



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

BRAIN TUMOR DETECTION USING DEEP LEARNING

S. Savitha¹, Dr. J. Evangelin Deva Sheela MCA., M.Phil., Ph.D²

Department of Computer Applications, Sarah Tucker College (Autonomous), Tirunelveli-7.

ABSTRACT :

Brain tumors are among the most critical neurological conditions, and their early and accurate detection from Magnetic Resonance Imaging (MRI) significantly influences treatment planning. Manual interpretation of MRI scans is time-consuming and can be subjective, motivating the use of automated deep learning based classification systems. This paper presents a comparative study of four deep learning architectures — a Convolutional Neural Network (CNN) baseline, DenseNet121, EfficientNetB0, and ResNet50 — for classifying brain MRI images into four categories: glioma, meningioma, no tumor, and pituitary tumor. The three transfer-learning models were trained under an identical two-stage protocol (frozen-backbone training followed by fine-tuning of the last 30 layers) using the same image size, batch size, random seed, and augmentation strategy, ensuring a fair comparison. On a held-out test set of 1,600 images, ResNet50 achieved the highest test accuracy of 93.06%, followed by EfficientNetB0 (87.75%), DenseNet121 (85.75%), and the CNN baseline (67.94%). The best-performing model, ResNet50, was deployed in an interactive Streamlit web application that returns a predicted class with a confidence score. The work also discusses class-wise performance patterns, including consistently lower recall for the glioma class across all transfer-learning models, and outlines the limitations and future scope of the system.

Keywords: brain tumor classification, MRI, deep learning, transfer learning, CNN, DenseNet121, EfficientNetB0, ResNet50, Streamlit

1. INTRODUCTION

A brain tumor is an abnormal growth of cells within the brain that can be benign or malignant, and its type, size, and location strongly influence the choice of treatment. Magnetic Resonance Imaging (MRI) is the primary non-invasive imaging modality used to detect and characterize brain tumors, but manual review of MRI slices by radiologists is time-consuming and can vary between observers, particularly when tumor boundaries are unclear. Deep learning, and convolutional neural networks in particular, has shown strong potential for automating medical image classification tasks by learning discriminative visual features directly from data.

Training a deep network from scratch typically requires a very large labelled dataset, which is difficult to obtain for medical imaging. Transfer learning addresses this limitation by reusing networks pretrained on large natural-image datasets such as ImageNet and adapting them to a new task through fine-tuning, which has been shown to substantially improve accuracy on limited medical imaging datasets.

This work trains and evaluates four architectures — a CNN baseline and three ImageNet-pretrained backbones (DenseNet121, EfficientNetB0, ResNet50) — on a public brain MRI dataset covering four classes: glioma, meningioma, no tumor, and pituitary tumor. Unlike many published comparisons that use different image sizes, splits, or training schedules for each architecture, all three transfer-learning models in this study are trained under an identical protocol so that the reported differences reflect the architectures themselves rather than inconsistent experimental settings. The best-performing model is deployed in a Streamlit web application for interactive demonstration.

The remainder of this paper is organized as follows. Section 2 reviews related work on deep learning based brain tumor classification. Section 3 states the problem. Section 4 lists the objectives of this work. Section 5 describes the proposed system and its modules. Section 6 details the dataset, model architectures, and training configuration. Section 7 lists the implementation stack. Section 8 presents the experimental results. Section 9 discusses the findings and limitations, Section 10 concludes the paper, and Section 11 outlines directions for future work.

2. RELATED WORK

A number of recent studies have applied transfer learning to brain tumor MRI classification. Table 1 summarizes selected studies relevant to the proposed comparison.

Study	Approach	Main contribution / limitation
Swati et al. (2019) [1]	DenseNet-based transfer learning with block-wise fine-tuning	Proposed a block-wise fine-tuning strategy for pretrained CNNs on a closed brain tumor MRI dataset; showed transfer learning is effective even with limited medical data.

