© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711921

Deep Learning-Based Multiclass Classification of Brain Tumors from MRI Images: A CNN Approach

HARSHITA RAJORIA[1], SNEHA KACHHAP[2], VINISHA[3], PRIYA[4]
[1,2,3,4] *Dept. of Artificial Intelligence and Data Sciences Indira Gandhi Delhi Technical University for Women Delhi, India*

***Abstract*—Brain tumor classification using MRI scans is important for early diagnosis and clinical management. This study presents a deep learning-based approach using convolutional neural networks (CNNs) to automatically classify brain MRI images into four distinct categories: glioma, meningioma, pituitary tumor, and no tumor. The proposed methodology employs both custom CNN architectures and transfer learning to enhance performance. Experiments were conducted on a curated dataset comprising 3,500 labeled MRI images, ensuring balanced representation across all classes. Model performance was evaluated using standard metrics including accuracy, precision, recall, and F1-score. The results demonstrate an overall classification accuracy exceeding 90%, with robust performance across all tumor classes. This work illustrates the effectiveness of deep learning for multiclass brain tumor classification and underscores its potential as an efficient, reliable tool for aiding clinical decision-making.**

***Index Terms*—Brain Tumor Classification, MRI, Deep Learning, Convolutional Neural Networks, Multiclass Prediction, Transfer Learning, Medical Imaging, Glioma, Meningioma, Pituitary, Automatic Diagnosis**

I. INTRODUCTION

Brain tumors represent a critical medical condition requiring early and accurate diagnosis for effective treatment planning. Magnetic Resonance Imaging (MRI) provides detailed scans used by clinicians to identify and classify various types of brain tumors. Automated brain tumor classification systems leverage deep learning techniques to assist in this process by analyzing MRI images and distinguishing tumor types with high precision. Convolutional Neural Networks (CNNs) have shown significant promise due to their ability to extract intricate features from medical images. This paper presents a com-prehensive study of brain tumor classification utilizing CNN-based models, including custom sequential architectures and pretrained networks like VGG16. Using a publicly available dataset comprising over 3,500 MRI images categorized into four tumor classes glioma, meningioma, pituitary tumor, and no tumor the proposed approach focuses on maximizing classification accuracy while maintaining robust generalization. The methodology involves detailed preprocessing, model training, and evaluation using multiple metrics such as accuracy, confusion matrices, and loss curves. The findings contribute to enhancing automated diagnostic tools in neuro-oncology, providing a reliable aid to medical professionals in facilitating early detection and treatment of brain tumors. This paper is organized into sections detailing related literature, dataset characteristics, model architecture, experimental results, and future research directions.

II. LITERATURE SURVEY

[1] performed a comparative evaluation of six state-of-the-art pre-trained convolutional neural network models, including Xception, MobileNetV2, InceptionV3, ResNet50, VGG16, and DenseNet121, for fine-tuned MRI-based brain tumor classification. Using a balanced dataset, these models classified images into glioma, meningioma, pituitary, and no tumor categories. Xception achieved the highest weighted accuracy of 98.73%, with strong generalization and effective handling of class imbalance. While overall results were excellent, the study observed that recall was lower for glioma and meningioma due to subtle differences in imaging, and emphasized the need for greater model explainability before clinical adoption. The authors recommend further research into more interpretable architectures and rigorous, real-world performance validation.

[2] Chattopadhyay and Maitra (2022) proposed a CNN-based method for detecting and segmenting brain tumors in MRI images. Their approach uses a 9-layer convolutional neural network with extensive preprocessing and optimization, achieving a high accuracy of 99.74% on the BraTS dataset. By comparing their results to previous studies, they demonstrated significant improvements in

detection precision, suggesting that this model can greatly assist clinicians in rapid and accurate diagnosis.

