet50, and Inception V3. Understanding these design principles is crucial for hypothesizing why each model might respond differently to specific dataset characteristics.

3. Materials and Methods

To contextualize these results, it is essential to understand the fundamental design choices that differentiate these models.

3.1.Network Architecture

This work employs three distinct supervised learning models. Supervised learning aims to infer a mapping from inputs to outputs using a labeled dataset, where xi denotes an input instance and yi its associated class label. The training objective is to find the model parameters θ that minimize a loss function $\ell(y,\hat{y})$, which quantifies the discrepancy between the label and the model's prediction by measuring the error between the true label y and the predicted output $\hat{y} = f\theta(x)$.

One of the most prominent supervised methods is the neural network. A neural network is made up of interconnected layers of processing units (neurons). The

basic function of a single neuron is to compute a weighted sum of its inputs, add a bias term, and then pass the result through a non-linear activation function:

$$a = \sigma(w^T x + b)$$

For a single neuron, and for a multi-layer network, this operation is stacked across multiple layers. The complete network function $f_\theta(x)$ that maps inputs to predictions ŷ can be defined recursively. For a network with L layers:

$$a^{(0)}=x$$
$$z^{(l)}=W^{(l)}a^{(l-1)}+b^{(l)}$$
$$\text{for } l=1,2,\ldots,L$$
$$a^{(l)}=\sigma^{(l)}(z^{(l)}) \quad \text{for } l=1,2,\ldots,L$$
$$\hat{y}=f_\theta(x)=a^{(L)}$$

Between the input and output layers are the hidden layers, which are considered deep, hence the term deep learning.

The final layer is typically a fully connected layer, meaning that every neuron in the previous layer is connected to every output neuron (a *feedforward* network). The parameters θ of the entire network, including these final weights and biases, are most commonly learned via maximum likelihood estimation, optimized with stochastic gradient descent (SGD) or its variants [30]. In SGD, parameter updates are computed using small, randomly sampled subsets of the data (mini-batches), making training feasible for large datasets by providing a noisy but efficient estimate of the true gradient [31].

The functional role of this final fully connected layer is to project the integrated feature vector into interpretable logits for each class. These logits are then passed through a softmax function to produce a valid probability distribution over the possible classes, P(y | x;θ) [32]. For a multiclass problem with C classes, the softmax function is defined as:

$$P(y \mid x;\theta) = softmax(f(x;\theta)) = \frac{\exp([f(x;\theta)]_k)}{\sum_{j=1}^{C}\exp([f(x;\theta)]_j)}$$

Where $f(x;\theta)_k$ is the logit or score for class k, computed by the model.

The training objective of maximizing likelihood is equivalent to minimizing the negative log-likelihood, which manifests as the binary cross-entropy loss for two-class problems and the categorical cross-entropy loss for multiclass tasks. For a dataset of n examples, this results in the following minimization objective, known as the cross-entropy loss:

$$\ell(\theta) = -\frac{1}{n}\sum_{i=1}^{N}\log P(y_i \mid x_i;\theta)$$

Convolutional Neural Networks (CNN) were also built on this optimization framework, which employs a fundamentally different connectivity pattern specifically designed for spatial data. In CNNs, weights are shared across the entire input space via convolutional operations, where the same filter is applied to all local regions of an image. This architectural approach yields two critical advantages: first, it ensures translation equivariance, meaning the network can detect features regardless of their position without requiring separate detectors for each location; second, it achieves dramatic parameter efficiency by decoupling the number of learnable weights from the input image dimensions. As [33] explains in *Inside Deep Learning*, CNNs emerge as a structurally more efficient solution, particularly for spatial inputs such as images, by inherently encoding geometric priors into the network architecture.

3.2.Visual Geometric Group (VGG16)

Renowned for its systematic and efficient

design, the VGG16 network [14], as presented in Table 1, processes 224×224 RGB images using exclusively 3×3 convolutional filters with a 1-pixel stride and padding to preserve spatial resolution. It is organized into blocks of 2-3 convolutions, all ReLU-activated, each followed by a 2×2 stride-2 max-pooling layer to progressively reduce feature dimensions hierarchically. In total, VGG16 comprises 13 convolutional layers and 3 max-pooling layers, for a total of 16 parameterized layers that give the architecture its name. All convolutional layers employ ReLU (Rectified Linear Unit) activation functions to introduce non-linearity and accelerate training.

Following the convolutional stage, an adaptive average pooling layer reduces feature maps to a fixed 7×7 dimension, enabling efficient transition to the fully connected stage. The network concludes with three fully connected layers: the first two contain 4,096 neurons each, while the final layer outputs scores for the 4 target classes, followed by a softmax activation to produce class probabilities.

Table 1. VGG Architecture.

