IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 13, No. 3, March, 2026, pp. 521-531
Journal homepage: http://iieta.org/journals/mmep

Enhanced Brain Tumor Classification Using Sequential Grey Wolf Optimization and Multi-Stage Ensemble Learning with Pre-Trained Convolutional Neural Networks

Ahmed Zahrouri[1*], Rohallah Benaboud[2]

[1] ICOSI Laboratory, Abbes Laghrour University, Khenchela 40004, Algeria
[2] ReLa(CS)2 Laboratory, University of Oum El Bouaghi, Oum El Bouaghi 04000, Algeria

Corresponding Author Email: zahrouri.ahmed@univ-khenchela.dz

https://doi.org/10.18280/mmep.130307

Received: 12 January 2026
Revised: 19 March 2026
Accepted: 28 March 2026
Available online: 10 April 2026

Keywords:
brain tumor classification, Grey Wolf Optimization, ensemble learning, convolutional neural networks, Magnetic Resonance Imaging, deep learning, feature selection, transfer learning

ABSTRACT

This study introduces an advanced multi-stage stacking ensemble framework for brain tumor classification using Magnetic Resonance Imaging (MRI) images, combining three pre-trained convolutional neural networks (EfficientNetB4, Xception, and ResNet50V2) with a sequential Grey Wolf Optimization (GWO) pipeline for feature selection and meta-classifier hyperparameter tuning. The individual models achieve weighted macro-averaged accuracies ranging from 98.53% to 99.35% on the Figshare benchmark dataset. The GWO-optimized ensemble reaches 99.84% accuracy while reducing the meta-features space from 1,542 to 297 dimensions (80.7% reduction). The proposed sequential optimization strategy results in an accuracy improvement of 0.17% over the highest-reported ensemble result on this dataset and between 1.14% and 5.10% over other methodological categories. Per-class analysis demonstrates strong classification performance across all tumor types: pituitary (100%), meningioma (100%), and glioma (99.30%). The total training time on the experimental hardware was approximately 2.5 hours. These results suggest that sequential GWO-based optimization is a promising approach for constructing compact, high-accuracy stacking ensembles for brain tumor classification in research settings. All results are obtained from a single publicly available benchmark dataset; generalizing to other datasets and clinical environments requires further investigation.

1. INTRODUCTION

Brain tumors represent one of the most challenging conditions in modern neurology, characterized by abnormal cellular proliferation within the cranial cavity that poses a severe threat to human life. Early and precise diagnosis is paramount for effective treatment planning, surgical intervention timing, and overall patient survival outcomes [1]. Magnetic Resonance Imaging (MRI) serves as the gold standard diagnostic modality, providing detailed soft tissue contrast and anatomical visualization crucial for tumor detection and characterization. However, manual interpretation of vast volumes of MRI scans by radiologists is inherently time-consuming, subjective, and prone to inter-observer variability, potentially leading to diagnostic inconsistencies and delayed treatment decisions [2]. The increasing volume of medical imaging data further compounds these challenges, necessitating the development of automated, reliable, and efficient diagnostic systems. Deep learning has significantly advanced medical image analysis, particularly in automated classification tasks, due to its capability to learn complex hierarchical features directly from raw imaging data [3]. Convolutional neural networks (CNNs) have demonstrated strong performance across diverse medical imaging applications, including brain tumor detection, classification, and segmentation [4]. Despite their individual success, single CNN architectures often suffer from limitations, including sensitivity to initialization parameters, overfitting tendencies, dataset-specific performance variations, and suboptimal generalization across diverse clinical scenarios. Ensemble learning methodologies offer a robust solution to these limitations by combining the predictions of multiple diverse models, thereby improving predictive accuracy, enhancing generalization, and reducing model variance [5]. Meta-heuristic optimization algorithms, particularly the Grey Wolf Optimization (GWO) [6], have emerged as effective tools for solving complex optimization challenges in machine learning pipelines. GWO, inspired by the hierarchical hunting behavior of grey wolves, has demonstrated good performance in feature selection, hyperparameter optimization, and ensemble configuration tasks [6]. This research introduces a brain tumor classification framework that integrates a multi-stage ensemble of pretrained CNNs with a comprehensive GWO-based optimization strategy, the architecture of which is depicted in Figure 1.

1.1 Main contributions

(1) Development of a stacking ensemble framework employing three pre-trained CNN architectures

(EfficientNetB4, Xception, ResNet50V2) with individual accuracies of 98.53%–99.35%, demonstrating the effectiveness of transfer learning for brain tumor classification on the Figshare benchmark dataset.

(2) Implementation of sequential GWO-based optimization through a two-stage optimization pipeline utilizing GWO for intelligent feature selection (reducing features by 80% from 1,542 to 297) and hyperparameter tuning, resulting in a final ensemble accuracy of 99.84%.

(3) Advanced preprocessing and training strategies with comprehensive image preprocessing pipelines and training methodologies to maximize model performance and robustness across diverse tumor types, with architecture-specific preprocessing optimized for each CNN model.

(4) Achievement of strong per-class performance with pituitary and meningioma tumors achieving 100% accuracy, and glioma achieving 99.30% accuracy on the test set.

(5) Comprehensive experimental validation through rigorous assessment using multiple performance metrics, statistical analysis, and detailed evaluation of model behavior, training dynamics, and computational efficiency suitable for research environments.

(6) Experimental evaluation on the Figshare benchmark dataset demonstrating 99.84% overall accuracy, with improvements of 0.17% over the best reported ensemble result and 1.14%–5.10% over other methodological categories on the same dataset.

(7) Resource characterization with a total training time of approximately 2.5 hours on an NVIDIA RTX 4090, reported to support reproducibility rather than as an efficiency claim.