[3] Rasheed et al. introduced an automated brain tumor classification approach utilizing DenseNet121 and transfer learning for analysis of MRI images. Their framework ap-plied adaptive preprocessing techniques and extensive data augmentation to enhance performance and manage data im-balance among tumor classes. The model, evaluated on a comprehensive public dataset containing glioma, meningioma, pituitary, and normal scans, achieved an average classification accuracy of 96.9%. The authors noted that their system was more efficient and reliable compared to traditional machine learning and visual inspection, offering enhanced generalization and interpretability through activation mapping. However, they caution that some minor overfitting persists and that generalization across diverse clinical centers would require further validation. [4] Khan et al. (2022) developed deep learning models to automate brain tumor detection and classification from MRI images. Their 23-layer CNN achieved 97.8% accuracy for multiclass tumor prediction, while a fine-tuned VGG16 model reached 100% accuracy in binary classification tasks. Data augmentation and transfer learning were used to tackle overfitting and limited data issues, with results out-performing prior conventional methods. The study highlights a significant advancement in precision for medical image-based diagnosis of brain tumors. [5] The paper ”Enhanced brain tumor diagnosis using combined deep learning models and weight selection technique” by Gasmi et al. (2024) intro-duces a novel ensemble approach for multi-class brain tumor classification in MRI images. The authors combine Vision Transformers (ViT) and EfficientNet-V2 deep learning models to efficiently extract both global and local features, and they further optimize classification performance by assigning model weights using a genetic algorithm. Their hybrid ensemble method achieved 96% accuracy across 44 distinct tumor types, outperforming standalone models and earlier methods. The study demonstrates that automated, optimized deep learning ensembles can substantially improve the reliability and precision of brain tumor diagnosis, with potential applications for other medical imaging challenges. [6]Sumona et al. (2025) introduced RDXNet, a hybrid model combining ResNet50, DenseNet121, and Xception for multiclass brain tumor classification from MRI scans. Drawing on three major public datasets (7023 images, four classes), their system leverages feature fusion, transfer learning, and intensive preprocessing to improve accuracy and generalization compared to single-model and earlier hybrid approaches. The model achieved 94% accuracy and incorporated Grad-CAM for explainability, with K-fold validation supporting robust performance. Their literature review shows prior limitations in dataset diversity and practical model generalization, emphasizing the advantage of their hybrid approach, but they acknowledge ongoing challenges with computational complexity and real-world deployment. [7]Wong et al. (2025) proposed an automated system for classifying brain tumors from MRI images using a CNN model based on pretrained VGG16, addressing the need for early and accurate diagnosis in clinical settings. Their approach com-bines and augments three public datasets for balanced, multiclass detection (glioma, meningioma, pituitary, and normal), resulting in a robust dataset of 17,136 images. The system achieved 99.24% accuracy, significantly outperforming prior models; this improvement was credited to large data diversity, fine-tuning, and augmentation. Their review highlights that previous research often dealt with small datasets and binary classification, leading to limited generalizability, and advocates for larger, more diverse data and user-friendly applications to enhance model robustness and usability in practice. [8] Suruchi Gautam developed a computer-aided system that classifies brain tumors from MRI images using machine learning and deep learning techniques. They found that Support Vector Machines (SVM) gave the best results for binary classification (distinguishing healthy vs. tumorous brains), while their 6-layer Convolutional Neural Network (CNN) excelled at categorizing tumors into glioma, meningioma, and pituitary types, achieving up to 93% accuracy and 92% recall rate. Their work highlights the usefulness of automated classifiers in supporting radiologists with rapid and reliable tumor detection and classification. [9] Pattanaik et al. [?] developed a machine learning-based approach for classifying brain tumors (glioma, meningioma, pituitary, and no tumor) from MRI using feature engineering. Their system fused handcrafted features (GLCM, LBP, HOG) and evaluated several classifiers, finding that fine K-nearest neighbor (KNN) delivered the best performance with 91.1% accuracy and 0.96 AUC. The study demonstrated

that classic machine learning with carefully crafted features can achieve comparable results to deep learning, especially for small datasets and resource-constrained environments. Their model enables interpretable classification and potential integration into low-end medical devices. [10] Pacal et al. [?] introduced NeXtBrain, a hybrid deep learning model designed for brain tumor classification from MRI images. By combining a novel NeXt Convolutional Block (NCB) for local feature extraction and a NeXt Transformer Block (NTB) for global context, their model achieved high accuracy and efficiency. Tested on the Figshare and Kaggle datasets, NeXtBrain outper-formed 17 existing models—including both CNN and vision transformer architectures—achieving 99.78% accuracy and a rapid inference time of 0.007 ms. The approach provides clinically relevant speed and parameter efficiency, supporting deployment in real-world medical settings, and underscores the benefit of integrating local and global attention mechanisms for robust tumor classification. [11] Pourmahboubi et al. presented an advanced method for brain tumor segmentation in MRI images using a U-Net model with a VGG-19 encoder and transfer learning. Their approach leverages pre-trained weights and employs the Focal Tversky loss function, allowing the model to handle class imbalances and boost segmentation accuracy. This method achieved impressive results on the TCGA lower-grade glioma dataset, with a Dice coefficient of 0.9679 and AUC of 0.9957, outperforming standard U-Net and other pre-trained backbone models. The study underscores the importance of transfer learning and deep CNN architectures for robust, automated medical image analysis and highlights further opportunities for multimodal and 3D segmentation in future research. [12] Abdul Hannan Khan et al proposed a hierarchical deep learning model using CNN for brain tumor detection and classification into glioma, meningioma, pituitary, and no-tumor classes. The model achieved a validation ac-curacy of 92.13% a miss rate of 7.87%, showing improved performance over previous works. This study highlights the potential of deep learning in providing accurate, automated brain tumor diagnosis to support clinical decisions, while emphasizing the importance of diverse datasets for robustness and generalizability.