Layer	Type	Parameters	Output	Activation
1–2	Conv2D	3 × 3 × 64, pad=1	224 × 224 × 64	ReLU
3	MaxPool	2 × 2, stride=2	112 × 112 × 64	–
4–5	Conv2D	3 × 3 × 128, pad=1	112 × 112 × 128	ReLU
6	MaxPool	2 × 2, stride=2	56 × 56 × 128	–
7–9	Conv2D	3 × 3 × 256, pad=1	56 × 56 × 256	ReLU
10	MaxPool	2 × 2, stride=2	28 × 28 × 256	–
11–13	Conv2D	3 × 3 × 512, pad=1	28 × 28 × 512	ReLU
14	MaxPool	2 × 2, stride=2	14 × 14 × 512	–
15–17	Conv2D	3 × 3 × 512, pad=1	14 × 14 × 512	ReLU
18	MaxPool	2 × 2, stride=2	7 × 7 × 512	–
19	Flatten	–	7 × 7 × 512 = 25088	–
20	Dense	25088 × 4096	4096	ReLU
21	Dense	4096 × 4096	4096	ReLU
22	Dense	4096 × 4	4	Softmax

3.3.Residual Network

While VGG16 demonstrated the power of deep convolutional architectures, its substantial computational requirements and training challenges highlighted the need for more efficient deep networks. As network depth increases further, conventional architectures like VGG16 encounter the degradation problem, where accuracy saturates and then rapidly degrades due to optimization difficulties in very deep networks.

This limitation motivated the development of ResNet50, which introduced a groundbreaking solution through residual learning. Unlike traditional sequential networks, ResNet50 [16] incorporates skip connections that bypass one or more layers, enabling the network to learn residual functions rather than complete transformations. These identity shortcuts facilitate gradient flow during backpropagation, effectively addressing the vanishing gradient problem and enabling stable training of networks with 50 or more layers, and also allow ResNet50 to achieve superior performance with significantly fewer parameters than VGG16, while maintaining training stability through its innovative bypass connections, as summarized in Table 2.

Table 2. ResNet Architecture.

Layer	Type	Parameters	Output	Activation
3		$7 \times 7 \times 64, s = 2, pad = 3$ $3 \times 3, s = 2$	$112 \times 112 \times 64$ $56 \times 56 \times 64$	
	Bloque Residual 1 (conv2 x)	Conv2D: 1 × 1 × 64 Conv2D: 3 × 3 × 64,*s* = 1 Conv2D: 1 × 1 × 256 x3 bloques, Shortcut: 1 × 1 × 256,*s* = 1	56 × 56 × 256	ReLU
4	Bloque Residual 2 (conv3 x)	Conv2D: 1 × 1 × 128 Conv2D: 3 × 3 × 128,*s* = 2 Conv2D: 1 × 1 × 512 x4 bloques, Shortcut: 1 × 1 × 512,*s* = 2	28 × 28 × 512	ReLU
5	Bloque Residual 3 (conv4 x)	Conv2D: 1 × 1 × 256 Conv2D: 3 × 3 × 256,*s* = 2 Conv2D: 1 × 1 × 1024 x6 bloques, Shortcut: 1 × 1 × 1024,*s* = 2	14 × 14 × 1024	ReLU
6	Bloque Residual 4 (conv5 x)	Conv2D: 1 × 1 × 512 Conv2D: 3 × 3 × 512,*s* = 2 Conv2D: 1 × 1 × 2048 x3 bloques, Shortcut: 1 × 1 × 2048,*s* = 2	7 × 7 × 2048	ReLU
7	GlobalAvgPool	7 × 7	1 × 1 × 2048	-
8	Dense	2048 × 4	4	Softmax

3.4.InceptionV3

While ResNet50 successfully addressed the challenges of training very deep networks through residual connections, another parallel innovation emerged focusing on multi-scale feature extraction and computational efficiency within each layer. This led to the development of the Inception architecture, with InceptionV3 representing one of its most advanced and refined iterations. InceptionV3 is a convolutional neural network architecture developed by Google, introduced in the seminal paper "Rethinking the Inception Architecture for Computer Vision" [15]. Unlike the sequential layer stacking of VGG16 or the residual blocks of ResNet50, InceptionV3 introduces a fundamentally different approach through its inception modules. These modules perform parallel convolutions with multiple filter sizes (1×1, 3×3, 5×5) within the same layer, capturing features at different scales simultaneously. This architectural paradigm, represented in Table 3, enables the network to efficiently model both local and global patterns without requiring excessively deep stacking.

Table 3. InceptionV3 Structure.

Layer	Type/Parameters	Output	Act.
1	Conv2D: 3 × 3 × 32, s=2	149 × 149 × 32	ReLU
2	Conv2D: 3 × 3 × 32	147 × 147 × 32	ReLU
3	Conv2D: 3 × 3 × 64, pad=1	147 × 147 × 64	ReLU
4	MaxPool: 3 × 3, s=2	73 × 73 × 64	-
5	Conv2D: 1 × 1 × 80	73 × 73 × 80	ReLU
6	Conv2D: 3 × 3 × 192	71 × 71 × 192	ReLU
7	MaxPool: 3 × 3, s=2	35 × 35 × 192	-