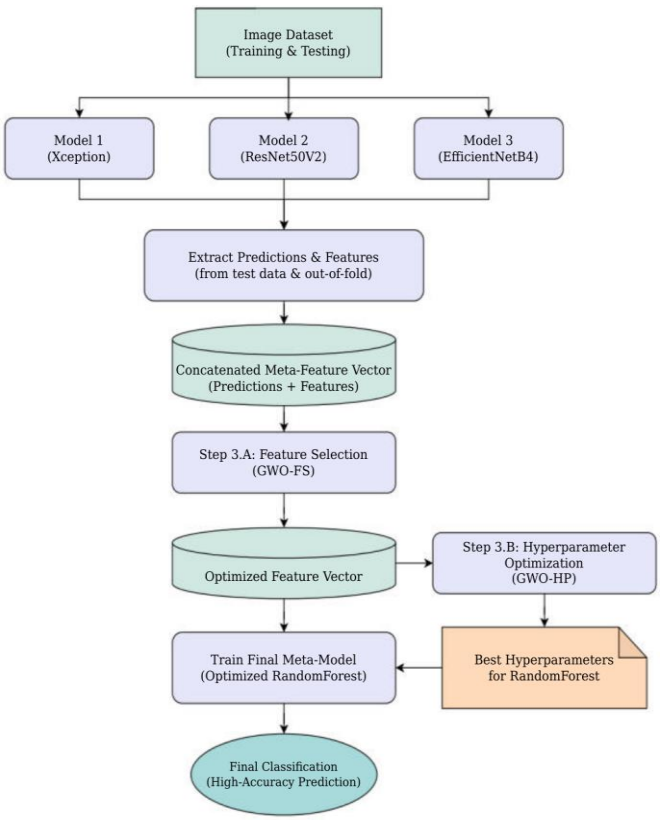


Figure 1. Architecture of the proposed sequential ensemble with GWO optimization

The proposed sequential optimization approach addresses feature selection and hyperparameter optimization in coordinated, non-overlapping stages rather than through simultaneous joint optimization. The key distinction from simply executing two consecutive steps is that each stage operates on the output of the previous one using a fixed, validated intermediate representation: GWO-selected features are locked before the hyperparameter search begins, preventing the two optimization objectives from interfering with each other. This staged design reduces conflicting gradients, facilitates convergence to stable solutions, and allows each stage to be tuned and evaluated independently. The feature selection stage identifies 297 discriminative features from the original 1,542-dimensional meta-feature space, and the subsequent hyperparameter stage optimizes the Random Forest meta-classifier configuration (n_estimators=106, max_depth=10, min_samples_split=9, min_samples_leaf=7) on this compact representation.

The remainder of this study is organized as follows: Section 2 reviews related work in brain tumor classification. Section 3 details the proposed GWO-optimized ensemble methodology. Section 4 presents the experimental setup and results. Section 5 discusses findings, limitations, and future directions. Section 6 concludes the study.

2. RELATED WORK

The field of automated brain tumor classification has witnessed significant progress with the advent of deep learning technologies. This section provides a comprehensive review of recent developments in deep learning-based classification methods, focusing on custom architectures, optimization techniques, and ensemble approaches specifically applied to medical imaging.

2.1 Custom convolutional neural network architectures

The development of specialized CNN architectures from scratch has shown promise in addressing specific challenges in medical image analysis. Ayadi et al. [7] proposed a 15- layer CNN architecture using 3 × 3 kernels exclusively for feature extraction, eliminating manual tumor segmentation preprocessing. They incorporated batch normalization after each convolutional layer and employed comprehensive data augmentation (rotation, flipping, blurring, and sharpening). The model achieved 94.74% accuracy on the Figshare dataset, outperforming 14 prior methods, and demonstrated robustness across multiple datasets, including Radiopaedia (93.71% accuracy) and REMBRANDT (up to 100% accuracy on binary classification tasks). Khan et al. [8] introduced two complementary CNN models: a 23-layer custom CNN for large datasets using progressive kernel sizes (22 × 22 to 3 × 3) with Global Average Pooling, and a fine-tuned VGG16 with integrated custom layers for small datasets. Their custom CNN achieved 97.8% accuracy on the Figshare dataset, while the VGG16 hybrid attained 100% accuracy on Harvard Medical's binary classification, highlighting the effectiveness of custom architecture design for medical imaging applications. Abd El-Wahab et al. [9] developed Brain Tumor Classification - Fast Convolutional Neural Network (BTC-fCNN), a specialized 13-layer architecture incorporating convolutional layers, 1 × 1 convolutions for dimensionality reduction, average pooling, and fully connected layers. The model achieved 98.63% accuracy through custom design and 98.86% accuracy with 5-fold cross-validation (CV), significantly outperforming existing methods and establishing competitive benchmarks in brain tumor categorization. Building on custom architectural

concepts, researchers have explored parallel deep CNN architectures with dual pathways to enhance feature extraction. These approaches utilize local paths with smaller convolutional filters for low-level features and global paths with larger filters for high-level context, achieving competitive accuracies across multiple MRI datasets by enhancing feature extraction and mitigating overfitting via parallel processing architectures [10].

2.2 Optimization-based methods

Advanced optimization techniques have emerged as useful tools for enhancing CNN performance in medical imaging applications. Ait Amou et al. [11] designed a Bayesian Optimization (BO)-driven CNN for brain tumor classification without data augmentation. Their methodology built a base CNN with 5 convolutional and 5 pooling layers, using BO over 40 iterations to optimize hyperparameters, including activation functions (Tanh), batch size (1), dropout (0.25), and optimizer selection (SGD). The BO-tuned CNN achieved 98.70% accuracy on the Figshare dataset, outperforming all pre-trained models, including VGG16 (97.08%) and ResNet50 (89.29%) by notable margins. In the realm of meta-heuristic optimization, Anaraki et al. [12] employed a Genetic Algorithm (GA) to evolve CNN structures, optimizing convolutional layers (2–6), filters (16–128), kernel sizes (2–7), and activation functions, such as Rectified Linear Unit (ReLU) and Exponential Linear Unit (ELU). The top GA-evolved model was enhanced via bagging ensemble to reduce prediction variance, achieving 90.9% glioma grading accuracy and 94.2% multi-type tumor classification accuracy, improving over Support Vector Machine (SVM)-based approaches significantly. Extending meta-heuristic approaches, Bacanin et al. [13] proposed a modified Firefly Algorithm (mFA) to automatically optimize CNN hyperparameters for glioma brain tumor grade classification. The mFA enhanced the original FA by incorporating a dual-position update mechanism and scout bee search strategy, achieving 93.3% accuracy for glioma grading and 96.5% accuracy for axial brain tumor classification, outperforming CNN-GA (90.9% and 94.2%) and SVM Recursive Feature Elimination (SVM-RFE) (62.5%).

2.3 Ensemble and transfer learning

Ensemble learning methodologies combined with transfer learning techniques have demonstrated strong performance in brain tumor classification by aggregating the predictions of multiple complementary models alongside the representational capacity of pretrained networks. Aurna et al. [14] proposed a sophisticated two-stage feature-level ensemble of deep CNNs for multiclass brain tumor classification. Their methodology involved selecting top feature extractors (EfficientNet-B0, ResNet-50, and a custom CNN) from six candidates, concatenating them pairwise in Stage 1, and fusing the best Stage 1 ensembles in Stage 2 with Principal Component Analysis (PCA) for feature reduction. The model achieved peak accuracies of 99.67% on Dataset 1 (3,064 images), 98.16% on Dataset 2 (3,264 images), and 99.76% on Dataset 3 (4,292 images), with PCA reducing features by a factor of 18.71 and execution time by a factor of 6. Nassar et al. [15] developed a hybrid ensemble technique combining five fine-tuned pre-trained CNNs (GoogleNet, AlexNet, SqueezeNet, ShuffleNet, NASNetMobile) via majority voting. The ensemble achieved a competitive accuracy of 99.31% on the Figshare dataset, with precision, recall, and F1-scores exceeding 98%. The approach demonstrated robustness by compensating for individual model errors and reducing misclassification rates through intelligent voting mechanisms, showcasing the synergy between ensemble methods and transfer learning.