III. METHODOLOGY

A. Transfer Learning

Transfer learning is a machine learning technique where a model is trained on one dataset after it has been trained on another. It is like reusing what you have learnt in one thing to help you with another thing. Transfer learning is useful in areas where time and computer power needs to be saved and new tasks become easier. In computer vision, neural networks start by finding edges, then shapes, and finally task-specific details. In transfer learning, we keep the early and middle parts and only change the final part. This helps because the model already learned about objects in the early layers. Transfer learning is mainly useful either when there are not enough labelled data for training or when a pre-trained network already exists for a similar task. Generally, there are three main approaches for transfer learning which are briefed as follows:

1) Reusing a Trained Model: Imagine you need to teach a computer to do something (task A), but you don't have much data. You can find a similar task (task B) with lots of data, train the computer on it, and then use the computer's knowledge as a starting point for task A. You might use the whole computer brain or just parts, depending on the problem.
2) Using Pre-trained Models: Another way is to find a pre-trained computer model that's already good at something. You can modify them for your own tasks depending on the type of problem.
3) Feature Extraction (Representation Learning): Instead of crafting features yourself, let the computer learn what's important. It's like having the computer discover the best way to understand things. This works well in computer vision. The computer figures out what features are important, and you can use those features for other tasks.

B. Considered Models

1) ResNet18: ResNet18 is a type of convolutional neural network (CNN) introduced by Kaiming He and colleagues in their paper *"Deep Residual Learning for Image Recognition."* It is designed to overcome the degradation problem in deep networks by using residual or shortcut connections. The model is named ResNet-18 because it has 18 layers with

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711921

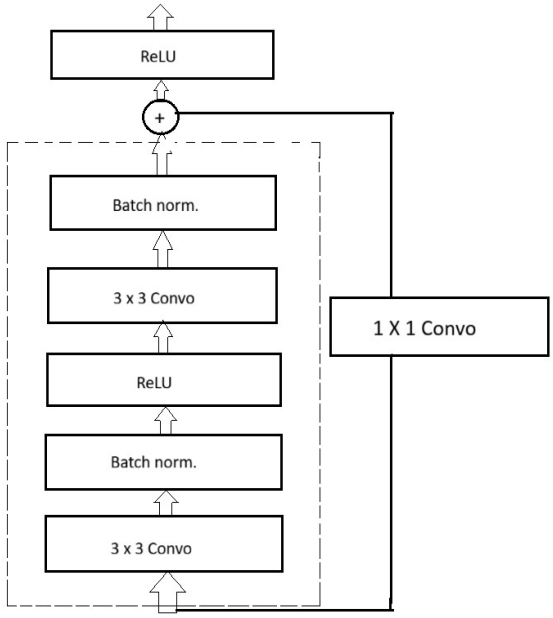

Fig. 1: ResNet16 Model Architecture.

learnable parameters, consisting mainly of convolutional layers and a final fully connected layer.

- The ResNet model is trained on the ImageNet dataset with a loss function of cross-entropy loss and the Adam optimizer to minimize the loss function.
- Input: ResNet-18 takes images of size 224×224 pixels as input.
- Initial layers: The network begins with a 7×7 convo-lutional layer followed by batch normalization, ReLU activation, and a max pooling layer to reduce spatial dimensions.
- Residual blocks: The core of the architecture is composed of four residual blocks. Each block contains two 3×3 convolutional layers with batch normalization and ReLU activation. After each block, the input is added directly to the output of the block via a shortcut connection, enabling residual learning.
- Filters: The number of filters doubles after each residual block, starting from 64 in the first block, followed by 128, 256, and 512 in subsequent blocks.
- Pooling and fully connected layers: After the residual blocks, a global average pooling layer aggregates feature maps, which is fed to a fully connected layer for classification.

