Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

ASO-Optimized EfficientNet-B2 with CBAM for Brain Tumor MRI Classification

Ramisetty Balaji[1], Anantha Natarajan V.[2]

Department of Computer Science and Engineering, School of Computing, Mohan Babu University, Tirupati, Andhra Pradesh, India.

E-mail: [1]ramisettybalaji.phd@gmail.com, [2]v.ananth.satyam@gmail.com

Orcid ID: [1]0009-0007-6614-1120, [2]0000-0002-1577-0708

Abstract

Magnetic resonance imaging (MRI) plays a critical role in brain tumor diagnosis; however, accurate classification remains a challenging task in medical image analysis. Gradient-based optimizers form the foundation of deep learning models; however, they are susceptible to suboptimal local minima and hyperparameter sensitivity. This paper proposes a hybrid model in which the CBAM attention module, the classifier head, and the final two convolutional blocks of EfficientNet-B2 are optimized using Atom Search Optimization (ASO), while the remaining backbone layers are retained as fixed feature extractors. ASO generates a population of candidate solutions by simulating atomic motion using Lennard-Jones and bond-length potentials, without relying on gradient information, and is used to search the high-dimensional parameter space. Experiments on the Figshare brain tumor dataset (3,064 T1-weighted MRI images of glioma, meningioma, and pituitary tumors) demonstrate that the ASO-optimized EfficientNet-B2+CBAM model has an accuracy of 99.35% and F1-score of 99.38%, outperforming Adam (98.76%) and SGD (98.12%). Statistically significant improvements ($p < 0.01$) are observed, and convergence analysis indicates ASO is less susceptible to plateaus of loss, and the variance is reduced by 44.7% across runs. The effectiveness of ASO is further evaluated through ablation studies, ROC analysis, and Grad-CAM visualizations, demonstrating its potential as an alternative optimizer for brain tumor MRI classification. Further validation on multi-center clinical datasets is required to assess generalizability and clinical applicability.

Keywords: Brain Tumor Classification; Atom Search Optimization; Metaheuristic Optimization; EfficientNet; Convolutional Block Attention Module; Population-Based Training.

1. Introduction

Brain tumors are among the most severe diseases of the central nervous system, characterized by high mortality and recurrence rates, thus requiring accurate and timely diagnosis for effective treatment. According to the Central Brain Tumor Registry of the United States, approximately 29.7% of brain and central nervous system tumors are malignant, while the remaining 70.3% are non-malignant [1]. MRI is widely regarded as the gold standard for

DOI: https://doi.org/10.36548/jiip.2026.3.006
Received: 29.04.2026, received in revised form: 02.06.2026, accepted: 23.06.2026, published: 06.07.2026

detecting and classifying brain tumors due to its superior soft-tissue contrast and ability to visualize heterogeneous tumors of varying sizes, shapes, and intensity properties. However, manual radiological analysis is time-consuming, subject to inter-observer variability, and increasingly impractical given the rapid increase in medical imaging data [2].

Recent developments in deep learning have significantly improved the automatic processing of brain tumors, with convolutional neural networks (CNNs) demonstrating exceptional performance in classification and segmentation tasks [3]. Some research has reported classification accuracies exceeding 99% on benchmark datasets, and networks such as EfficientNet and attention-based models have achieved state-of-the-art performance [4]. However, these improvements are largely dependent on the training algorithm used [5]. The majority of current approaches rely on gradient-based optimizers (e.g., Adam and stochastic gradient descent (SGD)), which are computationally efficient and widely adopted. Nevertheless, they are hyperparameter-sensitive, necessitate differentiable objective functions, and are local search techniques that can converge to suboptimal local minima [6]. These limitations are further exacerbated by the highly non-convex optimization surfaces exhibited by deep neural networks, which tend to impair generalization performance [7].