Study	Approach	Main contribution / limitation
Asadujjaman (2025) [2]	Modified ResNet50 vs VGG16/19, EfficientNetB0, Inception, DenseNet121	Reported a fine-tuned ResNet50 as the best performer (99.31%) on a combined SARTAJ/Br35H/Figshare dataset, highlighting gains from added custom layers.
Vadde and Bukaita (2025) [3]	CNN baseline vs ResNet18	Found ResNet18 clearly outperforming a plain CNN baseline, reinforcing the value of residual connections and transfer learning for MRI classification.
Golkarieh et al. (2025) [4]	VGG19, ResNet50, InceptionV3, EfficientNetV2 on a 17-class dataset	Found EfficientNetV2 the strongest of the evaluated models on a larger multiclass dataset (4,449 images), showing results can shift with more classes and more data.

Table: 2.1 Summary of related transfer-learning studies on brain tumor MRI classification.

These studies consistently show that transfer learning from ImageNet substantially improves brain tumor classification accuracy over training from scratch, though the best-performing architecture varies with dataset size, class count, and fine-tuning strategy [1-4]. Reported accuracies across the literature also range widely, from around 91% to over 99%, which reflects differences in dataset difficulty, number of classes, image resolution, and how rigorously the train/test split was constructed, rather than a single definitive ranking of architectures. Few of these works evaluate multiple architectures under a strictly identical training protocol and image resolution within the same study, which motivates the controlled, fair comparison undertaken in this work.

3. PROBLEM STATEMENT

Manual MRI-based diagnosis of brain tumors is time-intensive and dependent on radiologist expertise, and reported accuracies for automated approaches vary widely across published work due to differing datasets, image resolutions, and training protocols, which makes architecture comparisons difficult to interpret directly. In particular, when two architectures are trained at different image resolutions, with different augmentation strategies, or for a different number of epochs, an observed accuracy gap could be caused by any of these factors rather than the architecture itself. There is therefore a need to build and rigorously compare multiple deep learning architectures for brain MRI classification under identical experimental conditions, and to deploy the most accurate model as an accessible demonstration tool.

4. OBJECTIVES

- To preprocess and organize a public brain MRI dataset into four balanced classes for training, validation, and testing.
- To train a CNN baseline from scratch and three ImageNet-pretrained backbones (DenseNet121, EfficientNetB0, ResNet50) using an identical two-stage transfer-learning protocol.
- To evaluate every model using accuracy, precision, recall, F1-score, and confusion matrices on the same held-out test set.
- To identify the best-performing model and deploy it in an interactive web application that reports a confidence score alongside the prediction.

5. PROPOSED SYSTEM

The proposed system takes a brain MRI image as input and processes it through a deep learning pipeline to classify it into one of four tumor categories. The input image is first preprocessed to make it suitable for the model, and is then passed through a transfer learning model that has been trained to recognize patterns associated with each tumor type. During training, the model learns from a large set of labeled MRI images, and the best-performing model is selected based on its accuracy. This selected model is then used to produce the final output, which is the predicted tumor class for the given MRI image. Figure 1 shows the overall architecture of the proposed system.

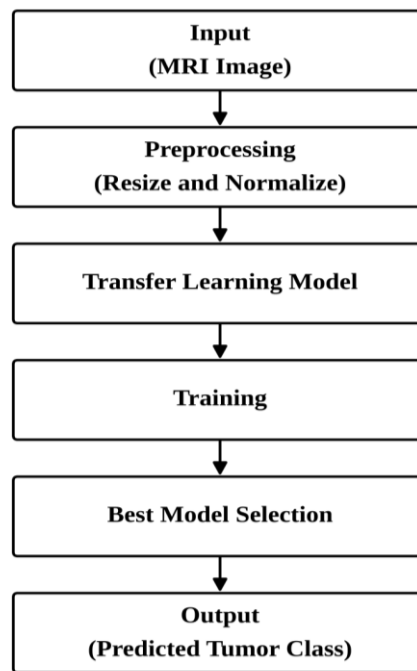


Figure 5.1 Architecture of the proposed brain tumor detection.