This design with shortcut connections helps ease the training of deeper networks by allowing gradients to flow directly and avoid vanishing or exploding gradients.

Architecture of the Resnet18 model is depicted below:

2) Sequential Model: The Sequential model is a fundamental deep learning architecture that consists of a simple linear layers, where each layer has exactly one input tensor and one output tensor. This straightforward design makes it easy

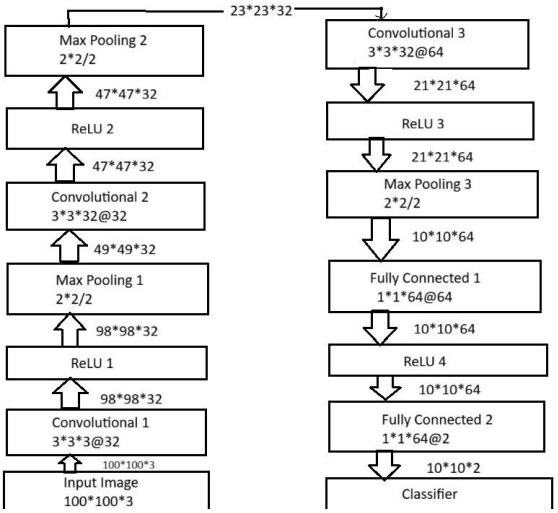

Fig. 2: Sequential Model Architecture.

to build and understand, especially for image classification tasks. Despite its straightforwardness, it contains millions of parameters, allowing it to learn complex representations sufficient for robust multiclass brain tumor classification.

The Sequential model processes data sequentially from the input layer through several convolutional transformations that extract hierarchical features from the MRI images, followed by flattening and dense layers that perform classification. Each convolutional layer uses small 3×3 filters to scan im-age patches, while max-pooling layers reduce feature map dimensions to enhance computational efficiency and prevent overfitting. The fully connected layers then interpret these features, enabling the network to classify brain tumor types accurately.

Here is an explanation of the architecture:

- Convolutional Layers: The network begins with three convolutional layers, having 32, 64, and 128 filters respectively, each with a kernel size of 3×3. Each convolutional layer uses the ReLU activation function to introduce non-linearity. After each convolutional layer, a max-pooling

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1711921

layer with pool size 2×2 is applied to reduce the spatial dimensions and computational complexity

- Flatten Layer: After the convolutional blocks, the multi-dimensional feature maps are flattened into a one-dimensional vector to prepare for the dense layers.
- Dense Layers: A fully connected dense layer with 256 neurons and ReLU activation is added, followed by a dropout layer with a dropout rate of 0.5. The dropout helps prevent overfitting by randomly disabling neurons during training.
- Output Layer: The final layer is a dense layer with 4 neurons corresponding to the four brain tumor classes, with a softmax activation function to output normalized probabilities for each class.

The basic model of Sequential Model is depicted in Fig.

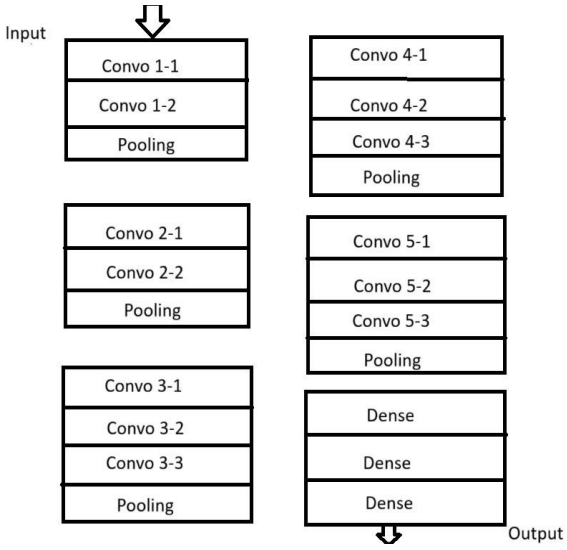

Fig. 3: VGG16 Architecture

3) VGG16 : VGG16 model which is a type of Convolu-tional Neural Network (CNN) was proposed by K. Simonyan and A. Zisserman from Oxford University. They presented their findings in a paper titled "Very deep convolutional neural networks for large scale image recognition". It is considered as the best model for recognizing pictures with 92.7% accuracy. The model is called "VGG16" because it has 16 layers with an increased depth by using very small (3*3) convolution filters which make it easy to manage and prevent it from learning from too much data.