8	Inception-A Branch 1: 1 × 1 × 64 Branch 2: 1 × 1 × 48 → 5 × 5 × 64 Branch 3: 1 × 1 × 64 → 3 × 3 × 96 → 3 × 3 × 96 Branch 4: 3 × 3 avg-pool → 1 × 1 × 32	35 × 35 × 288	ReLU
9	Inception-B Branch 1: 3 × 3 × 384, s=2 Branch 2: 1 × 1 × 64 → 3 × 3 × 96 → 3 × 3 × 96, s=2 Branch 3: MaxPool 3 × 3, s=2	17 × 17 × 768	ReLU
10	Inception-C Branch 1: 1 × 1 × 192 Branch 2: 1 × 1 × 128 → 1 × 7 × 128 → 7 × 1 × 192 Branch 3: 1 × 1 × 128 → 7 × 1 × 128 → 1 × 7 × 128 → 7 × 1 × 128 → 1 × 7 × 192 Branch 4: 3 × 3 avg-pool → 1 × 1 × 192	17 × 17 × 1280	ReLU
11	Reduction Branch 1: 1 × 1 × 192 → 3 × 3 × 320, s=2 Branch 2: 1 × 1 × 192 → 1 × 7 × 192 → 7 × 1 × 192 → 3 × 3 × 192, s=2 Branch 3: MaxPool 3 × 3, s=2	8 × 8 × 2048	ReLU
12	AvgPool: 8 × 8	1 × 1 × 2048	-
13	Dense: 2048 × 4	4	Softmax

Furthermore, InceptionV3 incorporates several key innovations, including factorized convolutions, which replace large filters with smaller asymmetric convolutions (like 1×3 and 3×1 instead of 3×3), significantly reducing computational complexity while maintaining representational power. The network also employs auxiliary classifiers and sophisticated regularization techniques that enhance gradient flow and prevent overfitting within its deep architecture.

3.5.Supervised training

In this paper, the training of the mentioned models (ResNet/VGG16/InceptionV3) follows an iterative, supervised learning process. All experiments were conducted in a controlled computational environment: Python 3.11.5, PyTorch 1.10.0, torchvision 0.11.0, running Ubuntu 24.04 LTS. The experiments were conducted on a system equipped with an Intel Core i7-8565U CPU (1.80 GHz) and 8 GB of RAM, without a dedicated NVIDIA GPU. These resource limitations constrained the feasible training duration, leading to a maximum of 10 epochs. Despite the modest hardware, the loss curves stabilized within this range, indicating sufficient convergence for the purposes of this study.

The models were initialized with ImageNet pre-trained weights provided by torchvision. Their original classifier heads were replaced with a new fully connected layer of 4 units (one per class). During training, all layers except the new head were initially frozen (feature extractor mode). After the first 5 epochs, the last three convolutional blocks of each model were unfrozen and fine-tuned with a learning rate of 1e-4, while the head was trained with the base learning rate $\eta = 0.001$.

Given a dataset, where xi represents input images and yi ∈ {meningioma, pituitary, glioma, no tumor} their corresponding true class labels, each model was trained for 10 epochs (E = 10) with a batch size B = 256.

To calculate the batches per epoch, it is assumed that each batch Bj contains B samples: 1,2,...,⌈N/B⌉, where for each batch Bj, the model predictions ŷ are generated by the forward pass:

$$\hat{y} = f_\theta(x)$$

Where fθ denotes the model architecture (e.g., ResNet, VGG16, or InceptionV3), and θ represents the model's trainable parameters.

The optimization objective uses multiclass cross-entropy loss. The loss for a batch Bj is computed as:

$$\ell_j(\theta) = \frac{1}{B}\sum_{k=1}^{B} \ell_{CE}(f_\theta(x_k^{(j)}), y_k^{(j)})$$

Where y_i is the one-hot encoded representation of the true label.

The model parameters are updated using gradient descent, implemented via backpropagation:

$$\theta_{\text{new}} = \theta - \eta \nabla_\theta \ell_j(\theta)$$

Where:

- η is the learning rate (lr)
- $\nabla_\theta \ell_j(\theta)$ is the gradient of the loss with respect to the parameters, computed automatically with loss.backward()
- The parameter update is performed by the optimizer with optimizer.step()

For this 4-class classification problem, the models were optimized using the Adam (Adaptive Moment Estimation) algorithm. The training configuration included a Multiclass Cross-Entropy Loss function, Adam optimizer, and a learning rate: η ∈ R+ (η = 0.001). This learning rate was chosen after a preliminary sweep (0.0001, 0.001, 0.01) showed best validation accuracy at 0.001, which is also the default recommended value for Adam. Due to those hardware constraints previously mentioned, the batch size was set to 256 as a compromise between memory capacity and gradient stability; smaller batches (64, 128) led to slower convergence in early tests while larger batches caused out-of-memory errors. The total number of epochs (10) was determined by early stopping, as validation accuracy showed no further improvement after 8–10 epochs in initial experiments, and further training was not feasible given the computational limitations.