5.1 SYSTEM MODULES

Module	Function
Data Loading and Preprocessing	Loads MRI images from train/validation/test folders, resizes them to a fixed size, and batches and shuffles the training set with a fixed random seed for reproducibility.
Data Augmentation	Applies random horizontal flip, rotation, and zoom during training to improve generalization and reduce overfitting.
CNN Baseline	A convolutional network trained from scratch to establish a baseline accuracy for comparison.
Transfer Learning Backbones	DenseNet121, EfficientNetB0, and ResNet50, pretrained on ImageNet, used as feature extractors.
Two-Stage Fine-Tuning	Stage 1 trains only a new classifier head on the frozen backbone; Stage 2 unfreezes the last 30 backbone layers (BatchNormalization layers kept frozen) and continues training at a lower learning rate.
Evaluation	Computes accuracy, precision, recall, F1-score, and a confusion matrix on the held-out test set for every model.
Deployment	A Streamlit web application that loads the best model and returns a predicted class with a confidence score for an uploaded MRI image.

Table: 5.1 Functional modules of the proposed system.

6. METHODOLOGY

6.1 Description

The publicly available Brain Tumor MRI Dataset [8] was used, which contains MRI images across four classes: glioma, meningioma, no tumor, and pituitary tumor. The images were organized into train, validation, and test folders, with the test set fixed at 1,600 images (400 per class) so that every model is evaluated on exactly the same unseen data. For the transfer-learning models, images were resized to 256x256 pixels; the CNN baseline was trained and evaluated at 224x224 pixels. A batch size of 16 and a fixed random seed of 42 were used throughout for reproducibility.

6.2 Model Architectures

The CNN baseline is a convolutional network trained from scratch without pretrained weights, serving as a reference point for the benefit of transfer learning. The three transfer-learning models (DenseNet121 [6], EfficientNetB0 [7], and ResNet50 [5]) use their respective ImageNet-pretrained convolutional base with the original classification head removed, followed by a GlobalAveragePooling2D layer, a Dropout layer, and a final Dense layer with softmax activation over the four classes. Random horizontal flip, rotation, and zoom augmentation layers are applied before the backbone during training.

6.3 Training Configuration

Each transfer-learning model was trained in two stages. In Stage 1, the pretrained backbone was frozen and only the new classifier head was trained for up to 10 epochs using the Adam optimizer with a learning rate of 0.0005. In Stage 2, the backbone was unfrozen except for its final 30 layers being fine-tuned (BatchNormalization layers were kept frozen throughout to preserve stable running statistics), and training continued for up to 15 epochs with a reduced learning rate of 0.00001. Sparse categorical cross-entropy was used as the loss function. Three callbacks were used during training: ModelCheckpoint (saving the model with the lowest validation loss), EarlyStopping (patience of 5 epochs on validation loss, restoring the best weights), and ReduceLROnPlateau (halving the learning rate after 2 epochs without improvement, down to a minimum of 1e-6).

Hyperparameter	Stage 1 (frozen backbone)	Stage 2 (fine-tuning)
Trainable layers	Classifier head only	Head + last 30 backbone layers (BatchNorm frozen)
Optimizer	Adam	Adam
Learning rate	0.0005	0.00001
Max epochs	10	15
Dropout rate	0.45	0.45
Image size	256x256 (224x224 for CNN baseline)	256x256 (224x224 for CNN baseline)
Batch size	16	16
Random seed	42	42

Table: 6.1 Hyperparameter summary for the two-stage transfer-learning protocol.

6.4 Evaluation Metrics

Every trained model was evaluated on the same held-out test set of 1,600 images using overall accuracy, per-class precision, recall and F1-score, macro-averaged precision/recall/F1-score, and a confusion matrix, computed using scikit-learn. For a given class, TP, TN, FP and FN denote the number of true positives, true negatives, false positives and false negatives respectively; the metrics were calculated as follows:

$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

For each class, precision, recall and F1-score were calculated separately, and then averaged equally across all four classes using macro-averaging. This method treats every class with the same importance, irrespective of the number of test images present in that class.