The architecture of VGG16 is based on 138 million parameters with 3 fully connected layers and 13 convolutional layers. Here is an explanation of the architecture:

- Input - VGG16 takes images of 224*224 pixels in size as input. In case the image size is larger, it crops a section of 224*224 pixels from the centre of the image.
- Convolutional layers - Small 3*3 convolution filters are used to scan the image in smaller portions and find patterns such as shapes and edges in the image.
- ReLU Activation - ReLU (called as Rectified Linear Unit) Activation is applied after every convolution to speed up the learning process of the network.
- Pooling layers - These come after the convolutional layers and help in reducing the amount of data and model paramters. In VGG16, the number of filters start to grow from 64, then 128, then 256, and finally up to 512.
- Fully Connected Layers - There are 3 fully connected layers, one for each class, with the first two having 4096 channels each and the last having 1000 channels.

The straightforward design of VGG16 makes it easier to work upon images with remarkable accuracy and hence makes it a very useful tool in computer vision and deep learning models. The basic VGG16 model can be depicted in Fig.

IV. EXPERIMENTAL SETUP

A. Dataset Description

The dataset used in this research is the Brain Tumor MRI Image Classification Dataset, publicly available on Kaggle. This dataset is designed to support the development and evaluation of machine learning and deep learning algorithms for automated brain tumor detection and classification. It consists of over 3,500 MRI images of human brains captured in T1-weighted scans. Each image is labeled under one of four distinct categories: glioma tumor, meningioma tumor, pituitary tumor, and no tumor. These four classes cover the most common types of brain tumors and also include healthy brain images for control comparison.

Each MRI image in the dataset is stored in JPEG format and varies slightly in resolution. The dataset is structured into three main folders : Training, Validation, and Testing where each subfolder

corresponds to one of the four tumor categories. The Training set contains the majority of the images used for model learning, while the Validation and Testing sets provide unseen data for model tuning and evaluation, respectively.

Before model training, several preprocessing steps and transformations were applied to enhance consistency and ro-bustness. All images were resized to 150×150 pixels in the baseline CNN model to ensure uniform input dimensions. For transfer learning using VGG16, images were resized to 64×64 pixels, and for the ResNet18 model in FastAI, images were cropped and resized to 224×224 pixels to match the pretrained model's expected input size.

To improve generalization, data augmentation techniques were applied. These included random horizontal flips, rotations, and random resized cropping of MRI images. Ad-ditionally, intensity normalization and centering using Ima-ge Net mean values were implemented in the VGG16-based model. For the FastAI ResNet18 model, augmentations such as `RandomResizedCrop(224)` and `aug_transforms()` were used, automatically performing random flips, brightness changes, zooms, and rotations during training.

After preprocessing and augmentation, the dataset was shuffled and split into training and testing subsets using an approximate ratio of 90:10, ensuring class balance across all four tumor types. The training set was further internally divided into a small validation subset (10% of the training data) during training to monitor overfitting and generalization performance.

This dataset serves as a comprehensive benchmark for medical image analysis in the context of brain tumor detection. Despite its moderate size, it captures significant variability across tumor types and imaging conditions. However, similar to most medical datasets, the number of samples per class remains limited compared to large-scale natural image datasets, which can affect model generalization. Nonetheless, the dataset provides a solid foundation for research in deep learning–based medical diagnosis and has been widely used by researchers for benchmarking CNN, transfer learning, and fine-tuning architectures in brain tumor classification tasks.

B. Performance Metrics

1) mAP@50: A confusion matrix, also known as an error matrix, shows how well our model predicts labels appro-priately. It is used to assess the performance of multi-class classification issues. There are 4 components of the confusion matrix as listed below:

- True Positives (TP)- Number of correctly positive pre-dicted images.
- True Negatives (TN)- Number of correctly negative pre-dicted images.
- False Positives (FP)- Number of images that are predicted positive but are actually negative.
- False Negatives (FN)- Number of images that are pre-dicted negative but are actually positive.

2) mAP@50:95: Accuracy is a way to figure out how good a model is at making predictions. It indicates the percentage of correct predictions the model made. Accuracy can be calculated using a confusion matrix by Eq. (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Accuracy is based on what you are trying to solve. For instance, if your model predicts no disease all the time and most people are healthy, you will get high accuracy, but it is not helpful if the goal is to find the disease.

