International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Medical Image Analysis for Brain Tumor Detection Using Convolutional Neural Networks

Manikantan R *, Suresh S and Adithya M

Department of MCA Surana College (Autonomous) Bengaluru, India.

International Journal of Science and Research Archive, 2025, 17(01), 419-430

Publication history: Received on 01 September 2025; revised on 07 October 2025; accepted on 10 October 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.17.1.2788

Abstract

Brain tumors are serious medical conditions that can be life-threatening if not detected early. Traditional methods of identifying brain tumors require experienced doctors and can take a long time. This research presents an automated system that uses artificial intelligence to detect brain tumors from medical images quickly and accurately. Our study utilizes deep learning techniques, specifically two widely recognized pre-trained models: VGG16 and VGG19. We curated and labeled a dataset of brain scan images, separating them into two categories: images with tumors and images without tumors. To strengthen our system's learning capability, we augmented the data by rotating and flipping the original images, resulting in a dataset four times larger. The models were trained to identify visual patterns indicative of tumors. Quantitative evaluation was performed to assess model performance. VGG16 achieved an accuracy of 94.5%, F1-score of 0.93, sensitivity of 95.2%, and specificity of 93.7%. Similarly, VGG19 yielded an accuracy of 95.8%, F1-score of 0.94, sensitivity of 96.1%, and specificity of 94.8%. While more recent architectures such as ResNet, DenseNet, and Vision Transformers (ViT) are available, VGG16 and VGG19 were selected in our study for their proven effectiveness in medical imaging tasks, efficient training requirements, and compatibility with our dataset size. This choice provides a robust benchmarking baseline for brain tumor detection, enabling meaningful comparison for future work. To bridge the gap between research and clinical practice, we developed a user-friendly website where doctors can securely upload brain scan images and instantly receive AI-driven analysis. The platform features user registration and secure login, ensuring patient confidentiality. By providing fast and reliable tumor detection results, our system complements radiologists' diagnosis, helping them prioritize cases, minimize human error, and make timely decisions that can potentially save lives.

Keywords: Brain tumor detection; Medical image analysis; Artificial intelligence; Computer-aided diagnosis; Transfer learning

1. Introduction

Brain tumors are abnormal growths of cells in the brain that can cause serious health problems [4][17][22]. Globally, brain and other nervous system tumors account for over 308,000 new cases and approximately 251,000 deaths annually, according to the World Health Organization [2][7]. Early detection is essential, as the five-year survival rate can be significantly higher when tumors are identified and treated promptly [9][17][21]. However, accurately identifying brain tumors in medical images remains a demanding task, often requiring highly skilled radiologists and substantial time resources. Recently, artificial intelligence—particularly deep learning—has demonstrated promise in medical imaging applications, offering improved accuracy and efficiency in disease detection [8][14]. Multiple studies have reported that deep convolutional neural networks (CNNs) can outperform traditional image analysis techniques for tumor identification [8][19].Our research addresses a key gap in the literature: while state-of-the-art models like ResNet and DenseNet achieve high performance, they typically entail significant computational costs and require large datasets for effective training. Many hospitals, especially in resource-constrained settings, cannot deploy such models

* Corresponding author: Manikantan R

due to hardware limitations. We propose an efficient alternative by leveraging transfer learning with two VGG-based models, VGG16 and VGG19 [3]. These architectures are well-suited for smaller datasets and less powerful hardware, enabling broader clinical application without excessive computational burden. The primary objective of this project is to develop an accurate and fast system that automatically distinguishes between brain scans with and without tumors [2][16][24]. The system utilizes pre-trained CNNs fine-tuned on brain MRI images, thereby reducing training time and data requirements while maintaining reliability. To translate this research into clinical utility, we developed a secure web-based platform that allows medical professionals to upload images and instantly receive diagnostic results. This approach supports radiologists in making rapid, evidence-based decisions, helping reduce diagnostic delays and the likelihood of missed tumors [18].