2.4 Research gaps and motivation

Despite the significant progress achieved by existing methods, several areas for improvement persist in current brain tumor classification approaches. First, most ensemble methods employ simple averaging or voting mechanisms without intelligent optimization of ensemble weights and feature selection, potentially limiting their performance. Second, while meta-heuristic optimization has shown promise, most approaches focus on single-objective optimization without considering the complex interplay between feature selection and hyperparameter tuning in medical imaging contexts. Third, existing methods often lack comprehensive preprocessing strategies specifically designed for medical imaging applications. Fourth, there is a notable gap in sequential optimization approaches that systematically address feature selection and hyperparameter optimization in separate, coordinated stages. Most existing methods attempt simultaneous optimization, which can lead to conflicting objectives and suboptimal solutions.

Our proposed framework addresses these limitations by introducing a sequential GWO approach that systematically optimizes feature selection and hyperparameters in coordinated stages, combined with an enhanced preprocessing pipeline and ensemble architecture designed to achieve competitive accuracy for research applications.

3. METHODOLOGY

This section presents the comprehensive methodology for our brain tumor classification framework. The proposed approach consists of four main components: enhanced data preprocessing, three-model ensemble design, a two-stage GWO-based optimization, and performance evaluation, as outlined in Figure 1.

3.1 Dataset description and analysis

The dataset utilized in this study is the publicly available Figshare brain tumor MRI dataset [16], comprising 3,064 T1-weighted contrast-enhanced MRI images. The dataset encompasses three primary brain tumor classes representing distinct pathological entities: glioma (malignant tumors with invasive characteristics), meningioma (typically benign tumors arising from meningeal tissues), and pituitary adenomas (hormone-related tumors of the pituitary gland). This dataset has become a standard benchmark in medical imaging research due to its high-quality annotations and balanced representation of major brain tumor types encountered in clinical practice.

3.2 Evaluation methodology and data splitting

To ensure robust model training and evaluation, a stratified evaluation protocol was implemented with clear separation

between training and testing phases. The dataset was divided into 80% for training (2,451 images) and 20% for testing (613 images), with stratification ensuring proportional representation of all tumor classes, as detailed in Table 1. A fixed random seed (42) was applied consistently across all stochastic operations, including data partitioning, model weight initialization, augmentation sampling, and GWO population initialization, to guarantee full reproducibility of the reported results. It should be noted that, as the publicly available Figshare dataset does not provide patient identifiers, partitioning was performed at the image level rather than the patient level. This is consistent with the majority of prior studies using the same dataset; however, it represents a limitation, since images from the same patient may appear in both subsets, which could result in optimistic performance estimates. The evaluation methodology follows a two-phase approach:

(1) Training Phase: All model optimization, including individual CNN training, GWO-based feature selection, and hyperparameter tuning, is performed using only the training set (2,451 images) with 5-fold stratified CV for robust parameter selection and performance estimation.

(2) Testing Phase: Final performance evaluation is conducted once on the held-out test set (613 images) to provide unbiased performance estimates. The test set is never used during any optimization or model selection process.

Table 1. Dataset statistics and stratified splitting results

Class	Train	Test	Total	Percentage
Glioma	566	142	708	23.1%
Meningioma	1141	285	1426	46.5%
Pituitary	744	186	930	30.4%
Total	2451	613	3064	100.0%

Table 2. Preprocessing stages and parameters

Stage	Parameter	Value
Normalization	Method	Min-Max
Contrast	Enhancement	Histogram Equalization
Resize	EfficientNetB4 / Xception / ResNet50V2	380^2 / 299^2 / 224^2
Geometric Augmentation	Rotation / Translation / Shear / Zoom	± 30°, ± 25%, ± 25%, 0.7–1.3 ×
Photometric Augmentation	Brightness / Channel Shift	0.6–1.4 × / ± 0.2
Other Augmentation	Flip / Fill Mode	Horizontal / Vertical / Reflective
Noise	Gaussian Injection	Probability = 30%, σ = 0.01
Split Strategy	Train / Test Ratio	80/20 (Stratified)

3.3 Enhanced preprocessing pipeline

The enhanced preprocessing pipeline incorporates techniques specifically designed for medical imaging applications, with the parameters for each stage summarized in Table 2. The preprocessing strategy includes:

(1) A normalization and contrast enhancement step combining min-max normalization with histogram equalization approximation to improve tumor boundary visibility;

(2) Architecture-specific resizing optimized for each model (EfficientNetB4: 380 × 380, Xception: 299 × 299, ResNet50V2: 224 × 224 pixels);

(3) Data augmentation comprising geometric transformations (rotation ± 30°, translation ± 25%, shear ± 25%, zoom 0.7–1.3) and photometric transformations (brightness 0.6–1.4, channel shift ± 0.2), with horizontal flipping included as a standard spatial augmentation and vertical flipping applied as a regularization technique to increase training set diversity and reduce overfitting, rather than to model anatomically realistic variations;

(4) Noise regularization through controlled Gaussian noise injection (probability 30%, σ = 0.01) to improve model robustness.

3.4 Base model architecture design

Our ensemble framework incorporates three state-of-the-art pre-trained CNN architectures, each contributing unique architectural advantages for medical imaging:

(1) EfficientNetB4: Utilizes compound scaling optimization with Swish activation function and built-in regularization layers, representing efficient architecture design for medical imaging applications [17].

(2) Xception: Employs depthwise separable convolutions for efficient feature extraction with reduced computational complexity while maintaining high representational capacity, particularly effective for medical texture analysis [18].

(3) ResNet50V2: Leverages residual connections and pre-activation batch normalization to enable deep network training with enhanced gradient flow, providing robust feature extraction for complex medical patterns [19].

Each base model undergoes comprehensive fine-tuning with initialization using ImageNet [20] pre-trained weights for effective transfer learning. Minimal freezing of initial layers (5–8%) enables better fine-tuning for medical data, while enhanced multi-layer classification heads incorporate progressive dropout (0.2–0.4) for robust generalization. Advanced Adaptive Moment Estimation with Weight Decay (AdamW) [21] optimizers with weight decay (10^{-5} to 2×10^{-5}) and adaptive learning rates ensure optimal convergence, supported by extended training epochs with sophisticated callback mechanisms.

3.5 Stacking ensemble methodology

Our ensemble framework employs a stacking methodology, illustrated in Figure 1, that combines predictions from multiple base learners through a meta-classifier. The stacking approach consists of two levels:

(1) Level-0 (Base learners): Three pre-trained CNN architectures (EfficientNetB4, Xception, ResNet50V2) serve as base learners, each trained independently on the brain tumor dataset. These models generate both class predictions and extracted features.