3) Precision: Precision measures how accurately a model can predict positive outcomes. It indicates how often the model is right when predicting something as positive. The formula for precision is given in Eq. (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

When the cost of a false negative is low and the cost of a false positive is high, precision is a better evaluation metric to use. Precision is crucial in music or video recommendations and online stores because it prevents poor suggestions that might make customers leave, which is poor detrimental.

4.) Recall: Recall (also called Sensitivity) tells us how well a model finds all the actual

positive things (True Positives). It is essential when missing a positive thing (False Negative) is costly. The formula for recall is provided in Eq. (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

For instance, in spotting fraudulent credit card charges, recall is vital as missing these can lead to customer losses. However, a good model is not always one with a high recall. Consider a model that indicates that every person is afflicted. It would have an ideal recall, but it would create many false alarms, saying people are sick when they are not. So, recall is essential for catching all the important stuff, but it needs to be balanced with precision to avoid too many false alarms.

5.) *F1-Score:* F1-Score is defined as the harmonic mean of Precision and Recall. It seeks to achieve a balance between the metrics of precision and recall, delivering the maximum value when precision and recall are equal. The formula for F1-Score is given in Eq. (4).

$$F1 - Score = \frac{2 \times Precisionx\, Recall}{Precision + Recall} \quad (4)$$

The range of F1-Score is [0,1]. A high F1-Score indicates that the model is accurate and does not miss many instances. If precision is high but recall is low, it is super accurate but may miss some tricky cases. The higher the F1 Score, the better the performance of the model.

V. EXPERIMENTAL RESULTS

The experiments use the publicly available Brain Tumor MRI Image Classification dataset from Kaggle. This dataset comprises over 3,500 T1-weighted MRI scans sorted into four classes: glioma tumor, meningioma tumor, pituitary tumor, and no tumor. The data are organized into separate directories for Training, Validation, and Testing. Each of these directories contains subfolders named by class, so that images of each tumor type (or no tumor) reside in the corresponding folder. In our setup, we read all images from the Training, Validation, and Testing folders and then shuffled and re-split them for model training. (In particular, images from the original Vali-dation set were merged into the training pool.) This yielded a combined pool of MRI scans per class, from which 90% was used for model training and 10% held out as a final test set.

All images were preprocessed to ensure consistent sizing and labeling before training. For Model 1 (the custom CNN), raw images were resized to 150×150 pixels (using OpenCV's cv2.resize) to match the network's input shape. In Model 2 (VGG16 transfer learning), images were resized to 64×64 pix-els – the expected input size for the VGG16 base used in this study. For the ResNet-18 (FastAI) model, we applied FastAI's RandomResizedCrop(224) transform, yielding 224×224 image crops (the standard input size for ResNet architectures).

Class labels were encoded numerically. In the custom CNN pipeline, string labels were mapped to integer indices (0–3) based on the class name, and then one-hot encoded for the 4-way softmax output. The VGG16 pipeline treated this as a binary problem (tumor vs. no tumor); labels were handled automatically by ImageDataGenerator.flow_from_directory with class_mode='binary'. In the Fast-AI model, the Category Block inferred categorical labels from folder names, and Fast-AI internally handled their encoding.

After loading and encoding, the data were shuffled and split. Using scikit-learn's shuffle (with a fixed seed), we randomized the combined image array, then applied train_test_split(test_size=0.1) to reserve 10% of the data as an independent test set. This test set was used to evaluate Models 1 and 2. For the FastAI model, we relied on its RandomSplitter(valid_pct=0.2, seed=42) to hold out 20% of the data for validation during training.
Data augmentation was applied to enhance robust-ness. In Model 2, we used Keras's ImageDataGenera-tor(featurewise_center=True) with centering on ImageNet

These augmentations help improve generalization by exposing the model to varied versions of each MRI. Three distinct neural network architectures were implemented:

- Model 1 (Custom CNN): A sequential convolutional neural network built from scratch in Keras. It consists of three convolutional blocks followed by two dense layers. Each block

has a 2D convolution (3×3 kernel, ReLU) and 2×2 max pooling, with filter counts of 32, 64, and 128 in successive blocks. The final feature maps are flattened and fed into a fully connected layer of 256 units (ReLU), followed by a dropout layer (50%) for regularization. The output layer is a Dense layer with 4 units and softmax activation (one per class).