7. IMPLEMENTATION

Component	Technology used
Programming language	Python
Deep learning framework	TensorFlow / Keras
Evaluation metrics	scikit-learn (classification_report, confusion_matrix)

Component	Technology used
Visualization	Matplotlib, Seaborn, Plotly
Web application	Streamlit
Hardware	Standard laptop / desktop (CPU); no specialized hardware required

Table: 7.1 Software components used in the implementation.

8. EXPERIMENTAL EVALUATION AND RESULTS

All four models were evaluated on the identical 1,600-image test set. Table 4 reports the overall test accuracy and macro-averaged precision, recall, and F1-score for each model.

Model	Test Accuracy	Macro Precision	Macro Recall	Macro F1-score
CNN Baseline	67.94%	70.04%	67.94%	61.73%
DenseNet121	85.75%	86.36%	85.75%	85.32%
EfficientNetB0	87.75%	88.30%	87.75%	87.33%
ResNet50	93.06%	93.61%	93.06%	92.86%

Table: 8.1 Overall test-set performance of the four evaluated models.

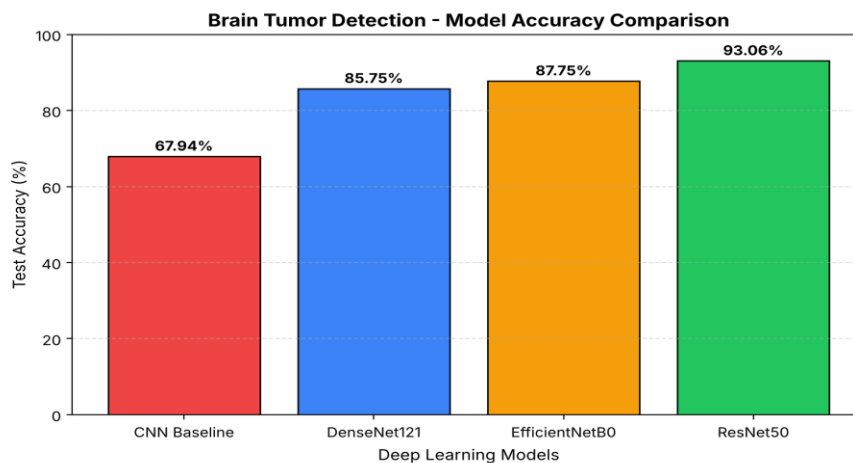


Figure 8.1 Performance comparison on accuracy across four evaluated models.

8.1 ResNet50 — Class-wise Results

ResNet50 achieved the highest overall accuracy and was selected as the deployed model. Table 5 reports its per-class precision, recall, and F1-score on the test set.

Class	Precision	Recall	F1-score	Support
Glioma	99.35%	76.75%	86.60%	400
Meningioma	85.94%	96.25%	90.80%	400
No Tumor	93.02%	100.00%	96.39%	400
Pituitary	96.13%	99.25%	97.66%	400

Table 8.2 Class-wise performance of the ResNet50 model on the test set.

8.2 Class-wise Pattern Across Models

A consistent pattern was observed across all three transfer-learning models: glioma recall was noticeably lower than for the other three classes (DenseNet121: 67.75%, EfficientNetB0: 68.75%, ResNet50: 76.75%), while precision for the glioma class remained high in every model. This indicates that when a model predicts glioma, it is usually correct, but a substantial fraction of true glioma images are instead classified as another class, most likely meningioma, whose MRI appearance can overlap with glioma in some slices.

8.3 The Streamlit Application

The verified ResNet50 model was deployed in a multi-page Streamlit web application to make the system usable for interactive demonstration. The application provides four sections. The Home page shows a summary of dataset statistics and an automatically ranked leaderboard of all evaluated models, generated directly from the saved evaluation results rather than hardcoded values. The Predict page allows a user to upload an MRI image and returns the predicted class together with a confidence score; if the confidence falls below a fixed threshold of 70%, the application displays an explicit low-confidence warning so that uncertain predictions are not presented with false assurance. The Model Comparison page visualizes accuracy and macro precision/recall/F1-score across all four models, along with a per-class breakdown and confusion matrix for a user-selected model. The About page documents the dataset, classes, and modeling approach. Throughout the application, a visible disclaimer states that the tool is for research and educational purposes only and is not a medical device. Figures 3 and 4 show the Predict page and the Model Comparison page of the deployed application.