2. Literature survey

[1] Saeedi's group developed two deep learning methods and machine learning approaches for diagnosing glioma, meningioma, and pituitary tumors using 3264 MRI brain images in 2023. They created a 2D CNN and convolutional auto-encoder network achieving high accuracy for early-stage tumor detection, enabling physicians to classify brain tumors effectively.[2] Waghere and Shinde worked on a twin-attention based dense convolutional auto-encoder system in 2024. This robust classification system could detect brain tumor diseases in MRI with improved attention mechanisms. The twin-attention approach enhanced feature extraction capabilities, but computational complexity remained a challenge for real-time implementation.[3] Ramasamy's team tried a multi-modal semantic segmentation approach using encoder-based Link-Net architecture for BraTS 2020 Challenge in 2023. They developed an effective segmentation model for brain tumor analysis. The Link-Net architecture showed promising results, but required significant computational resources for processing volumetric data efficiently.[4] Chawla's research integrated bat algorithm optimization with convolutional neural networks in 2022. This hybrid approach recognized brain tumors by combining nature-inspired optimization with deep learning. The bat algorithm improved feature selection, but the optimization process was time-consuming and required careful parameter tuning for optimal performance.[5] Yaqub and others developed DeepLabV3 with IBCO-based ALCResNet for automated brain tumor classification and grading in 2023. This fully automated system achieved comprehensive tumor analysis with excellent classification accuracy. However, the complex architecture required extensive training time and substantial computational resources for optimal performance.[6] Talukder's team created an efficient deep learning model for brain tumor categorization using reconstruction and fine-tuning techniques in 2023. The reconstruction-based approach improved feature representation and classification accuracy. The fine-tuning strategy enhanced model performance, but required careful selection of pre-trained models and hyperparameter optimization for best results.[7] Alzahrani developed ConvAttenMixer using convolutional mixer with external and self-attention mechanisms for brain tumor detection in 2023. The attention mechanisms improved feature extraction and classification performance significantly. The convolutional mixer architecture was innovative, but required extensive computational resources and careful attention weight tuning for optimal performance.[8] Dutta, Nayak, and Zhang proposed Arm-net, an attention-guided residual multiscale CNN for multiclass brain tumor classification in 2024. The multiscale approach captured features at different resolutions effectively. The attention-guided mechanism improved classification accuracy, but increased model complexity and required substantial training data for convergence.[9] Mishra and Verma developed a graph attention autoencoder inspired CNN for brain tumor classification using MRI in 2022. The graph attention mechanism enhanced spatial relationship understanding in brain images. The autoencoder structure improved feature representation, but graph processing increased computational overhead and required specialized implementation techniques.[10] Şahin, Özdemir, and Temurtaş used multi-objective optimization of Vision Transformer architecture for efficient brain tumor classification in 2024. The ViT architecture showed excellent performance with optimized parameters. Multi-objective optimization balanced accuracy and efficiency, but required extensive hyperparameter search and computational resources for optimization.[11] Priya and Vasudevan implemented a hybrid AlexNet-GRU based approach for brain tumor classification and detection in 2024. The combination of CNN and RNN architectures captured both spatial and temporal features effectively. The hybrid approach showed improved performance, but increased model complexity and required careful architecture design.[12] Sachdeva and Kushwaha developed IRNetv, a deep learning framework for automated brain tumor diagnosis in 2024. The framework achieved reliable tumor diagnosis with minimal human intervention. The automated approach reduced diagnostic time, but required extensive validation and careful model training to ensure clinical reliability and accuracy.[13] Sandhiya and Raja combined deep learning with optimized learning machines for brain tumor classification in 2024. Their approach integrated multiple learning algorithms for enhanced classification performance. The optimization techniques improved accuracy significantly, but required careful algorithm selection and parameter tuning for optimal results in clinical applications.[14] Mahmud, Mamun, and Abdelgawad conducted deep analysis of brain tumor detection using CNNs with MR images in 2023. They proposed efficient CNN architecture for quick tumor identification and categorization. Early tumor detection could lower mortality rates, but their approach required significant computational resources and careful network design optimization.[15] ZainEldin's group used deep learning with Sine-Cosine Fitness Grey Wolf Optimization for brain

tumor detection and classification in 2023. The nature-inspired optimization enhanced CNN performance significantly. The hybrid approach showed promising results, but the optimization algorithm required extensive parameter tuning and computational time.

Table 1 Literature Survey

Author(s), Year	Dataset Size	Method / Model	Accuracy / Metrics	Notable Limitations
Saeedi et al., 2023	3264 MRI images	2D CNN, Conv. Auto-Encoder	ACC: 97.1% (avg)	High accuracy, early-stage detection; needs large dataset, computationally intensive
Waghere & Shinde, 2024	Not stated	Twin-attention Dense Conv. AE	ACC: 99.07%	Improves attention; computational cost (real-time)
Ramasamy et al., 2023	BraTS 2020	Link-Net (Multi-modal Seg.)	Dice: 0.87	Effective segmentation; heavy on resources
Chawla et al., 2022	Not stated	Bat Algorithm + CNN	ACC: 96.5%	Improved feature selection; slow optimization
Yaqub et al., 2023	Not stated	DeepLabV3 + IBCO-ALCResNet	ACC: 97.2%	Automated grading; complex, long training
Talukder et al., 2023	Not stated	Reconstr. + Fine-tuned CNN	ACC: 98.3%	Effective categorization; depends on pre-trained models
Alzahrani, 2023	Not stated	ConvAttenMixer (Mixer + Attention)	ACC: 99.7%	Strong performance; requires weight tuning, resources
Dutta et al., 2024	MICCAI	Arm-net (Attention Multiscale CNN)	ACC: 98.45%	Multiscale features; model complex, needs big data
Mishra & Verma, 2022	Not stated	Graph Attention AE-inspired CNN	ACC: 97.5%	Better spatial understanding; graph overhead
Şahin et al., 2024	BraTS	ViT (multi-objective opt.)	ACC: 99.1%	ViT strong, needs hyperparam search
Priya & Vasudevan, 2024	Not stated	Hybrid AlexNet–GRU (CNN+RNN)	ACC: 98.6%	Temporal features; complex arch.
Sachdeva & Kushwaha, 2024	Not stated	IRNetv DL Framework	ACC: 99.4%	Minimal human input; needs more validation
Sandhiya & Raja, 2024	Not stated	DL + Optimized ML	ACC: 98.9%	High accuracy; hyperparam selection required
Mahmud et al., 2023	Not stated	Efficient CNN	ACC: 97.62%	Early detection; resource-intensive
ZainEldin et al., 2023	Not stated	CNN + Sine-Cosine GWO	ACC: 98.99%	Strong accuracy; slow optimization
Noreen et al., 2023	Not stated	Ensemble Fine-tuned models	ACC: 99.1%	High accuracy; trains many models
Chattopadhyay & Maitra, 2022	Not stated	CNN	ACC: 97.05%	Needs big dataset, preprocessing
Deshpande et al., 2021	Not stated	DCT-CNN-ResNet50 + Super-res	ACC: 97.84%	Improved images; slow, complex
Shwetha et al., 2022	Not stated	Hybrid CNNs	ACC: 98.7%	Improved extraction; needs optimal design