3.6.Dataset

The experimental analysis in this work is based on a dataset of 7,023 high-quality brain Magnetic Resonance Imaging (MRI) scans, curated by Msoud Nickparvar from three public repositories (Figshare, SARTAJ, and Br35H), hosted on Kaggle [34]. This amalgamation ensures diverse representation of tumor types and imaging variations. The dataset comprises samples evenly distributed across four diagnostic classes: glioma, meningioma, no tumor, and pituitary. This intrinsic class balance (approximately 1,755 images per class) eliminates concerns regarding external imbalance, thereby promoting stable training dynamics and reducing the risk of model bias. The balanced distribution of the dataset is presented in Table 4 and Figure 1. Each image is accompanied by ground-truth labels, facilitating a supervised learning framework.

Table 4. Dataset Statistics.

	Training Set	Validation Set	Test Set
Total Samples	4,569	1,143	1,311
Number of Classes	4	4	4
Glioma	1,058 (23.16%)	263 (23.01%)	300 (22.88%)
Meningioma	1,059 (23.18%)	280 (24.50%)	306 (23.34%)
No Tumor	1,291 (28.26%)	304 (26.60%)	405 (30.89%)
Pituitary	1,161 (25.41%)	296 (25.90%)	300 (22.88%)

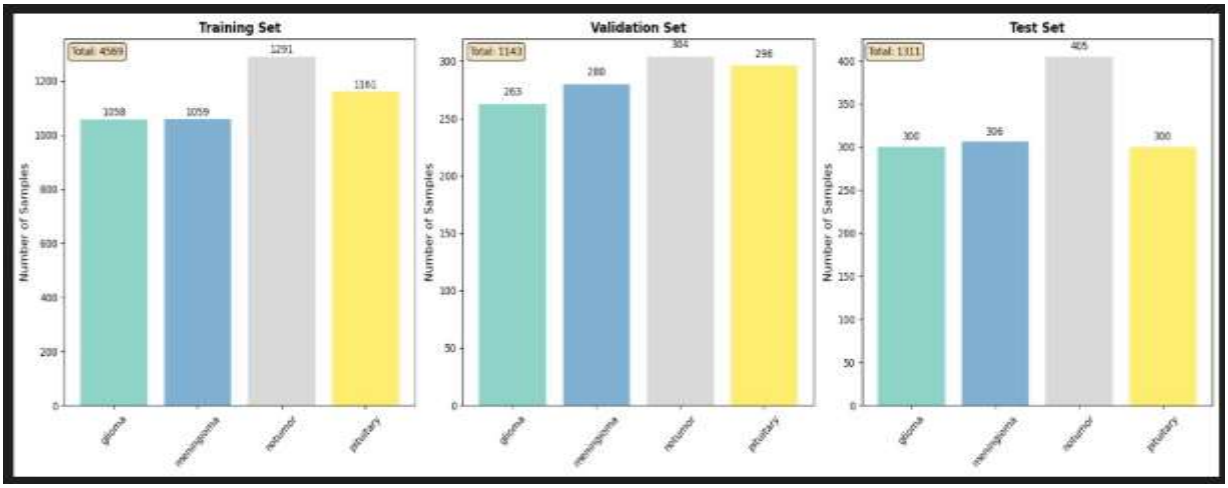


Figure 1. Dataset distribution visualization.

Prior to training, the dataset underwent a comprehensive preprocessing routine designed to avoid data leakage: all dataset splits (training, validation, and testing) were performed before any transformation was applied. Each image was resized to a fixed resolution of 224×224 pixels to conform to the input requirements of the chosen network architecture. No additional intensity normalization was applied; pixel values were only scaled to the [0,1] range by dividing by 255. To further reinforce the model's generalization and combat overfitting, a suite of simple data augmentation operations was employed on the training split. These included random horizontal flips, random rotations (±10°), and slight zoom (0.9–1.1). All augmentations were implemented using PyTorch's torchvision.transforms with a fixed random seed to ensure reproducibility.

4. Results and discussion

To assess the training dynamics and generalization capability of each model, we plotted its learning curves. Each figure comprises two graphs: (a) the training and validation loss per epoch, and (b) the corresponding training and validation accuracy. A well-generalized model is characterized by a decreasing loss for both sets, with the validation metrics closely following the training metrics.

For the VGG16 model, the validation loss closely follows the training loss, decreasing from about 0.58 to 0.32 over 8 epochs, while validation accuracy increases from 78% to 85%. The lack of divergence between these curves indicates good generalization and no significant overfitting during the training period

shown, suggesting that training could continue for additional epochs to further improve performance. Despite the apparent convergence of training and validation metrics in Figure 2, the model achieved a test accuracy of only 84.21%, suggesting that the validation set may not have fully captured the true test distribution.