9. SAMPLE SCREENSHOTS:

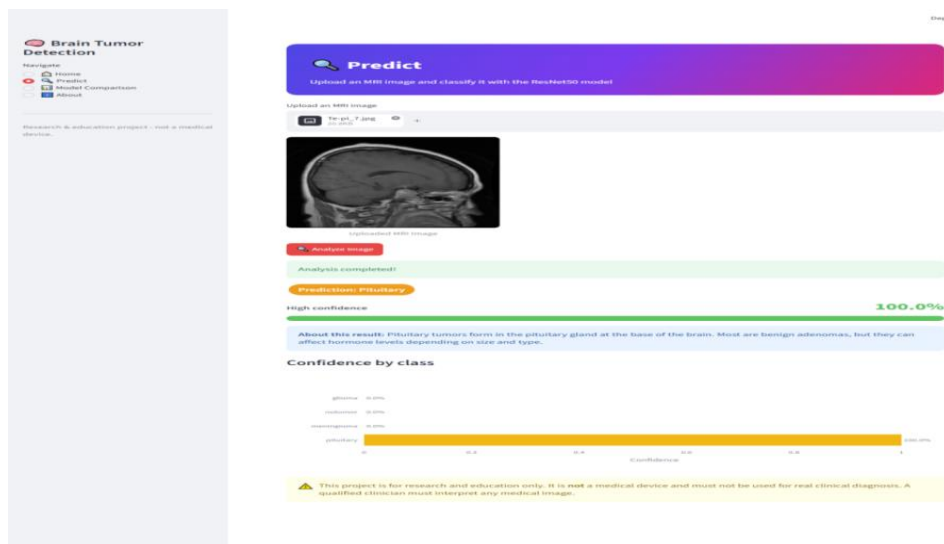


Figure: 9.1 Predict page showing a prediction with confidence score.

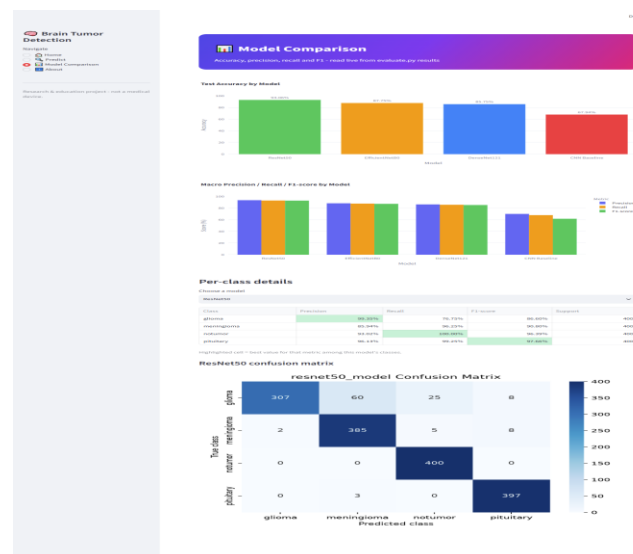


Figure: 9.2 Model Comparison page showing per-class metrics for ResNet50.

10. DISCUSSION

10.1 Comparison with Existing Literature

Because all three transfer-learning models were trained with the same image size, batch size, random seed, augmentation, and two-stage fine-tuning schedule, the accuracy differences observed in Table 4 can be attributed primarily to the architectures themselves rather than inconsistent experimental conditions — a controlled comparison that is not always present in the related work summarized in Table 1. Under this protocol, ResNet50 was the most accurate architecture, consistent with prior work that also reports strong ResNet50 performance on brain MRI classification [2], although other studies report DenseNet121 or EfficientNet variants performing best under different datasets and protocols [1, 4], underscoring that the best architecture for this task is dataset- and protocol-dependent rather than universally fixed. The relatively large gap between the CNN baseline (67.94%) and the transfer-learning models (85.75% – 93.06%) is consistent with the broader literature finding that ImageNet-pretrained features transfer well to medical imaging tasks even though natural images and MRI scans differ substantially in appearance, likely because low-level filters learned on ImageNet (edges, textures, shapes) remain useful general-purpose feature detectors.