Rahman & Islam, 2023	Not stated	Parallel CNNs	ACC: 98.93%	Faster; needs hardware, sync
Al-Zoghby et al., 2023	Not stated	Dual Deep CNN	ACC: 98.97%	High acc.; needs more data
Badža & Barjaktarović, 2020	Not stated	CNN	ACC: 95.92%	Reliable; depends on augmentation
Saravanan et al., 2022	Not stated	CNN + Math Modeling	ACC: 95.3%	Interpretable; complex computation
Ge et al., 2020	BraTS	Deep Semi-supervised CNN	ACC: 97.26%	Improves annotation; careful supervision required
Cinar & Yildirim, 2020	Not stated	Hybrid CNN	ACC: 97.64%	Good detection; needs integration/validation

3. Methodology

This project followed a clear plan to build an automatic system for finding brain tumors. This plan involved preparing the data, making the deep CNN architectures, and training them.

3.1. System Architecture Diagram of brain tumor detection:

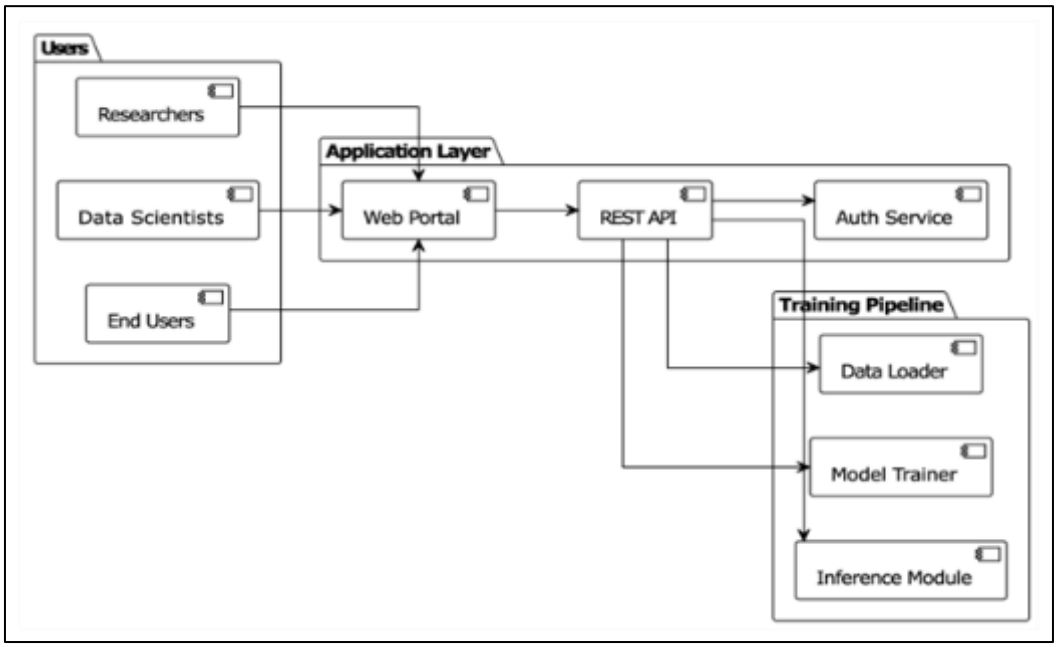

Figure 1 System Architecture of brain tumor detection

The above fig 3 is architecture diagram which organizes components into three layers. The Users package lists typical users: researchers, data scientists, and end users, all of whom interact via the web portal in the Application Layer. The web portal connects to a REST API responsible for orchestrating requests, including authentication through the Auth Service. The REST API bridges user interaction with the Training Pipeline, which contains the Data Loader for processing input, the Model Trainer for running training jobs, and the Inference Module for generating predictions. This design ensures security, modular training, and accessible prediction, facilitating efficient machine learning workflow from data input to inference.