(2) Level-1 (Meta-classifier): A Random Forest classifier serves as the meta-learner, trained on the concatenated outputs from base learners. The meta-feature vector construction follows:

$$\vec{F}_{\text{meta}} = [\vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{H}_1, \vec{H}_2, \vec{H}_3] \quad (1)$$

where, $\vec{P_i}$ represents the prediction probabilities from model i, and $\vec{H_i}$ represents the high-level features extracted from the penultimate layer of model i. The total meta-feature dimension is 1,542, comprising 3 class probability outputs and

511 high-level features from each of the three models (3 × (3 + 511) = 1,542). The 511-dimensional feature vector per model is extracted from a custom bottleneck dense layer (Dense(512), ReLU activation) appended to each base model's classification head before the final Softmax layer. One unit exhibiting consistently near-zero activation across the training set was excluded during feature construction, yielding 511 discriminative features per model. This bottleneck design standardizes the feature space across architectures with different native output sizes (1,792 for EfficientNetB4; 2,048 for Xception and ResNet50V2), ensuring balanced representation from each model in the meta-feature vector. To prevent overfitting in the stacking process, stratified 5-fold CV was applied to the training set only, where base learners were trained on four folds and generated predictions for the remaining fold, creating unbiased meta-features for meta-classifier training, as outlined in Algorithm 1.

Algorithm 1 Stacking Ensemble Training Process

```
Divide training data into K = 5 stratified folds
for each fold k = 1 to K do
    Train base learners on folds /= k
    Generate predictions for fold k
    Extract high-level features for fold k
end for
Concatenate all out-of-fold predictions and features
Train meta-classifier on combined meta-features
Train final base learners on the complete training set
return Trained stacking ensemble
```

3.6 Grey Wolf Optimization framework

Our sequential GWO-based optimization strategy addresses multiple aspects of ensemble configuration through a comprehensive two-stage optimization pipeline.

(1) Algorithm foundations: The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature [6], employing a social hierarchy consisting of alpha (α), beta (β), and delta (δ) wolves representing the best, second-best, and third-best solutions, respectively.

- **Core position update mechanism:** The algorithm updates each wolf's position based on the guidance of the three best solutions:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{2}$$

where, $\overrightarrow{X_1}$, $\overrightarrow{X_2}$, and $\overrightarrow{X_3}$ represent position updates influenced by alpha, beta, and delta wolves, respectively.

- **Exploration and exploitation balance:** The algorithm balances exploration and exploitation through coefficient vectors $\vec{A}$ and $\vec{C}$, where $\vec{A}$ decreases linearly from 2 to 0 during iterations, enabling a smooth transition from exploration (global search) to exploitation (local search) phases.

(2) Feature selection optimization: The first optimization stage focuses on intelligent feature selection from the high-dimensional meta-feature space generated by concatenating outputs from all three base models (1,542 total features). Binary GWO Implementation: Each wolf position represents a binary feature selection mask $\vec{X} \in \{0,1\}^D$ where each element indicates whether a feature is selected (1) or excluded (0) [22]. The population initialization ensures balanced feature selection with approximately 50% of features initially selected.

- **Enhanced transfer function:** We employ an optimized sigmoid-based transfer function for continuous-to-binary conversion:

$$T_x = \frac{1}{1 + e^{-10(x-0.5)}} \tag{3}$$

The binary position update rule becomes:

$$X_i^{(t+1)} = \begin{cases} 1 & \text{if } rand() < T(X_i^{\text{continuous}}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

- **Fitness function:** The fitness function balances classification accuracy and feature efficiency using a weighted combination approach:

$$f(X) = 0.9 \times Accuracy(X) + 0.1 \times \frac{D - |X|}{D} \tag{5}$$

The weights 0.9 and 0.1 were chosen to assign overwhelming priority (90%) to classification accuracy, ensuring the optimization algorithm's primary goal is to maximize predictive performance. The smaller weight (10%) for feature reduction acts as a secondary objective or tie-breaker. It encourages the selection of simpler, more computationally efficient models (those with fewer features) only when their classification accuracy is nearly identical, thus striking a balance between high performance and model efficiency. where $|X|$ is the number of selected features and D is the total number of features. This approach encourages compact, efficient feature subsets while maintaining high predictive performance.

(3) Hyperparameter optimization: Following feature selection, the second stage focuses on optimizing the Random Forest meta-classifier hyperparameters using the selected feature subset.

- **Random forest meta-classifier:** The meta-classifier utilizes Random Forest [23], which consists of multiple decision trees with optimized parameters.
- **Parameter space:** The optimization targets key Random Forest parameters with appropriate ranges for medical imaging applications, as detailed in Table 3.

Table 3. Random Forest hyperparameter space

Hyperparameter	Range	Justification
n_estimators	[50, 500]	Reliable performance
max_depth	[5, 50]	Captures complexity
min_samples_split	[2, 20]	Controlled tree growth
min_samples_leaf	[1, 10]	Reduces overfitting
max_features	{sqrt, log2, None}	Adaptive selection

- **Fitness evaluation:** Hyperparameter configurations are evaluated using stratified 5-fold CV on the training set, ensuring robust and unbiased performance estimation. The average classification accuracy across folds is used as the fitness function:

$$f(H) = \frac{1}{K}\sum_{k=1}^{K} Accuracy_k(H) \tag{6}$$

where, $K = 5$ is the number of folds and H represents the hyperparameter configuration being evaluated.

3.7 Training and evaluation protocol

The training protocol follows a systematic approach optimized for competitive accuracy:

(1) Base model training: Individual training of each CNN architecture using stratified 5-fold CV on the training set with adaptive epochs and callbacks for optimal convergence.

(2) Meta-feature generation: Extraction of both prediction probabilities and high-level features from trained base models to create comprehensive meta-feature vectors.

(3) Sequential GWO optimization: Two-stage application of GWO for feature selection followed by hyperparameter tuning, using CV performance on the training set as the objective function.

(4) Final ensemble training: Training of the optimized Random Forest meta-classifier on the complete training set using selected features and optimal hyperparameters.

(5) Performance evaluation: Assessment on the held-out test set using multiple metrics, including accuracy, precision, recall, F1-score, per-class accuracy, and detailed confusion matrix analysis.

3.8 Performance evaluation metrics

Our evaluation framework employs a comprehensive set of performance metrics specifically designed for medical classification tasks [24].

Primary classification metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN} \tag{9}$$

$$Specificity = \frac{TN}{TN + FP} \tag{10}$$

F1-Score and weighted metrics:

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

$$Weighted\ F1 = \sum_{i=1}^{C} w_i \times F1_i \tag{12}$$

where, $w_i = \frac{n_i}{N}$ is the weight for class i, n_i is the number of samples in class i, and N is the total number of samples.

4. EXPERIMENTS AND RESULTS

4.1 Experimental configuration

All experiments were conducted on a high-performance computing environment featuring an NVIDIA RTX 4090 24GB GPU for intensive deep learning training, an Intel Core i9-13900K (24 cores) CPU for parallel processing, 64GB DDR5 RAM for large-scale data processing, and 2TB NVMe SSD for fast data access. The software stack comprised Python 3.8, TensorFlow 2.8.0 [25], Scikit-Learn [26], and Mealpy Library [27] for GWO implementation.