To address these issues, bio-inspired and physics-inspired metaheuristic optimization algorithms have emerged as promising alternatives. Among these, physics-based approaches rooted in atomic and molecular dynamics, known as atomic optimization, offer gradient-free global search, a high exploration/exploitation ratio, and solutions to complex optimization problems [8]. In this paper, we propose Atom Search Optimization (ASO) as an alternative to traditional gradient-based optimizers for training a deep neural network, specifically combining EfficientNet-B2 with the Convolutional Block Attention Module (CBAM). The main contributions are as follows:

- New Optimization Framework:A population-based ASO training strategy that optimizes the CBAM module, classification head, and the final two EfficientNet-B2 blocks as an alternative to conventional gradient-based optimization.
- Architectural Enhancement: The proposed architecture (Integration of EfficientNet-B2 and CBAM) for effective feature extraction and refinement.
- Comprehensive Evaluation: A detailed comparison with Adam, SGD, and RMSprop using the Figshare dataset, including convergence, accuracy, statistical, and ablation analysis.

Existing studies have reported only limited investigations of ASO as a primary training optimizer for brain tumor MRI classification, which provides a novel perspective on metaheuristic-based medical deep learning. The remainder of this paper presents related work, methodology, experimental setup, results, and conclusions.

2. Related Work

2.1 Deep Learning for Brain Tumor Classification

CNN-based models now deliver high accuracy in brain tumor MRI classification. Aamir et al. [9] used guided filter with deep learning to obtain 99.67% on Figshare. According to Khushi et al. [10], EfficientNetB0 with Adam obtained 99.13%. EfficientNetV2 networks [11]

and hybrid attention structures further improved performance; the CBAM-SMK architecture [12] achieved 97.0% using 7,023 images. It is reported that many hybrid models outperform 98% [13]. However, almost all research uses only classical optimization algorithms such as Adam or SGD, while optimization algorithms themselves are not sufficiently researched [14].

2.2 Attention Mechanisms in Medical Imaging

Attention-based mechanisms improve CNNs by identifying clinically important features while removing irrelevant background information. Attention is used in CBAM [15] as channel and spatial attention to enhance feature representation. It has been used together with ResNet50 [16], MobileNet [17], and even custom architectures, where it improved accuracy and localization. Works such as CBAM-SMK have shown less background noise and more focused tumor detection, and even larger studies [3] have proven that attention helps models learn clinically important brain regions.

2.3 Deep Learning and Metaheuristic Optimization

Metaheuristic optimization techniques can serve as a valuable alternative to enhance the performance of deep learning in medical imaging. A meta-analysis of 106 studies suggested that bio-inspired optimization methods can enhance the robustness and segmentation of multimodal MRI applications [8]. Algorithms such as Harris Hawks Optimization, Salp Swarm Algorithm, Falcon Finch Optimization, etc., have been successfully incorporated with CNN-based frameworks [18]. Most previous research applies metaheuristics for hyperparameter tuning, not to train the neural network itself. A recent study indicates that optimizing a neural network in a population-based way can be done without suffering from local minima [19].

2.4 Atom Search Optimization

Atom Search Optimization (ASO), a physics-inspired algorithm proposed by Zhao et al. (2013) and later modified in 2014 [20], simulates atomic interactions via Lennard–Jones potential and bond-length constraints to balance global exploration with local exploitation. It has performed well on benchmark optimization and engineering problems [8]. Recent work extends ASO through hybrid and reinforcement learning-based behavioral modeling to predict action sequences [21]. However, ASO remains largely unexplored as a main training optimizer for deep neural networks in medical image classification, which motivates the proposed approach.

3. Proposed Methodology

The mathematical modeling and algorithmic procedure of a brain tumor classification framework are introduced in this section. The methodology has five main parts:Dataset Preparation and Preprocessing, Feature Extraction from EfficientNet-B2, Attention Refinement from CBAM, Design of Classification Head, and Optimization with ASO. The overall architecture of the proposed framework is depicted in Figure 1.