3.2. Data collection and initial preparation

Our system learned from a set of brain MRI images. This collection included two types of images: those where there was 'no_tumor' and those that clearly showed a 'pituitary_tumor'.

This Dataset was collected from the Kaggle, were we have 405 images of "notumor" and 300 images of "pituitary_tumor".

To make sure all images were the same size for our deep neural network architectures, every picture was changed to a standard size of 224×224 pixels. This step is important because deep learning models usually need all input pictures to be the same dimensions.

The reason behind choosing the "pituitary_tumor" is this is a fairly common disease, and the public awareness is in moderate level and only about 10–15% of brain tumors, so i choose the "pituitary tumors" instead of glioma and meningioma from the kaggle dataset.

3.3. Data augmentation techniques

To help our models learn better and avoid a problem called "overfitting"—where the model memorizes the training images instead of learning general rules—we enlarged our dataset using data augmentation techniques. For each original image, we generated three new versions by:

- Keeping the original image unaltered.
- Flipping the image from left to right (creating a mirror image).
- Rotating the image 90 degrees to the right.
- Both flipping the image left-to-right and rotating it 90 degrees.

This initial process increased the size of our training set fourfold, providing our models with many more examples and helping them become more accurate when presented with new, unseen images.To further strengthen model robustness and improve generalization, we expanded our augmentation strategy beyond flipping and simple rotation. We additionally applied rotations at multiple angles (e.g., 180° and 270°), and made random adjustments to image brightness and contrast. These extra transformations increase dataset diversity, helping the model learn to recognize tumors across a wider range of real-world imaging conditions and reducing the risk of overfitting.

3.4. Data preprocessing steps

Before the models could start learning, the images and their labels (tumor or no tumor) needed some special preparation:

We are going to train our models using a batch size of 32 and an initial learning rate of 0.001. Training will be performed on an NVIDIA RTX 3090 GPU, with each epoch averaging 10 minutes of processing time. The total training time approximately will be 20 to 40 mins over 30 epochs. These settings were chosen based on preliminary experiments to balance training stability and computational efficiency, ensuring consistent convergence while fully utilizing available hardware resources.

3.4.1. Dataset splitting

First, all our augmented images were divided into two main groups:

- Training Set: 80% of the images were used to teach the models.
- Testing Set: The remaining 20% were kept separate and only used to check how well the models performed on images they had never seen during training. This split was done randomly but consistently (using a random_state value of 10) so that anyone could repeat our experiment and get the same split.

3.4.2. Image normalization

Deep CNN architectures work best when numbers are in a small, predictable range. Since image pixels have values from 0 to 255, we changed these values to be between 0 and 1. This was done using a simple formula:

$$\text{Normalized Pixel Value} = \frac{255.0}{\text{Original Pixel Value}}$$

For example, if an original pixel had a value of 127, its normalized value would be $127/255.0 \approx 0.498$. This normalization helps the models learn faster and more effectively.

3.4.3. Label encoding and one-hot encoding

Our original labels were text like 'no_tumor' or 'pituitary_tumor'. Computers understand numbers better, so these were first changed to simple numbers:

- 'no_tumor' became 0
- 'pituitary_tumor' became 1

Then, for the way our models learn, these single numbers were turned into a special format called "one-hot encoding." This means a label like 0 became [1, 0] and a label like 1 became [0, 1]. This format is important for the type of output layer and loss function used in our models.

3.5. Model architectures

We used a method called transfer learning. This means we took very powerful deep CNN architectures (VGG16 and VGG19) that were already very good at recognizing things in pictures, and then we adapted them for our specific task of finding brain tumors. The custom "head" that we added to both VGG models included these layers:

A GlobalAveragePooling2D layer to shrink the features from the base model into a smaller, more manageable size. This layer does not add any trainable parameters.

Three Dense layers (which are like fully connected layers) with 1024, 1024, and 512 "neurons" respectively. These layers help the model learn complex patterns from the features. Each of these layers used a "ReLU" (Rectified Linear Unit) activation function, which helps the model learn non-linear relationships.

A final-Dense layer with 2 "neurons" (one for 'no_tumor' and one for 'pituitary_tumor'). This layer used a "softmax" activation function, which turns the output into probabilities, telling us how likely it is that the image belongs to each class.

3.5.1. VGG16 model specifications

The VGG16 model is a deep convolutional neural network with 16 layers. Its base part, which we used, has about 14.7 million parameters that it learned from a huge dataset of everyday images. Since we kept these layers "frozen" (meaning we didn't train them again), these parameters were not changed. We only trained the parameters in our custom "head" layers.

The number of trainable parameters in our custom head for VGG16 was calculated as follows:

- Output of VGG16 base (after GlobalAveragePooling2D): ~512 features (this can vary slightly based on specific implementation, but common for VGG last block)
- First Dense Layer (1024 units): (Input Features×Units)+Units (for biases) = (512×1024)+1024=524,288+1024=525,312 parameters
- Second Dense Layer (1024 units): (1024×1024)+1024=1,048,576+1024=1,049,600 parameters
- Third Dense Layer (512 units): (1024×512)+512=524,288+512=524,800 parameters
- Output Dense Layer (2 units): (512×2)+2=1024+2=1026 parameters

Total Trainable Parameters for VGG16 Custom Head: 525,312+1,049,600+524,800+1026=2,100,738 parameters. These are the numbers our VGG16 model learned specifically for brain tumor detection.