10.2 Limitations

The proposed system was trained and evaluated on a single public Kaggle dataset, while MRI images from real hospital scanners can differ in format (e.g., DICOM), sequence type, contrast, and scanner calibration; accuracy on such out-of-distribution images cannot be assumed to match the results reported here without separate validation. The dataset split used is also image-level rather than confirmed patient-level, so related slices from the same patient could potentially appear in both the training and test sets, and recent literature has highlighted patient-level splitting as important to avoid overly optimistic accuracy estimates on this class of dataset. This system remains a research and education prototype that has not undergone any clinical validation, and therefore must not be used to inform real diagnostic or treatment decisions.

11. CONCLUSION

This paper presented a fair, controlled comparison of a CNN baseline and three ImageNet-pretrained architectures (DenseNet121, EfficientNetB0, ResNet50) for four-class brain MRI classification, trained under an identical two-stage transfer-learning protocol. ResNet50 achieved the highest test accuracy of 93.06% on a held-out set of 1,600 images, clearly outperforming the CNN baseline (67.94%) and the other transfer-learning models, EfficientNetB0 (87.75%) and DenseNet121 (85.75%). The best model was deployed in an interactive Streamlit application that reports a prediction along with a confidence score, making the system usable for demonstration and further experimentation. A consistent weakness in glioma recall was identified across all transfer-learning models, pointing to a specific, well-defined direction for future improvement. Overall, the study confirms that a rigorously controlled, identical training protocol is essential for drawing fair conclusions when comparing deep learning architectures for medical image classification.

12. FUTURE WORK

Building on the results and limitations discussed above, several directions can extend this work. Explainability techniques such as Grad-CAM could be used to visualize which image regions influence each prediction, particularly to investigate the lower glioma recall observed across all transfer-learning models. Combining multiple models in an ensemble may further improve overall accuracy and robustness beyond the best single model, ResNet50. Adding DICOM support and evaluating the models on MRI images from additional sources would help study generalization beyond the single Kaggle dataset used here. A strict patient-level train/validation/test split should also be verified or enforced to rule out any possibility of data leakage between related slices of the same patient. Finally, the dataset could be extended with additional tumor grades or multi-institution data, and the system could be explored for deployment as a mobile application, broadening its accessibility beyond the current web-based demonstration.

REFERENCES

- [1] Z. N. K. Swati et al., “Brain tumor classification for MR images using transfer learning and fine-tuning,” *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34-46, 2019.
- [2] D. M. Asadujjaman, “Transfer Learning Based Multiclass Brain Tumor Classification Using MRI Data,” *Proceedings of the 2025 IEEE International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*, 2025.
- [3] V. Vadde and W. Bukaita, “Comparative Evaluation of CNN and ResNet18 Architectures for MRI-Based Brain Tumor Classification Using Deep Learning,” *Medical Research Archives*, vol. 13, no. 12, 2025.
- [4] A. Golkarieh, S. R. Boroujeni, K. Kiashemshaki, M. Deldadehasl, H. Aghayarzadeh, and A. Ramezani, “Breakthroughs in Brain Tumor Detection: Leveraging Deep Learning and Transfer Learning for MRI-Based Classification,” 2025.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of CVPR*, pp. 770-778, 2016.
- [6] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *Proceedings of CVPR*, pp. 4700-4708, 2017.
- [7] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *Proceedings of ICML*, pp. 6105-6114, 2019.
- [8] M. Nickparvar, “Brain Tumor MRI Dataset,” Kaggle, 2021. Available: <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>

[9] TensorFlow/Keras Documentation, “Keras Applications and tf.keras API Reference,” Google, 2024. Available:

https://www.tensorflow.org/api_docs

[10] scikit-learn Documentation, “Metrics and scoring: classification_report, confusion_matrix,” scikit-learn developers, 2024. Available: https://scikit-learn.org/stable/modules/model_evaluation.html