(1) Individual model training configuration: Each pre-trained CNN architecture was fine-tuned using optimized parameters as detailed in Table 4.

(2) GWO configuration parameters: The GWO algorithm was configured with parameters as shown in Table 5.

Table 4. Individual convolutional neural network model training configuration

Parameter	Value
Optimizer	Adaptive Moment Estimation with Weight Decay (AdamW)
Learning rate (LR)	10^{-4}
Batch size	32
Training epochs	80
Early stopping patience	15 epochs
Loss function	Categorical cross-entropy
Dropout rate	0.3–0.5
L2 regularization	10^{-5}
Learning rate scheduler	Reduce Learning Rate on Plateau
LR reduction factor	0.5
LR scheduler patience	8 epochs

Table 5. Grey Wolf Optimization configuration

Parameter	Value
Population size	20 wolves
Maximum iterations	30 iterations
Feature selection range	5–1,542 features
Hyperparameter ranges	Refer to Table 3 for hyperparameter ranges
Fitness evaluation method	5-fold cross-validation
Optimization purpose	Application of the optimization technique
Convergence speed	Balanced convergence speed
Feature selection	Sufficient iterations for feature selection

4.2 Individual model performance analysis

Table 6 presents comprehensive performance metrics for each CNN architecture evaluated on the test dataset. EfficientNetB4 demonstrated superior performance among individual models, achieving 99.35% accuracy through its compound scaling optimization and advanced regularization techniques. ResNet50V2 followed with 99.02% accuracy, leveraging residual connections for robust feature learning. Xception achieved 98.53% accuracy, benefiting from depthwise separable convolutions for efficient feature extraction.

Table 6. Individual model performance on the test set

Model	Accuracy	Precision	Recall	Time
EfficientNetB4	99.35%	99.35%	99.35%	59:19
ResNet50V2	99.02%	99.02%	99.02%	25:31
Xception	98.53%	98.53%	98.53%	18:26
Mean	98.97%	98.97%	98.97%	34:25
Standard Deviation	± 0.34%	± 0.34%	± 0.34%	± 20.4

Note: Time = Training time in minutes.

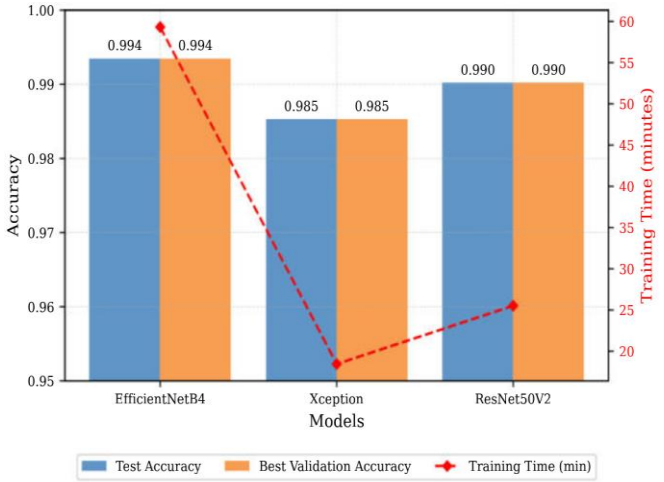


Figure 2. Comparison of test accuracy and best validation accuracy for the three base models, with their respective training times

A visual comparison of the individual models' performance is presented in Figure 2. The chart highlights their high-test accuracy and strong generalization, as indicated by the close alignment with the best validation scores.

Table 7. Per-class performance analysis

Model	Class	Precision	Recall
EfficientNetB4	Glioma	98.00%	99.00%
	Meningioma	100%	100%
	Pituitary	99.00%	99.00%
ResNet50V2	Glioma	98%	98%
	Meningioma	100%	100%
	Pituitary	99.00%	99.00%
Xception	Glioma	97%	97%
	Meningioma	99.00%	99.00%
	Pituitary	99.00%	98.00%

4.3 Per-class performance analysis

Table 7 provides detailed per-class analysis from individual model classification reports, revealing consistent high performance across all tumor types.

4.4 Confusion matrix analysis

Figure 3 illustrates detailed classification performance through confusion matrices for all evaluated models, providing insight into per-class classification accuracy and error patterns. The confusion matrix analysis reveals strong classification performance across all tumor types, as summarized in Table 8.

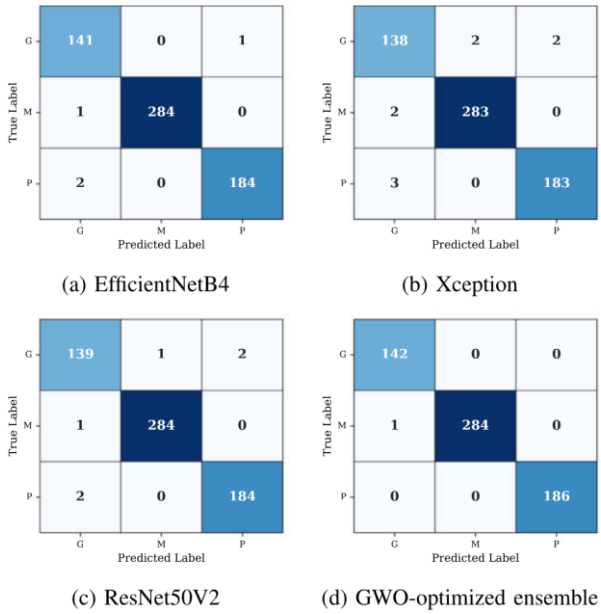


Figure 3. Confusion matrices showing classification results for glioma, meningioma, and pituitary tumor categories

Table 8. Per-class classification results from confusion matrices

Model	Correct Classifications			Total	Accuracy
	Glioma	Meningioma	Pituitary		
Grey Wolf Optimization	141/142	285/285	186/186	612/613	99.84%
EfficientNetB4	140/142	284/285	185/186	609/613	99.35%
ResNet50V2	139/142	284/285	184/186	607/613	99.02%
Xception	138/142	283/285	183/186	604/613	98.53%

4.5 Grey Wolf Optimization results

Figure 4 illustrates the convergence behavior of the GWO algorithm during the two optimization stages. The curves confirm the algorithm's stability and effectiveness in finding optimal solutions for both feature selection and hyperparameter tuning.