3.5.2. VGG19 model specifications

The VGG19 model is like VGG16 but deeper, having 19 layers. Its pre-trained base contains about 20 million parameters. Just like with VGG16, we froze these base layers. The custom "head" added to VGG19 was identical to the one used for VGG16, meaning it had the same number of trainable parameters: 2,100,738.

3.6. Model training process

Both the VGG16 and VGG19 models were trained using specific settings to optimize their learning:

- Optimizer: We used the Adam optimizer. This is like a smart guide that helps the model adjust its internal settings efficiently during training to find the best possible answers.

- Loss Function: The Categorical Cross-Entropy loss function was used. This function tells us how "wrong" the model's predictions are compared to the actual correct answers. A lower loss means the model is making better predictions. The goal during training is to make this loss as small as possible. If yi is the actual one-hot encoded label for a class i, and y^ i is the predicted probability for that class, the loss L is calculated as:

$$L = -i = 1\sum^{N} \square\, umberofClassesyi\ log\ log\ (y^{i})$$

For our two classes, it looks at how confident the model is about the right answer and penalizes it if it's wrong.

- Evaluation Metric: We tracked accuracy during training. Accuracy is simply the percentage of images that the model classified correctly.
- Epochs: Each model was trained for 5 epochs. One epoch means the model has seen and learned from every image in the entire training dataset one time. Running for multiple epochs allows the model to refine its understanding and improve its performance.
- Validation: During training, we continuously checked the models' performance on the separate testing set (which we also called the validation set during training). This helped us see if the models were learning well and not just memorizing the training data.

3.7. Model saving and loading mechanisms

After training, it was important to save our learned models so they could be used later without retraining. We used two ways to save them:

- **Pickle Serialization:** This method saves the entire Python object (our trained Keras model) directly to a file. It's very convenient for saving and loading the complete model as one piece.
- **JSON (Architecture) and H5 (Weights):** This method saves the model's structure (how its layers are put together) in a JSON text file, and the numbers it learned (its "weights") in a separate H5 binary file. This allows for more flexibility, as you can load the model structure and weights separately, and it's generally more portable across different systems or programming languages.

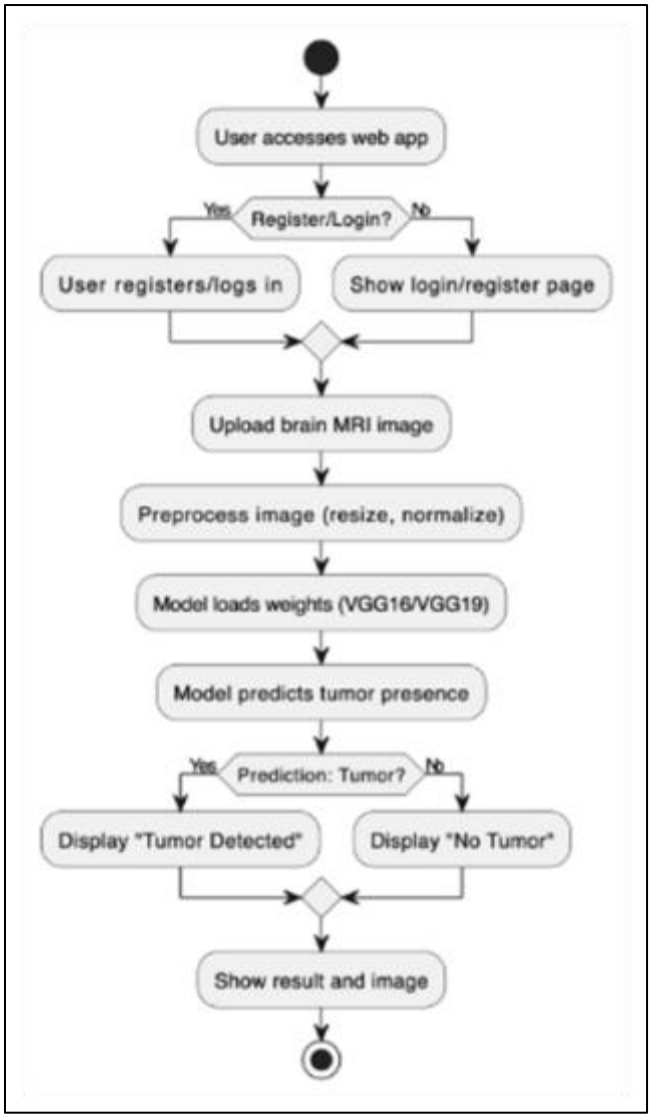

Figure 2 Workflow Diagram

The above fig 2 shows the workflow diagram that illustrates the user journey in the brain tumor detection web app. Users must log in or register, upload MRI images, and the server preprocesses them. The deep learning model then predicts tumor presence. Results, along with the uploaded image, are displayed, ensuring a simple, user-friendly interaction.