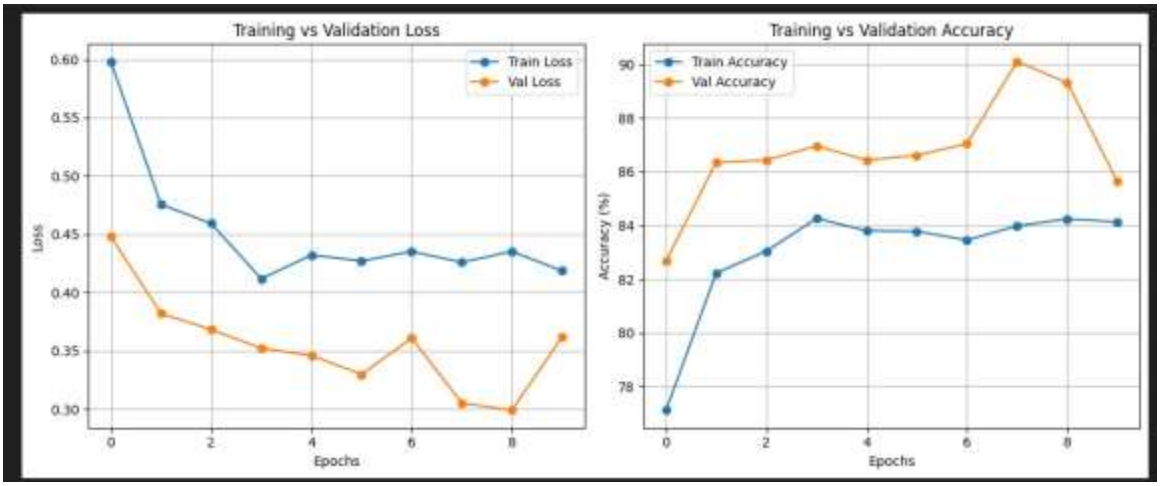


Figure 2. training/validation graphs of VGG16 model.

For the ResNet50 model, Figure 3 illustrates the model's path to its final performance. Both training and validation metrics improved steadily, with validation accuracy stabilizing around epoch 6 at the level that would be confirmed on the test set. While a slight gap emerges between training and validation performance (indicating minor overfitting), the validation accuracy plateaus at approximately 82%, which closely matches the final test performance of 82.21%.

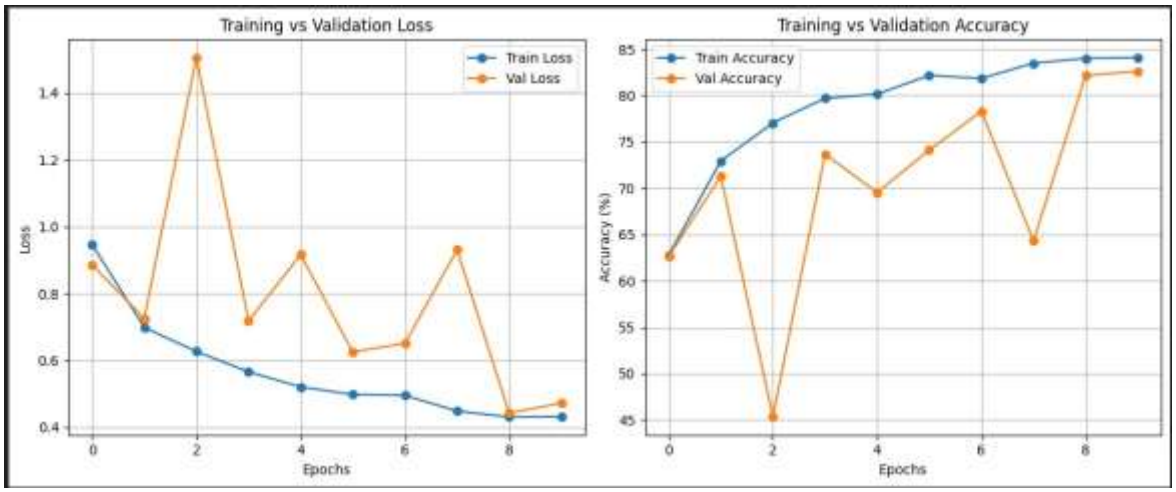


Figure 3. Training/validation graphs of ResNet50 model.

This alignment, coupled with a balanced F1-score of 82.31%, indicates that the model learned robust features without much overfitting to the validation distribution. The results are statistically reliable, based on a substantial test set, and suggest that the model has reached its performance ceiling given the current architecture and data. While this represents a solid baseline, the 17.85% error rate

indicates room for improvement, particularly in reducing false negatives, which could be critical in medical diagnostic applications.

For the InceptionV3 model, Figure 4 illustrates ideal learning behavior, characterized by converging training and validation curves that plateau at high performance levels. The validation accuracy closely tracks the training accuracy, improving from 84% in previous training to 92% over 8 epochs, with no signs of divergence. This pattern indicates robust feature learning and strong generalization capability, confirming the effectiveness of our modeling approach. The nearly identical training and validation trajectories predicted strong generalization, which was confirmed by a test accuracy of 94%, closely matching the final validation performance.