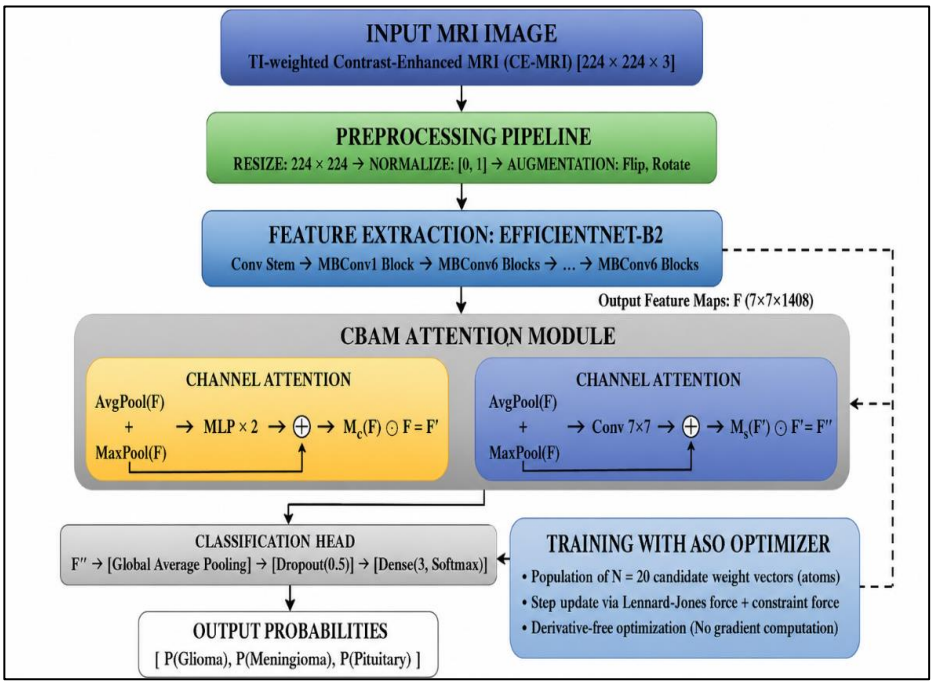

Figure 1. Complete architecture of the proposed ASO-optimized framework

3.1 Problem Statement

Given a dataset $D = \{(X_i, y_i)\}_{i=1}^{N}$ of N T1-weighted contrast-enhanced MRI slices, where $X_i \in \mathbb{R}^{H\times W\times C}$ and $y_i \in \{1,2,3\}$ labels the tumor as Glioma, Meningioma, or Pituitary, the goal is to find network parameters $\theta^* \in \mathbb{R}^D$ that minimize the empirical loss $L(\theta)$ on D. Conventional training uses gradient updates:

$$\theta_{t+1} = \theta_t - \eta_t \cdot m(\nabla_\theta L_t) \tag{1}$$

but non-convexity of $L(\theta)$ can trap these local methods in poor minima and make them sensitive to initialization.

To address this, we reframe training as a population-based global optimization. A set of M candidate parameter vectors $P = \{\theta_1, \dots, \theta_M\}$ is evolved over T iterations using the ASO algorithm—a physics-inspired metaheuristic that needs no gradient information. Each atom updates via interaction forces from the Lennard-Jones potential and constraint forces pulling toward the current best solution. Formally, we solve:

$$\theta^* = \arg\min_{\theta\in P(T)} L(\theta) \tag{2}$$

subject to the atomic motion dynamics in Section 3.5. The framework aims to achieve (i) lower validation loss, (ii) higher classification accuracy, and (iii) reduced run-to-run variance than gradient-based optimizers.

3.2 Dataset Description and Preprocessing

3.2.1 Figshare Brain Tumor Dataset

Cheng's Figshare dataset [22] provides 3,064 T1-weighted contrast-enhanced MRI slices from 233 patients, spanning three tumor classes: Glioma (1,426 images, arising from

glial cells), Meningioma (tumors of the brain's protective membranes), and Pituitary tumor (930 images, originating from the pituitary gland). The images cover coronal, sagittal, and axial views. This dataset is a widely used public benchmark, enabling direct comparison with prior work; however, single-dataset evaluation may not capture the variability of real-world clinical imaging.

3.2.2 Data Preprocessing Pipeline

The MRI images were preprocessed to ensure consistent input representation and improve model generalization. The preprocessing pipeline consisted of image resizing, normalization, patient-wise dataset splitting, and data augmentation.