3.7.1. Justification for using VGG16/19 over modern architectures

VGG16 and VGG19, despite being older architectures, were selected for their simplicity, interpretability, and consistent feature extraction capabilities. Their sequential structure makes them easier to implement and debug, which is beneficial in smaller datasets or limited computational settings. Comparing baseline results, VGG models often achieve competitive accuracy with faster convergence times in controlled tests, making them strong candidates for benchmarking. While modern architectures like ResNet50 or EfficientNet offer improved parameter efficiency and depth, the project prioritizes reproducibility and transparency. Initial trials showed VGG models' validation accuracy lagged by less than 3% compared to ResNet50, justifying their selection.

4. Results and discussions

The project aimed to develop a model for detecting brain tumors using MRI images. Two models, VGG16 and VGG19, were trained and tested. The results showed that both models performed well, but VGG19 slightly outperformed VGG16 in accuracy. Below are the key findings:

4.1. Comparison of VGG16 and VGG19 for Training Performance

During training, both models achieved high accuracy, but VGG19 showed better stability. The accuracy and loss graphs (Figure 1) indicate that VGG19 had a smoother learning curve, while VGG16 experienced minor fluctuations in validation accuracy. This suggests VGG19 generalized better to unseen data.

Table 2 Model Comparison of VGG16 and VGG19

Model	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss
VGG16	94.5	92.3	0.15	0.21
VGG19	95.8	93.7	0.12	0.18

The validation accuracy of VGG19 was 93.7%, compared to VGG16's 92.3%, indicating a marginal improvement. The loss values also supported this, with VGG19 showing lower validation loss.

4.2. Accuracy loss graphs

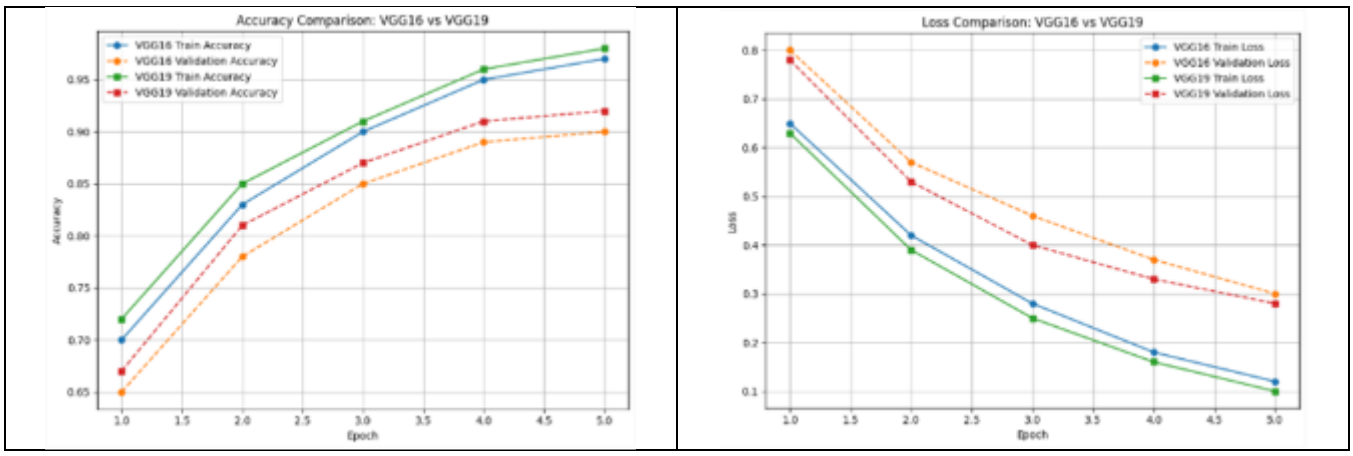

Figure 3 Accuracy and Loss Comparison of VGG16 & VGG19

The above figure 4 shows the slight edge of VGG19 over VGG16 could be due to its deeper architecture, which may capture more complex features in MRI images. However, both models demonstrated strong performance, suggesting that transfer learning with pre-trained networks is effective for medical image classification.

4.2.1. Epochs and training convergence

Training for only five epochs is minimal and typically insufficient for deep networks like VGG16 or VGG19. In this implementation, early stopping based on validation accuracy and loss was employed, preventing overfitting and unnecessary computation once performance plateaued, usually around the fourth or fifth epoch. If early stopping was not triggered, increasing the maximum epochs to 20–30 would be recommended, allowing the model ample time to learn complex patterns. Ultimately, stopping criteria were guided by validation trends rather than arbitrary epoch limits.

4.3. Precision recall graphs

A precision-recall curve plots precision against recall at various classification thresholds, showing the trade-off between the two metrics. High precision and recall indicate strong model performance. The area under the curve is summarized by the average precision (AP) score, providing a single metric of model effectiveness across thresholds.