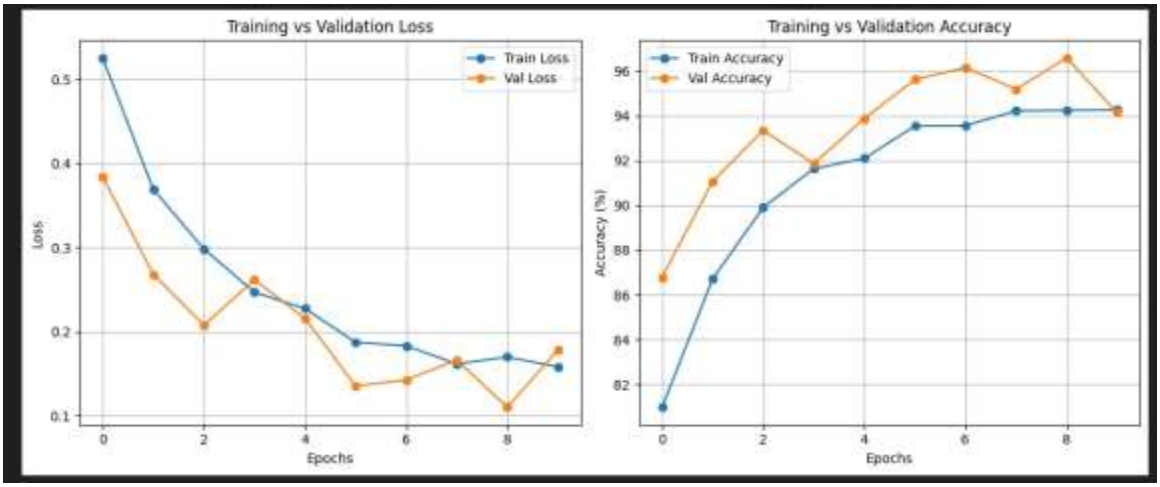


Figure 4. training/validation graphs of InceptionV3 model.

The models were evaluated from multiple perspectives to ensure a comprehensive assessment of their performance, with a particular focus on their applicability in medical diagnostics. Per-class metrics (Recall, F1-score, Support, CI[1]) are presented in Table 5, while global metrics (Accuracy, Macro-average F1, Weighted-average F1) are summarized in Table 6.

Recall (Sensitivity): This measured the model's ability to correctly identify all relevant cases of a class. A high recall rate was vital in medicine, as a False Negative (FN) could mean a true tumor was missed, potentially delaying life-saving treatment.

$$R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}$$

F1-Score: The harmonic mean of Precision and Recall, which constitutes a single metric that balances the trade-off between the two. This was especially informative for our multiclass, imbalanced dataset, as it provided a more holistic view of per-class performance than accuracy alone.

$$F1_c = 2 \cdot \frac{P_c \cdot R_c}{P_c + R_c}$$

Support: The number of actual occurrences of the class in the test dataset.

1 Confidence Intervals for Reported Test Accuracies: we compute the 95% confidence interval for accuracy using the normal approximation (Wald interval) or Wilson interval.

It indicated the statistical basis for the calculated metrics for that class.

$$S_c = \sum_{i=1}^{N} I(y_i = c)$$

Table 5. Performance of models with their per-metrics evaluation.

Model	Accuracy	95% CI(Wald)	Recall	F1-score	Support	Validation Accuracy
VGG16	84.21%	82.14-86.28	84.21%	83.18%	1311	85.65%
ResNet50	82.21%	77.28-83.68	79.48%	82.31%	1311	80.84%
InceptionV3	94%	92.66-95.34	94.36%	93.02%	1311	94.14%

Additionally, to evaluate the model's overall performance, the following aggregated metrics were used:

Accuracy: This was the overall proportion of correct predictions (including both true positives and true negatives) across all classes. While this metric can be misleading in imbalanced datasets, the balanced nature of our dataset (≈1,755 images per class) mitigates this concern, making accuracy a valid measure for model comparison.

$$\text{Accuracy} = \frac{\sum_{c=1}^{C} TP_c}{N}$$

Macro-Average F1: This metric computed the F1-score for each class independently and then took the arithmetic mean. It treated all classes equally, regardless of their support. It was the preferred metric when all classes were of equal clinical importance, as it penalized poor performance on minority classes.

$$\text{Macro-}F1 = \frac{1}{C}\sum_{c=1}^{C} F1_c$$

Weighted-Average F1: This metric calculates the average F1 Score, weighted by each class's support. It was appropriate because the class distribution in the test set reflected real-world prevalence, and the clinical cost of an error was considered proportional to the frequency of the class.

$$\text{Weighted-}F1 = \sum_{c=1}^{C} F1_c \cdot \frac{S_c}{N}$$

Table 6. Global performance metrics evaluation.

Model	Accuracy	Macro average	Weighted average
VGG16	84.21%	83.92%	84.56%
ResNet50	82.21%	82.93%	83.66%
InceptionV3	94.00%	93.68%	93.94%

The choice of the primary evaluation metric was guided by clinical and operational priorities. To prioritize the correct identification of all tumor cases (minimizing life-threatening false negatives), Recall was paramount. To prevent patient distress and avoid unnecessary procedures (minimizing costly false positives), precision, which we complement with the F1-score, was key. For a balanced view of a model's performance on each class, the Macro-Average F1 was the most robust metric, as it ensured that performance on rare tumor types was not overshadowed by common ones. The Weighted-Average F1 was used as it reflected the real-world prevalence of the conditions.