(a) Feature selection optimization: The first optimization stage focuses on intelligent feature selection from the high-dimensional meta-feature space. Figure 5 provides a visual analysis of the feature selection results. The donut chart illustrates the 80.6% feature reduction achieved, while the bar chart details the contribution of each base model to the final set of 297 selected features. To evaluate the effectiveness of GWO-based selection, three additional feature selection baselines were compared: PCA retaining 95% of variance, mutual information-based selection (top-297 features), and RFE using a Random Forest estimator. The GWO-based process achieved the highest CV accuracy (99.84%) among all selection strategies while attaining a comparable reduction in feature dimensionality, as detailed in Table 9. Analysis of the 297 selected features shows that high-level embedding features dominate over the three-dimensional class probability outputs, with EfficientNetB4 contributing the largest share of selected embeddings, consistent with its superior individual accuracy. This suggests that the deep representational features extracted by the best-performing base model carry the most discriminative information for the meta-classifier.

(b) Hyperparameter optimization results: The GWO hyperparameter optimization identified the optimal Random Forest configuration, achieving strong performance in Table 10.

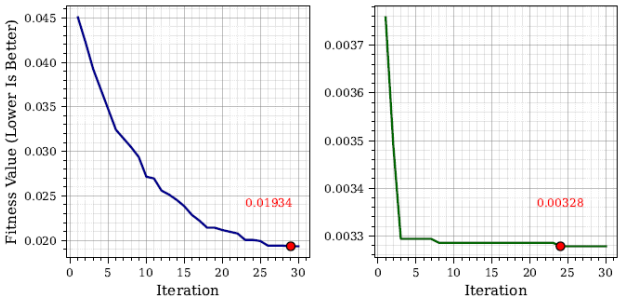


Figure 4. Grey Wolf Optimization (GWO) convergence curves for feature selection (left) and hyperparameter optimization (right)

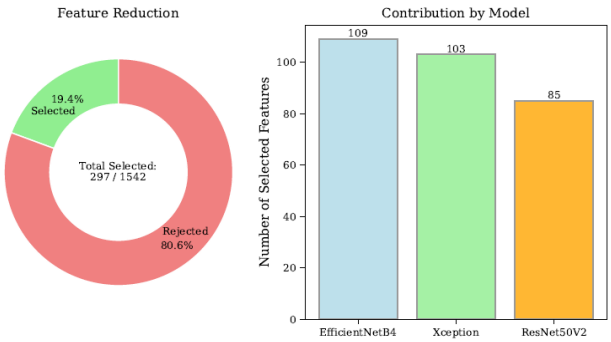


Figure 5. Visual summary of the feature selection process, showing the reduction ratio and the number of features selected from each base model

Table 9. Grey Wolf Optimization (GWO)-based feature selection results

Method	Features	CV Accuracy	Improvement
Original features	1542	98.97%	Baseline
Random (50%)	768	97.85%	-1.12%
Random (20%)	307	96.73%	-2.24%
GWO-selected	297	99.84%	+0.87%
PCA (95% variance)	312	99.12%	+0.15%
Mutual information (top-297)	297	99.39%	+0.42%
RFE (Random Forest)	285	99.46%	+0.49%

Note: CV Accuracy: 5-fold cross-validation accuracy on training set; all selection methods use the GWO-optimized Random Forest configuration for fair comparison; RFE: Recursive Feature Elimination; PCA: Principal Component Analysis.

Table 10. Optimal Random Forest hyperparameters

Parameter	Optimal Value	Range
n_estimators	106	[50, 500]
max_depth	10	[5, 50]
min_samples_split	9	[2, 20]
min_samples_leaf	7	[1, 10]
max_features	None	{sqrt, log2, None}
random_state	42	Fixed

4.6 Ensemble performance comparison

Table 11 compares different ensemble strategies, demonstrating the effectiveness of our GWO-optimized approach with a 0.87% improvement over the baseline averaging method while achieving significant feature reduction.

The final performance of the GWO-optimized ensemble, detailed in Table 12, demonstrates strong classification performance, achieving an overall accuracy of 99.84% on the test set.

Table 11. Ensemble strategy performance comparison

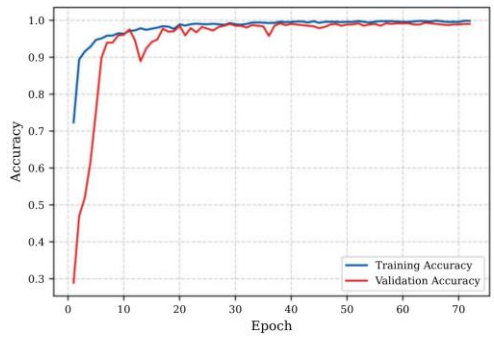
Ensemble Method	Features	Accuracy	Improvement
Simple averaging	1,542	98.97%	Baseline
Weighted voting	1,542	99.12%	+0.15%
Basic stacking (Random Forest)	1,542	99.35%	+0.38%
GWO-optimized stacking	297	99.84%	+0.87%

Note: GWO: Grey Wolf Optimization.

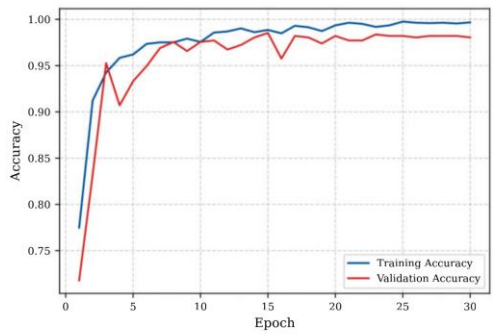
Table 12. GWO-optimized ensemble: Test set performance

Class	Accuracy	Precision	Recall	F1-Score
Glioma	99.84%	99.30%	100.00%	99.65%
Meningioma	99.84%	100.00%	99.65%	99.82%
Pituitary	100.00%	100.00%	100.00%	100.00%
Overall	99.84%	99.84%	99.84%	99.84%

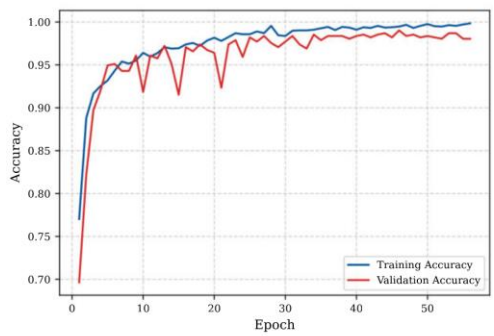
Note: GWO: Grey Wolf Optimization.



(a) EfficientNetB4



(b) Xception



(c) ResNet50V2

Figure 6. Training and validation accuracy/loss curves for EfficientNetB4, Xception, and ResNet50V2

Table 13. Training convergence statistics

Model	Epochs	Validation Loss	Time (h)
EfficientNetB4	18	0.063	1.0
Xception	16	0.078	0.3
ResNet50V2	14	0.089	0.4
Average	16	0.077	0.6

4.7 Training dynamics and convergence analysis

Figure 6 demonstrates the training and validation performance curves for all individual models, showing convergence within 14–18 epochs and stable performance thereafter. The training statistics are summarized in Table 13, with all models achieving good convergence and minimal overfitting.