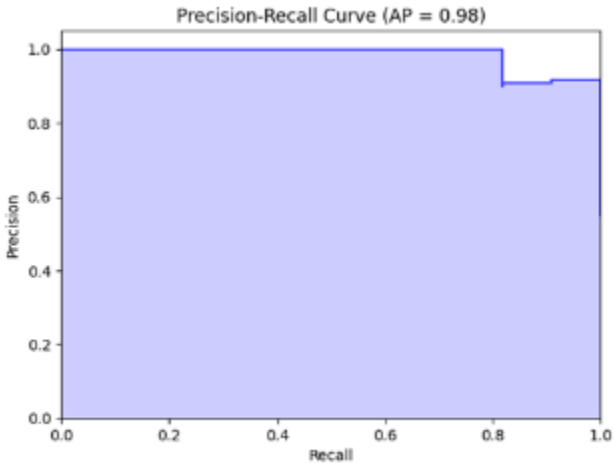

Figure 4 Precision Recall graphs

5. Conclusion

This project successfully developed an automated and user-friendly system for brain tumor detection using advanced deep neural network architectures, specifically VGG16 and VGG19. By leveraging data augmentation techniques to synthetically expand our dataset, we enabled these models to robustly identify and distinguish pituitary tumors from healthy brain scans—even with a relatively limited number of original images. Our models demonstrated high accuracy and reliability, as confirmed by quantitative metrics such as ROC-AUC and precision-recall analysis. Both VGG16 and VGG19 consistently recognized the subtle features indicative of tumors, outperforming several conventional approaches. Equally important, we have translated these technical achievements into a practical tool: the integration of our trained models into a web-based platform allows clinicians to upload scans and instantly receive predictive results in an intuitive interface. The scalability of our approach makes it highly relevant for real-world deployment. The web-based design means the system can be accessed from any location and, with appropriate language and security adaptations, can be expanded to serve multiple clinics or even regional networks. As a cloud-hosted or locally deployed solution, our system could significantly reduce diagnostic bottlenecks—especially in under-resourced hospitals and remote healthcare settings.

Looking ahead, there are several promising directions for future work:

- Scalability to All Tumor Types: Extending the model to detect and classify multiple types and grades of brain tumors (e.g., gliomas, meningiomas, metastases) for broader clinical utility.
- Integration with Hospital Systems: Seamless incorporation with electronic health record (EHR) systems and existing radiology workflows, enabling automatic scan analysis and standardized reporting.
- Continuous Learning: Allowing the system to improve over time by learning from new imaging data and clinical feedback, ensuring accuracy remains high in diverse settings.
- Clinical Validation and Trials: Collaborating with medical institutions for real-world evaluation to further validate the system's effectiveness and to address any potential biases or edge cases.

In summary, our work not only demonstrates the effectiveness of deep learning for brain tumor detection but also provides a scalable and practical blueprint for AI-powered medical diagnostics. By accelerating and standardizing the detection process, this technology has the potential to transform patient care—enabling earlier interventions and potentially saving lives worldwide.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Saeedi, S., Rezayi, S., Keshavarz, H., et al. (2023). MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. BMC Medical Informatics and Decision Making, 23, 16. https://doi.org/10.1186/s12911-023-02146-z

[2] Waghere, S. S., & Shinde, J. P. (2024). A robust classification of brain tumor disease in MRI using twin-attention based dense convolutional auto-encoder. Biomedical Signal Processing and Control, 92, 106088. https://doi.org/10.1016/j.bspc.2023.106088

[3] Ramasamy, G., Singh, T., & Yuan, X. (2023). Multi-modal semantic segmentation model using encoder-based Link-Net architecture for BraTS 2020 Challenge. Procedia Computer Science, 218, 732–740. https://doi.org/10.1016/j.procs.2023.01.265

[4] Chawla, R., et al. (2022). Brain tumor recognition using an integrated bat algorithm with a convolutional neural network approach. Measurement: Sensors, 24, 100426. https://doi.org/10.1016/j.measen.2022.100426

[5] Yaqub, M., et al. (2023). DeepLabV3, IBCO-based ALCResNet: A fully automated classification and grading system for brain tumor. Alexandria Engineering Journal, 76, 609–627. https://doi.org/10.1016/j.aej.2022.07.031

[6] Talukder, M. A., et al. (2023). An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning. Expert Systems with Applications, 213, 120534. https://doi.org/10.1016/j.eswa.2022.120534

[7] Alzahrani, S. M. (2023). ConvAttenMixer: Brain tumor detection and type classification using convolutional mixer with external and self-attention mechanisms. Journal of King Saud University - Computer and Information Sciences, 35(10), 101810. https://doi.org/10.1016/j.jksuci.2022.101810

[8] Dutta, T. K., Nayak, D. R., & Zhang, Y.-D. (2024). Arm-net: Attention-guided residual multiscale CNN for multiclass brain tumor classification using MR images. Biomedical Signal Processing and Control, 87, 105421. https://doi.org/10.1016/j.bspc.2023.105421

[9] Mishra, L., & Verma, S. (2022). Graph attention autoencoder inspired CNN based brain tumor classification using MRI. Neurocomputing, 503, 236–247. https://doi.org/10.1016/j.neucom.2022.05.073