Interestingly, our comparative evaluation of VGG16, ResNet50, and InceptionV3 reveals a clear performance hierarchy. As shown in Tables 5 and 6, InceptionV3 achieved the highest accuracy (94%), macro-average F1 (93.68%), and weighted-

average F1 (93.94%), outperforming VGG16 and ResNet50 across all metrics.

This finding aligns with the results of Hossain et al. (2023), whose systematic evaluation of six pre-trained CNNs showed that InceptionV3 was consistently among the top-performing individual models. In their study, the ensemble IVX16—which combined VGG16, InceptionV3, and Xception—further improved performance, but the strong individual results of InceptionV3 corroborate our observation that its multi-scale feature extraction capabilities are particularly effective for brain tumor MRI classification.

The variation in optimal model performance across different studies, presented in Table 7, reinforces the concept that there is no universally "best" architecture. Instead, model selection must be guided by context-specific evaluation and validation. This highlights the importance of conducting systematic architecture comparisons tailored to particular dataset characteristics and clinical requirements, rather than relying on generalized performance claims from the literature.

Moreover, the comparative evaluation of three previous deep learning models VGG16, ResNet50, and InceptionV3, reveals a clear performance hierarchy, indicating that InceptionV3 is the most effective model for this specific task, delivering robust and well-balanced performance.

Table 7. Review of Related Works.

Method /Year	Brain Tumor type	Method Used	Features	Classifier	Overall Accuracy
Zacharaki et al. (2009)	Glioma metastasis; high/low grade	Conventional + perfusion MRI; SVM with recursive feature elimination	Selected discriminative features	SVM	85% (metastasis) / 88% (grading)
Yang et al. (2012)	Brain tumors	CBIR with Margin Information Descriptor (MID)	MID	CBIR	89.3%
Huang et al. (2012)	Brain tumors	CBIR with BoVW incorporating spatial information	Bank of visual words (BoVW)	CBIR	91.0%
Ismael et al. (2018)	Meningioma, glioma, pituitary	DWT + Gabor filters for feature extraction	Wavelet and Gabor features	Not specified	91%
Shakeel et al. (2019)	Brain tumors	Fractional/multifractional dimensions; MLBPNN	Essential features within tumor area	MLBPNN	Not specified
Deepak & Ameer (2019)	Glioma, meningioma, pituitary	Transfer learning (GoogleNet / Inception-v1)	Deep features from ImageNet	Fine-tuned GoogleNet	High
Sultan et al. (2019)	Glioma grades (II, III, IV); multi-class tumors	Custom 16-layer CNN architecture	Custom deep features	16-layer CNN	96.13% (multi-class) / 98.7% (grading)
Hossain et al. (2023)	Multi-class brain tumors	Evaluation of 6 CNNs; IVX16 ensemble	Features from VGG16, VGG19, InceptionV3, ResNet50, InceptionResNetV2, Xception	IVX16 ensemble	96.94%
This Study (Current)	Multi-class tumors	Architecture Comparison	Deep features from pre-trained models	VGG16, ResNet50, Inception V3	Highest: Inception V3

5. Conclusion

This study provides a clear and systematic comparison of three widely adopted deep learning architectures (VGG16, ResNet50, and InceptionV3), in the specific task of classifying brain tumors from MRI scans. Rather than treating these models as opaque systems, we examined how their inherent design choices influence performance in medical imaging. Our results reveal a consistent and significant advantage for InceptionV3, which achieved 94% accuracy and a macro F1-score of 93.68%, substantially outperforming both VGG16 (84.21%) and ResNet50 (82.21%).

The strength of InceptionV3 lies in its ability to process image features at multiple scales simultaneously, which aligns well with the varied and complex appearances of brain tumors on MRI. This makes it accurate and reliable across different cases.

For practitioners and researchers, our findings offer practical guidance: InceptionV3 should be the architecture of choice for brain tumor classification in clinical or research settings where performance is a priority. When computational resources are limited, and a slight drop in accuracy is acceptable, VGG16 remains a reasonable, simpler alternative. In contrast, ResNet50, despite its success in many computer vision tasks, did not demonstrate a comparable advantage in this neuroimaging context, likely due to the unique challenges posed by medical data.

Despite the quality and balance of the dataset used, some limitations must be acknowledged:

- **Exclusive use of public data**: Although the dataset has been curated from public repositories (Figshare, SARTAJ, Br35H) and is available on Kaggle, it does not include images from real clinical environments with diverse equipment, protocols, or populations not represented in those sources. This may limit the model's ability to generalize to real-world data with different acquisition characteristics.
- **Lack of external clinical validation:** The training (4,569), validation (1,143), and test (1,311) sets are derived from the same public dataset split. No independent validation has been performed with external data, which is essential to assess the robustness and clinical applicability of the model.

To address these limitations, future research directions, such as testing in simulated clinical environments, or generalization to other tumor types and variants or collecting and evaluating the model on datasets from multiple clinical institutions, with different MRI scanner manufacturers (GE, Siemens, Philips) and acquisition parameters are highly recommended.