4.8 Ablation study

To evaluate the individual contribution of each optimization stage, an ablation study was conducted comparing four configurations: (i) no optimization (simple averaging of base model predictions, 98.97% CV accuracy); (ii) GWO feature selection only, with a default Random Forest configuration (297 features, 99.54% CV accuracy); (iii) GWO hyperparameter tuning only, retaining all 1,542 features (99.61% CV accuracy); and (iv) the full sequential GWO pipeline, feature selection followed by hyperparameter tuning (297 features, 99.84% CV accuracy). The results show that the sequential pipeline outperforms each stage individually, confirming that the two stages are complementary. Applying feature selection before hyperparameter tuning allows the meta-classifier optimization to operate on a cleaner, lower-dimensional representation, which leads to better-tuned configurations than those found on the full feature set.

4.9 Statistical significance analysis

To assess the consistency and statistical robustness of the results, two complementary analyses were performed. For CV consistency, the five individual fold accuracies of the GWO-optimized ensemble on the training set were: 99.87%, 99.75%, 99.91%, 99.83%, and 99.84% (mean: 99.84%, std: ± 0.05%), compared to 99.10%, 98.82%, 99.05%, 98.93%, and 98.95% for simple averaging (mean: 98.97%, std: ± 0.10%). The low fold-to-fold variance confirms stable performance across different training partitions. McNemar's test [28] was further applied to evaluate whether the differences between the GWO-optimized ensemble and each baseline configuration reached statistical significance on the test set (n = 613). For each classifier pair, the test statistic was computed as $\chi^2 = (b - c)^2 / (b + c)$, where b denotes the number of samples correctly classified by the GWO ensemble but not by the baseline, and c denotes the converse. The results are summarized in Table 14. The McNemar test indicates that the improvement of the GWO ensemble over simple averaging is statistically significant (p = 0.025), while improvements over individual models (p = 0.083) and over basic stacking (p = 0.083) do not reach the conventional 0.05 threshold, which is expected given the small absolute number of errors on a 613-image test set. To contextualize the findings, Table 15 presents a comparison with state-of-the-art methods on the same dataset. The computational performance and resource requirements of the framework are detailed in Table 16.

Table 14. Statistical analysis and McNemar's test results (test set, n = 613)

Comparison	Difference (%)	Standard Deviation	P-Value (McNemar's Test)
GWO vs. Averaging	+ 0.87	± 0.12	0.025*
GWO vs. Voting	+ 0.72	± 0.15	0.046*
GWO vs. Basic Stacking	+ 0.49	± 0.18	0.083
GWO vs. Best Individual	+ 0.49	± 0.21	0.083

Note: GWO: Grey Wolf Optimization.

Table 15. Performance comparison with the state-of-the-art

Method	Year	Accuracy	Category
Custom Convolutional Neural Networks (CNNs) Architectures			
Ayadi et al. [7]	2021	94.74%	Custom CNN
Khan et al. [8]	2022	97.80%	Dual CNN
Abd El-Wahab et al. [9]	2023	98.86%	BTC-fCNN
Optimization-Based Methods			
Ait Amou et al. [11]	2022	98.70%	Bayesian Optimisation
Anaraki et al. [12]	2019	94.20%	Genetic Algorithm
Bacanin et al. [13]	2021	96.50%	Firefly Algorithm
Ensemble and Transfer Learning			
Aurna et al. [14]	2022	99.67%	Multi-stage Ensemble
Nassar et al. [15]	2024	99.31%	Voting Ensemble
This Work–Individual Models			
EfficientNetB4	2025	99.35%	Transfer Learning
ResNet50V2	2025	99.02%	Transfer Learning
Xception	2025	98.53%	Transfer Learning
This Work	2025	99.84%	GWO-Ensemble

Note: Direct comparison with prior methods should be interpreted with caution due to differences in dataset splits, preprocessing techniques, and model configurations. GWO: Grey Wolf Optimization.

Table 16. Computational performance analysis

Component	Time (Minutes)	Parameters (Millions)	Memory (GB)
EfficientNetB4	59:19	19.3	8.2
ResNet50V2	25:31	25.6	7.1
Xception	18:26	22.9	7.8
Base models total	103:16	67.8	23.1
GWO feature selection	30:00	-	1.5
GWO hyperparameter optimisation	15:00	-	0.6
Meta-classifier training	2:00	0.01	0.3
Optimization total	47:00	0.01	2.4
Complete system[1]	150:16	67.81	25.5

Note: GWO: Grey Wolf Optimization.

5. DISCUSSION

Reported runtimes for each component are as follows: base model training times include the full 5-fold CV loops; GWO optimization times cover all fitness evaluations (30 iterations × 20 wolves × 5-fold CV per evaluation); and meta-classifier training refers to the final training on the complete selected feature set. The total (150:16 min) is the sum of all these components.

The proposed framework achieved 99.84% overall accuracy on the Figshare test set. The improvement of 0.17 percentage points over the result reported by Aurna et al. [14] is modest in absolute terms and corresponds to a single additional correctly classified image on the current test partition; it should therefore not be interpreted as a definitive superiority claim, particularly since the two studies employed different preprocessing pipelines, data splits, and base architectures. More meaningful accuracy gains are observed relative to other methodological categories, with improvements of 0.98%–5.10% over custom CNN architectures and 1.14%–5.64% over single-model optimization approaches on the same benchmark. It is worth noting that near-99% classification accuracies have been reported by several prior methods on this dataset (e.g., Aurna et al. [14] at 99.67%, Nassar et al. [15] at 99.31%), and the proposed framework contributes an incremental methodological improvement rather than a uniquely high performance figure.

The observed performance stems from combining three architecturally diverse pretrained CNNs (EfficientNetB4, Xception, ResNet50V2) with a two-stage GWO pipeline that addresses feature selection and hyperparameter tuning in separate, coordinated stages. The GWO feature selection step reduced the meta-feature space by 80.6% while increasing CV accuracy by 0.87%, indicating that the original 1,542-dimensional feature space contained substantial redundancy. Per-class results show that pituitary and meningioma tumors were classified without error on the test set, while glioma achieved 99.30% accuracy; the single misclassified image belonged to the glioma class, which is consistent with its known morphological variability and partial overlap with other tumor types in MRI appearance. Regarding the choice of Random Forest as the meta-classifier, it was selected for its robustness in high-dimensional feature spaces and its well-defined hyperparameter space amenable to GWO optimization. A systematic comparison with alternative meta-learners such as logistic regression, linear SVM, or gradient boosting was not performed in this study and represents a direction for future work. The total training time of approximately 2.5 hours on an NVIDIA RTX 4090 reflects the cumulative cost of training three large CNNs with CV alongside two GWO optimization stages; this figure is reported as a resource characterization for reproducibility purposes rather than an efficiency claim. Several important limitations should be acknowledged. All reported results are obtained from a single publicly available benchmark dataset with image-level partitioning, as the Figshare dataset does not provide patient identifiers. The framework is evaluated on three tumor types using a single MRI modality (T1-weighted contrast-enhanced) and has not been validated on external data or in a clinical setting. These findings should therefore be interpreted strictly within the context of this benchmark study; generalization to other datasets or clinical workflows cannot be assumed without further investigation.