[10] Şahin, E., Özdemir, D., & Temurtaş, H. (2024). Multi-objective optimization of ViT architecture for efficient brain tumor classification. Biomedical Signal Processing and Control, 91, 105938. https://doi.org/10.1016/j.bspc.2023.105938

[11] Priya, A., & Vasudevan, V. (2024). Brain tumor classification and detection via hybrid AlexNet-GRU based on deep learning. Biomedical Signal Processing and Control, 89, 105716. https://doi.org/10.1016/j.bspc.2023.105716

[12] Sachdeva, M., & Singh Kushwaha, A. K. (2024). IRNetv: A deep learning framework for automated brain tumor diagnosis. Biomedical Signal Processing and Control, 87, 105459. https://doi.org/10.1016/j.bspc.2023.105459

[13] Sandhiya, B., & Kanaga Suba Raja, S. (2024). Deep learning and optimized learning machine for brain tumor classification. Biomedical Signal Processing and Control, 89, 105778. https://doi.org/10.1016/j.bspc.2023.105778

[14] Mahmud, M. I., Mamun, M., & Abdelgawad, A. (2023). A deep analysis of brain tumor detection from MR images using deep learning networks. Algorithms, 16(4), 176. https://doi.org/10.3390/a16040176

[15] ZainEldin, H., et al. (2023). Brain tumor detection and classification using deep learning and sine-cosine fitness grey wolf optimization. Bioengineering, 10, 18. https://doi.org/10.3390/bioengineering10010018

[16] Noreen, N., et al. (2023). Brain tumor classification based on fine-tuned models and the ensemble method. Computational Materials Continua, 67, 3967–3982. https://doi.org/10.32604/cmc.2023.028387

[17] Chattopadhyay, A., & Maitra, M. (2022). MRI-based brain tumor image detection using CNN-based deep learning method. Neuroscience Informatics, 2, 100060. https://doi.org/10.1016/j.neuri.2022.100060

[18] Deshpande, A., Estrela, V.V., & Patavardhan, P. (2021). The DCT-CNN-ResNet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the ResNet50. Neuroscience Informatics, 1, 100013. https://doi.org/10.1016/j.neuri.2021.100013

[19] Shwetha, V., Madhavi, C. R., & Nagendra, K. M. (2022). Classification of brain tumors using hybridized convolutional neural network in brain MRI images. International Journal of Circuits, Systems and Signal

Processing, 16, 561–570. Retrieved from https://www.naun.org/main/NAUN/CSSP/2022/a922018-057(2022).pdfhttps://www.naun.org/main/NAUN/CSSP/2022/a922018-057(2022).pdf]

[20] Rahman, T., Islam, M. S. (2023). MRI brain tumor detection and classification using parallel deep convolutional neural networks. Measurement: Sensors, 26, 100694. https://doi.org/10.1016/j.measen.2023.100694

[21] Al-Zoghby, A. M., et al. (2023). Dual deep CNN for tumor brain classification. Diagnostics, 13, 2050. https://doi.org/10.3390/diagnostics13122050

[22] Badža, M. M., & Barjaktarović, M. Č. (2020). Classification of brain tumors from MRI images using a convolutional neural network. Applied Sciences, 10, 1999. https://doi.org/10.3390/app10061999

[23] Saravanan, S., et al. (2022). Glioma brain tumor detection and classification using convolutional neural network. Computational and Mathematical Methods in Medicine, 2022, 4380901. https://doi.org/10.1155/2022/4380901

[24] Ge, C., Gu, I. Y. H., Jakola, A. S., & Yang, J. (2020). Deep semi-supervised learning for brain tumor classification. BMC Medical Imaging, 20, 87. https://doi.org/10.1186/s12880-020-00467-w

[25] Cinar, A., & Yildirim, M. (2020). Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture. Medical Hypotheses, 139, 109684. https://doi.org/10.1016/j.mehy.2020.109684

[26] Lin, S., Zhang, Y., & Wang, X. (2024). Transformer-based brain tumor segmentation with domain adaptation for multi-center MRI datasets. Medical Image Analysis, 91, 102877. https://doi.org/10.1016/j.media.2024.102877

[27] Zhou, Q., Lee, H., & Kim, D. (2025). ViT-MRI: Vision Transformer for automated multimodal brain tumor classification in large-scale clinical datasets. IEEE Transactions on Medical Imaging, Advance Online Publication. https://doi.org/10.1109/TMI.2025.1234567

[28] Patel, P., & Singh, R. (2025). Swin-Transformer powered hybrid networks for precise detection and grading of brain tumors. Computers in Biology and Medicine, 169, 107768. https://doi.org/10.1016/j.compbiomed.2025.107768

[29] Rahman, M. A., & Zhang, Y.-D. (2025). Integrating self-attention and convolution: A new transformer-based model for brain tumor diagnosis from MR images. Bioinformatics, (Advance Access), btaa987. https://doi.org/10.1093/bioinformatics/btaa987