Ultimately, this work surpasses generic model suggestions by providing evidence-based, task-specific insights that can guide better decisions in developing and deploying AI tools for brain tumor diagnosis and analysis.

References

[1] Thomas H. Davenport. *The AI Advantage: How to Put the Artificial Intelligence Revolution to Work*. MIT Press, 2019.

[2] Eric Topol. *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, 2019.

[3] World Health Organization. Ethics and governance of artificial intelligence for health. Technical report, WHO, 2021.

[4] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, et al. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 71(3):209–249, 2021.

[5] World Health Organization. Cancer. Technical report, WHO, 2021.

[6] Quinn T. Ostrom et al. Cbtrus statistical report: Primary brain and other central nervous system tumors diagnosed in the united states in 2013–2017. *Neuro-Oncology*, 22:iv1–iv96, 2020.

[7] David N. Louis, Hiroko Ohgaki, Otmar D. Wiestler, and Webster K. Cavenee. *WHO Classification of Tumours of the Central Nervous System*. IARC Press, 5th edition, 2021.

[8] Patrick Y. Wen and Santosh Kesari. Malignant gliomas in adults. *New England Journal of Medicine*, 359(5):492–507, 2010.

[9] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436– 444, 2015.

[11] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner.´Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

[13] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] E Zacharaki, S Wang, S Chawla, Soo Yoo, D Wolf, R Melhem, E Davatzikos, and C Davatzikos. Classification of brain tumor type and grade using mri texture and shape in a machine learning scheme. *Magn Reson Med*, 62(6):1609–1618, 12 2009.

[18] M Huang, W Yang, M Yu, Z Lu, Q Feng, and W Chen. Retrieval of brain tumors with regionspecific bag-of-visual-words representations in contrast-enhanced mri images. *Computational and mathematical methods in medicine*, 2012, 2012.

[19] W Yang, Q Feng, M Yu, Z Lu, Y Gao, Y Xu, and W Chen. Content-based retrieval of brain tumor in contrast-enhanced mri images using tumor

margin information and learned distance metric. *Medical physics*, 39(11):6929–6942, 2012.

[20] M Everingham, S Eslami, L Van Gool, C Williams, J Winn, and A Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis*, 111:98–136, 2015.

[21] M Ismael and I Abdel-Qader. Brain tumor classification via statistical features and backpropagation neural network. In *Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT)*, Rochester, MI, USA; NJ, USA, 5 2018. Piscataway.

[22] P Shakeel, T Tobely, H Al-Feel, G Manogaran, and S Baskar. Neural network based brain tumor detection using wireless infrared imaging sensor. *IEEE Access*, 7:5577–5588, 2019.

[23] S Deepak and P Ameer. Brain tumor classification using deep cnn features via transfer learning. *Computers in biology and medicine*, 111:103345, 2019.

[24] H Sultan, N Salem, and W Al-Atabany. Multi-classification of brain tumor images using deep neural network. *IEEE Access*, 7:69215–69225, 2019.

[25] Muhammad Sajjad, Salman Khan, Khan Muhammad, Wanqing Wu, Amin Ullah, and Sung Wook Baik. Multi-grade brain tumor classification using deep cnn with extensive data augmentation. *Journal of Computational Science*, 30:174–182, January 2019.

[26] Pallavi Tiwari, Bhaskar Pant, Mahmoud M. Elarabawy, Mohammed Abd-Elnaby, Noor Mohd, Gaurav Dhiman, and Subhash Sharma. Cnn based multiclass brain tumor detection using medical imaging. *Computational Intelligence and Neuroscience*, 2022:1–8, June 2022.

[27] Saravanan Srinivasan, Divya Francis, Sandeep Kumar Mathivanan, Hariharan Rajadurai, Basu Dev Shivahare, and Mohd Asif Shah. A hybrid deep cnn model for brain tumor image multi-classification. *BMC Medical Imaging*, 24(1), January 2024.

[28] Shahriar Hossain, Amitabha Chakrabarty, Thippa Reddy Gadekallu, Mamoun Alazab, and Md. Jalil Piran. Vision transformers, ensemble model, and transfer learning leveraging explainable ai for brain tumor detection and classification. *IEEE Journal of Biomedical and Health Informatics*, 28(3):1261–1272, March 2024.

[29] Rudresh Pillai, Avinash Sharma, Neha Sharma, and Rupesh Gupta. Brain tumor classification using vgg 16, resnet50, and inception v3 transfer learning models. In *2023 2nd International Conference for Innovation in Technology (INOCON)*, pages 1–5, 2023.

[30] Leon Bottou. Large-scale machine learning with stochastic gradient descent. In´*Proceedings of COMPSTAT'2010*, pages 177–186. Physica-Verlag HD, 2010.

[31] Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with bayesian optimization. Technical report, 2016.

[32] I Goodfellow, Y Bengio, and A Courville. volume 6. MIT Press, 2016.

[33] Edward Raff. Inside Deep Learning: Math, Algorithms, Models. Manning Publications, Shelter Island, NY, 2022.

[34] Msoud Nickparvar. Brain tumor MRI dataset, 2021.