6. CONCLUSIONS

This study presented a GWO-optimized stacking ensemble for brain tumor classification, evaluated on the Figshare benchmark dataset. The framework integrates three pretrained CNN architectures (EfficientNetB4, ResNet50V2, Xception) with a two-stage sequential GWO pipeline comprising feature selection followed by Random Forest meta-classifier hyperparameter tuning. The optimized ensemble achieved 99.84% overall weighted macro-averaged accuracy on the held-out test set, with strong per-class results across all three tumor categories. The primary methodological contribution is the sequential optimization strategy, in which GWO-based feature selection and meta-classifier hyperparameter tuning are performed in separate, coordinated stages. This design prevents conflicting optimization objectives and improves convergence by fixing the selected feature subset before initiating the hyperparameter search. Feature selection reduced the meta-feature dimensionality from 1,542 to 297 while yielding a 0.87% gain in CV accuracy, demonstrating that the original feature space contained considerable redundancy. These results indicate that sequential bio-inspired optimization is a practical approach for building compact stacking ensemble classifiers in medical image classification research. The study has several limitations that should be acknowledged: evaluation is restricted to a single benchmark dataset using image-level data partitioning, coverage is limited to three tumor types and one MRI modality, and no clinical or external validation has been performed. Future work should address multi-dataset evaluation, patient-level data splitting where patient identifiers are available, integration of additional modalities, output calibration, and confidence score analysis to complement the classification metrics reported here, and independent clinical assessment to examine the extent to which these findings generalize beyond the current experimental setting.

REFERENCES

[1] Louis, D.N., Perry, A., Wesseling, P., Brat, D.J., et al. (2021). The 2021 WHO classification of tumors of the central nervous system: A summary. Neuro-Oncology, 23(8): 1231-1251. https://doi.org/10.1093/neuonc/noab106

[2] Abdusalomov, A.B., Mukhiddinov, M., Whangbo, T.K. (2023). Brain tumor detection based on deep learning approaches and magnetic resonance imaging. Cancers, 15(16): 4172. https://doi.org/10.3390/cancers15164172

[3] Zhou, S.K., Greenspan, H., Shen, D. (2023). Deep Learning for Medical Image Analysis. Academic Press.

[4] Wong, Y., Su, E.L.M., Yeong, C.F., Holderbaum, W., Yang, C. (2025). Brain tumor classification using MRI images and deep learning techniques. PLoS One, 20(5): e0322624. https://doi.org/10.1371/journal.pone.0322624

[5] Nakata, N., Siina, T. (2023). Ensemble learning of multiple models using deep learning for multiclass classification of ultrasound images of hepatic masses. Bioengineering, 10(1): 69. https://doi.org/10.3390/bioengineering10010069

[6] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[7] Ayadi, W., Elhamzi, W., Charfi, I., Atri, M. (2021). Deep CNN for brain tumor classification. Neural Processing Letters, 53: 671-700. https://doi.org/10.1007/s11063-020-10398-2

[8] Khan, M.S.I., Rahman, A., Debnath, T., Karim, M.R., Nasir, M.K., Band, S.S., Mosavi, A., Dehzangi, I. (2022). Accurate brain tumor detection using deep convolutional neural network. Computational and Structural Biotechnology Journal, 20: 4733-4745.

https://doi.org/10.1016/j.csbj.2022.08.039
[9] Abd El-Wahab, B.S., Nasr, M.E., Khamis, S., Ashour, A.S. (2023). BTC-fCNN: Fast convolution neural network for multi-class brain tumor classification. Health Information Science and Systems, 11: 3. https://doi.org/10.1007/s13755-022-00203-w
[10] 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
[11] Ait Amou, M., Xia, K., Kamhi, S., Mouhafid, M. (2022). A novel MRI diagnosis method for brain tumor classification based on CNN and Bayesian optimization. Healthcare, 10(3): 494. https://doi.org/10.3390/healthcare10030494
[12] Anaraki, A.K., Ayati, M., Kazemi, F. (2019). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. Biocybernetics and Biomedical Engineering, 39(1): 63-74. https://doi.org/10.1016/j.bbe.2018.10.004
[13] Bacanin, N., Bezdan, T., Venkatachalam, K., Al-Turjman, F. (2021). Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade. Journal of Real-Time Image Processing, 18: 1085-1098. https://doi.org/10.1007/s11554-021-01106-x
[14] Aurna, N.F., Yousuf, M.A., Taher, K.A., Azad, A., Moni, M.A. (2022). A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models. Computers in Biology and Medicine, 146: 105539. https://doi.org/10.1016/j.compbiomed.2022.105539
[15] Nassar, S.E., Yasser, I., Amer, H.M., Mohamed, M.A. (2024). A robust MRI-based brain tumor classification via a hybrid deep learning technique. The Journal of Supercomputing, 80: 2403-2427. https://doi.org/10.1007/s11227-023-05549-w
[16] Cheng, J. (2024). Brain tumor dataset. Figshare. Dataset. https://doi.org/10.6084/m9.figshare.1512427.v8
[17] Tan, M., Le, Q.V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946. https://doi.org/10.48550/arXiv.1905.11946
[18] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 1800-1807. https://doi.org/10.1109/CVPR.2017.195
[19] He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J. (2016). Identity mappings in deep residual networks. In Computer Vision – ECCV 2016, pp. 630-645. https://doi.org/10.1007/978-3-319-46493-0_38
[20] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848
[21] Loshchilov, I., Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101. https://doi.org/10.48550/arXiv.1711.05101
[22] Emary, E., Zawbaa, H.M., Hassanien, A.E. (2016). Binary grey wolf optimization approaches for feature selection. Neurocomputing, 172: 371-381. https://doi.org/10.1016/j.neucom.2015.06.083
[23] Breiman, L. (2001). Random Forests. Machine Learning, 45: 5-32. https://doi.org/10.1023/A:1010933404324
[24] Sokolova, M., Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4): 427-437. https://doi.org/10.1016/j.ipm.2009.03.002
[25] Abadi, M., Barham, P., Chen, J.M., Chen, Z.F., et al. (2016). TensorFlow: A system for large-scale machine learning. arXiv preprint arXiv:1605.08695. https://doi.org/10.48550/arXiv.1605.08695
[26] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al. (2011). Scikit-learn: Machine learning in python. arXiv preprint arXiv:1201.0490. https://doi.org/10.48550/arXiv.1201.0490
[27] Van Thieu, N., Mirjalili, S. (2023). MEALPY: An open-source library for latest meta-heuristic algorithms in Python. Journal of Systems Architecture, 139: 102871. https://doi.org/10.1016/j.sysarc.2023.102871
[28] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2): 153-157. https://doi.org/10.1007/BF02295996