Step 1: Image Resizing

All MRI images were resized to 224 × 224 pixels using bilinear interpolation while preserving the aspect ratio. This resolution was selected as a practical trade-off between computational efficiency and classification performance, since higher resolutions provided only marginal performance gains with increased training cost. The resized image is computed as

$$I_{\text{resized}}(x, y) = \sum_{i=0}^{1} \sum_{j=0}^{1} w_{ij}\, I_{\text{src}}(x_i, y_j) \tag{3}$$

where w_{ij}denotes the bilinear interpolation weights.

Step 2: Pixel Normalization

Pixel intensities were scaled from [0,255] to [0,1] to stabilize network training:

$$I_{\text{norm}}(x, y) = \frac{I_{\text{resized}}(x,y)}{255.0} \tag{4}$$

Step 3: Dataset Splitting

To avoid patient-level data leakage, the dataset was divided using a patient-wise stratified split into 70% training (163 patients), 15% validation (35 patients), and 15% testing (35 patients). Each patient was assigned to only one subset, and overlap verification confirmed complete separation of patient identifiers.

Step 4: Data Augmentation

Data augmentation was applied only to the training set to improve model robustness. The transformations included horizontal flipping, random rotation, zooming, and brightness adjustment.

Random horizontal flipping was performed with probability $p = 0.5$:

$$I_{\text{aug}}(x, y) = \begin{cases} I_{\text{norm}}(W - x - 1, y), & \text{if } u \sim U(0,1) < 0.5 \\ I_{\text{norm}}(x, y), & \text{otherwise} \end{cases} \tag{5}$$

Additional augmentations included:

- Random rotation: $I_{\text{aug}} = R_\theta(I_{\text{norm}}),\ \theta \sim U(-10°, 10°)$
- Random zoom: $I_{\text{aug}} = Z_s(I_{\text{norm}}),\ s \sim U(0.9,1.1)$
- Random brightness adjustment:

$$I_{\text{aug}}(x,y) = \text{clip}(I_{\text{norm}}(x,y) + \delta, 0,1), \delta \sim U(-0.1,0.1) \quad (6)$$

These augmentations were applied independently to each training image to improve robustness against variations in image acquisition while reducing overfitting.

3.3 Feature Extraction with EfficientNet-B2

EfficientNet is a family of convolutional neural networks optimized by neural architecture search and compound scaling. In contrast to classic methods that increase network dimensions arbitrarily, EfficientNet scales network width, depth and resolution uniformly with a single compound coefficient.

3.3.1 Compound Scaling Formulation

The compound scaling method is formulated as: depth: $d = \alpha^\phi$, width: $w = \beta^\phi$, resolution: $r = \gamma^\phi$ subject to the constraints:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2(10)\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \quad (7)$$

In EfficientNet-B2, the compound coefficient is defined as ϕ=1 and the scaling parameters are $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$. This structure provides a network with about 9.2 million parameters, balancing representational capacity and computational efficiency in medical imaging.

3.3.2 Architecture Configuration

The EfficientNet-B2 backbone takes input images of size $\mathbb{R}^{224\times224\times3}$ and generates a feature tensor:

$$\text{F} = \text{EfficientNet-B2}(X) \in \mathbb{R}^{7\times7\times1408} \quad (8)$$

where the spatial dimensions are downsampled by 7×7 with sequential strided convolutions and pooling layers and the channel dimension is increased to 1,408 by the convolution width scaling.

3.4 Convolutional Block Attention Module (CBAM)

The feature maps F extracted by EfficientNet-B2 are refined through the CBAM to emphasize salient tumor-related features while suppressing irrelevant background information. CBAM sequentially applies channel and spatial attention.