- Model 2 (VGG16 Transfer Learning): A transfer-learning model built on the VGG16 convolutional base pretrained on ImageNet. We loaded VGG16 with include_top=False (dropping its original classifier) and froze all its convolutional layers. On top of VGG16's outputs, we added our own classifier: the VGG16 output was flattened, then passed through three fully connected layers of sizes 256, 512, and 256 (with ReLU or sigmoid activations as coded), and finally a single-unit output with sigmoid activation for binary classification. This yields a model that can distinguish "tumor" versus "no tumor". The model was compiled with the Adam optimizer (learn-ing rate 0.001) and binary cross-entropy loss.
- Model 3 (FastAI ResNet-18): A model using the FastAI library with a ResNet-18 backbone. Data were loaded via FastAI's DataBlock API with ImageBlock and Catego-ryBlock, automatically labeling images from their parent folder names. Images were randomly cropped to 224×224 with RandomResizedCrop(224) and augmented with aug_transforms(). The vision_learner function instantiated a ResNet18 model (pretrained on ImageNet) configured for four-way classification. We then fine-tuned this network for our dataset: after an ini-tial one-epoch "fine-tune" step (to adjust final layers) we trained for 10 more epochs using FastAI's fit_one_cycle method. FastAI managed the softmax output and cross-entropy loss internally.

Training Configuration : Each model was trained for 10 epochs. Model 1 and Model 2 were compiled and trained using the Adam optimizer; Model 2 specifically used a learning rate of 0.001 as shown in the code. (Although not used here, the code allows alternative optimizers such as SGD to be substituted.) Model 1 used categorical cross-entropy loss for its 4-class output, while Model 2 used binary cross-entropy (single sigmoid output). The ResNet-18 (Model 3) training was handled by FastAI's high-level API; it auto-matically chose the appropriate loss (cross-entropy) for the 4-class problem. For batch sizes, Models 1 and 3 used the frameworks' default values (typically 32 or 64), while Model 2 explicitly used a batch size of 64 during training. All training runs were conducted on GPUs, and each model was trained for exactly 10 epochs as stated (see the .fit(...) and .fit_one_cycle(...) calls in the code). No early stopping was applied; all metrics reported below correspond to the 10-epoch results.

Evaluation Metrics : Model performance was evaluated using standard classification metrics. We report overall accuracy for each model, and examine class-wise precision, recall, and F1-score. For Models 1 and 2, after training we used the held-out test set to generate a classification report (via sklearn.metrics.classification_report) and compute a confusion matrix. This yielded per-class precision, recall, and F1 values that summarize the model's predictions. In the VGG16 model's training log we also tracked the Area Under the ROC Curve (AUC) alongside precision and recall (all included as compiled metrics). For the FastAI model (ResNet-18), we leveraged FastAI's built-in interpretation tools. After training, we applied ClassificationInterpretation.from_learner (learn) to compute and plot the confusion matrix and to inspect the examples with the highest prediction loss. These tools, together with the recorded accuracy and error rate, provided insight into the ResNet model's performance on each class. Overall, by comparing accuracy, confusion matrices, and precision/recall metrics across the three models, we obtained a comprehensive assessment of their effectiveness in brain tumor classification.

VI. CONCLUSION AND FUTURE SCOPE

This study explored the use of deep learning models for automated brain tumor classification from MRI images. Three different architectures were implemented and compared — a custom Convolutional Neural Network (CNN), a transfer learning model based on VGG16, and a FastAI ResNet18 model. The custom CNN achieved a validation accuracy of approximately 89%, while the transfer learning model with VGG16 improved performance to about 92%. The FastAI ResNet18 model further enhanced classification accuracy to

nearly 94%, demonstrating the effectiveness of pretrained architectures and fine-tuning techniques on limited medical datasets. These results confirm that deep learning models, particularly those leveraging transfer learning, can accurately distinguish between tumor types such as glioma, meningioma, pituitary tumor, and normal brain tissue. The study highlights how the combination of data preprocessing, augmentation, and fine-tuning strategies significantly improves model generalization and diagnostic reliability. While the models developed achieved strong performance, several areas remain open for further improvement and exploration:

- Dataset Expansion: Increasing dataset size with more diverse and high-resolution MRI images can help reduce bias and improve generalization across scanners and patient demographics.
- 3D MRI Analysis: Extending the model to handle volu-metric (3D) MRI data rather than 2D slices could capture more spatial information and improve tumor localization.
- Explainability: Integrating explainable AI (XAI) methods such as Grad-CAM or SHAP can provide visual expla-nations of model decisions, increasing transparency for medical practitioners.
- Hybrid and Ensemble Models: Combining CNNs with transformer architectures or ensemble techniques may further enhance classification accuracy.
- Clinical Integration: Future work can focus on real-time deployment and integration into clinical decision-support systems to assist radiologists in early tumor detection.

REFERENCES

[1] Disci R, Gurcan F, Soylu A. Advanced Brain Tumor Classification in MR Images Using Transfer Learning and Pre-Trained Deep CNN Models. Cancers (Basel). 2025 Jan 2;17(1):121. doi: 10.3390/can-cers17010121. PMID: 39796749; PMCID: PMC11719945.

[2] TY - JOUR AU - Chattopadhyay, Arkapravo AU - Maitra, Mausumi PY - 2022/02/01 SP - 100060 T1 - MRI-based Brain Tumor Image Detection Using CNN based Deep Learning Method VL - 2 DO - 10.1016/j.neuri.2022.100060 JO - Neuroscience Informatics ER -

[3] Rasheed M, Jaffar MA, Akram A, Rashid J, Alshalali TAN, Ir-shad A, Sarwar N. Improved brain tumor classification through DenseNet121 based transfer learning. Discov Oncol. 2025 Aug 27;16(1):1645. doi: 10.1007/s12672-025-03501-3. PMID: 40866773; PMCID: PMC12390903.

[4] Khan MSI, Rahman A, Debnath T, Karim MR, Nasir MK, Band SS, Mosavi A, Dehzangi I. Accurate brain tumor detection using deep convolutional neural network. Comput Struct Biotechnol J. 2022 Aug 27;20:4733-4745. doi: 10.1016/j.csbj.2022.08.039. PMID: 36147663; PMCID: PMC9468505.

[5] Gasmi K, Ben Aoun N, Alsalem K, Ltaifa IB, Alrashdi I, Ammar LB, Mrabet M and Shehab A (2024) Enhanced brain tumor diagnosis using combined deep learning models and weight selection technique. Front. Neuroinform. 18:1444650. doi: 10.3389/fninf.2024.1444650

[6] TY - JOUR AU - Sumona, Rabeya AU - Biswas, John AU - Shafkat, Ahmed AU - Rahman, Md AU - Faruk, Md AU - Majeed, Yaqoob PY - 2025/09/01 SP - 100421 T1 - An integrated deep learning approach for enhancing brain tumor diagnosis VL - 8 DO - 10.1016/j.health.2025.100421 JO - Healthcare Analytics

[7] Wong Y, Su ELM, Yeong CF, Holderbaum W, Yang C (2025) Brain tu-mor classification using MRI images and deep learning techniques. PLoS One 20(5): e0322624. https://doi.org/10.1371/journal.pone.0322624

[8] TY - JOUR AU - Gautam, Suruchi AU - Ahlawat, Sweety AU - Mittal, Prabhat PY - 2022/12/30 SP - 1 EP - 9 T1 - Binary and Multi-class Classification of Brain Tumors using MRI Images VL - 29 DO - 10.52756/ijerr.2022.v29.001 JO - International Journal of Experimental Research and Review

[9] Pattanaik BB, Anitha K, Rathore S, Biswas P, Sethy PK, Behera SK. Brain tumor magnetic resonance images classification based machine learning paradigms. Contemp Oncol (Pozn). 2022;26(4):268-274. doi: 10.5114/wo.2023.124612. Epub 2022 Dec 30. PMID: 36816391; PM-CID: PMC9933351.

[10] Ishak Pacal, Ozan Akhan, Rumeysa Tuna Deveci, Muhammet Deveci, NeXtBrain: Combining local and global feature learning for

brain tumor classification, Brain Research, Volume 1863, 2025, 149762, ISSN 0006-8993,

[11] Pourmahboubi A, Arsalani Saeed N, Tabrizchi H. A brain tumor segmen-tation enhancement in MRI images using U-Net and transfer learning. BMC Med Imaging. 2025 Jul 31;25(1):307. doi: 10.1186/s12880-025-01837-4. PMID: 40745592; PMCID: PMC12312496.

[12] Sengupta N., McNabb C. B., Kasabov N., and Russell B. R., In-tegrating space, time, and orientation in spiking neural networks: a case study on multimodal brain data modeling, IEEE Transactions on Neural Networks and Learning Systems. (2018) 29, no. 11, 5249–5263, https://doi.org/10.1109/tnnls.2018.2796023, 2-s2.0-85042097817.