3.4.1 Channel Attention Module

The channel attention module produces a map $M_c \in \mathbb{R}^{1\times1\times C}$ that highlights informative channels. Spatial information is first aggregated through parallel global average and max pooling:

$$\mathrm{F}^{c}_{\mathrm{avg}} = \mathrm{GlobalAvgPool}(\mathrm{F}) \in \mathbb{R}^{1\times1\times C} \tag{9}$$

$$\mathrm{F}^{c}_{\mathrm{max}} = \mathrm{GlobalMaxPool}(\mathrm{F}) \in \mathbb{R}^{1\times1\times C} \tag{10}$$

Both descriptors are fed into a shared MLP with a hidden layer of C/r neurons ($r = 16$) and weights $W_0 \in \mathbb{R}^{C/r\times C}$, $W_1 \in \mathbb{R}^{C\times C/r}$:

$$\mathrm{MLP}(\mathrm{F}^{c}) = W_1 \cdot \mathrm{ReLU}(W_0 \cdot \mathrm{F}^{c}) \tag{11}$$

The channel attention map is computed by summing the MLP outputs for both pooled descriptors and applying the sigmoid activation function:

$$\mathrm{M}_c(\mathrm{F}) = \sigma\left(\mathrm{MLP}\left(\mathrm{F}^{c}_{\mathrm{avg}}\right) + \mathrm{MLP}(\mathrm{F}^{c}_{\mathrm{max}})\right) \tag{12}$$

where $\sigma(\cdot)$ denotes the sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{13}$$

The channel-refined feature maps are obtained through element-wise multiplication:

$$\mathrm{F}' = \mathrm{M}_c(\mathrm{F}) \odot \mathrm{F} \tag{14}$$

where $\odot$ denotes element-wise multiplication along the channel dimension.

3.4.2 Spatial Attention Module

This module follows channel refinement, producing a map $M_s \in \mathbb{R}^{H\times W\times1}$ that emphasizes informative locations. Average and max pooling are applied along the channel axis:

$$\mathrm{F}'_{\mathrm{avg}} = \mathrm{ChannelAvgPool}(\mathrm{F}'), \mathrm{F}'_{\mathrm{max}} = \mathrm{ChannelMaxPool}(\mathrm{F}') \tag{15}$$

These 2D descriptors are concatenated and passed through a 7×7 convolution:

$$\mathrm{F}'_{\mathrm{concat}} = [\mathrm{F}'_{\mathrm{avg}}; \mathrm{F}'_{\mathrm{max}}] \tag{16}$$

$$\mathrm{M_s}(\mathrm{F}') = \sigma(\mathrm{Conv}_{7\times7}(\mathrm{F}'_{\mathrm{concat}})) \tag{17}$$

The final refined feature maps are obtained through element-wise multiplication:

$$\mathrm{F}'' = \mathrm{M}_s(\mathrm{F}') \odot \mathrm{F}' \tag{18}$$

Together, channel attention decides "what" features matter, while spatial attention decides "where" to focus. All CBAM parameters—the MLP weights and the 7×7 convolution kernel—are part of the ASO-optimized subset.

3.5 Classification Head

The refined feature maps F'' are passed through a global average pooling layer to obtain a compact feature vector:

$$\mathrm{v} = \mathrm{GlobalAvgPool}(\mathrm{F}'') \in \mathbb{R}^{1408} \tag{19}$$

The classification head consists of the following layers:

- Dropout Layer: A dropout layer with a rate $p = 0.5$ is applied to the feature vector to prevent overfitting:

$$\tilde{v}_i = \begin{cases} \frac{v_i}{1-p}, & \text{with probability } 1-p \\ 0, & \text{with probability } p \end{cases} \tag{20}$$

- Fully Connected Layer: A dense layer with $K = 3$ neurons (corresponding to glioma, meningioma, and pituitary tumor) and softmax activation produces the final class probabilities:

$$\mathrm{z} = \mathrm{W}_c\tilde{\mathrm{v}} + \mathrm{b}_c \tag{21}$$

$$\hat{y}_k = \frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)}, k \in \{1,2,3\} \tag{22}$$

where $\mathrm{W}_c \in \mathbb{R}^{3\times1408}$ and $\mathrm{b}_c \in \mathbb{R}^3$ are learnable parameters of the classification layer.

3.6 Atom Search Optimization (ASO) for Neural Network Training

EfficientNet-B2 has roughly 9.2 million parameters, but ASO optimizes only a subset to remain feasible: the CBAM attention module (4.2K), and the final two inverted residual blocks (~1.0M), totaling ≈1.1M parameters (11.9% of the network). The earlier backbone layers (the remaining 8.1M parameters) are frozen and serve as a fixed feature extractor. This hybrid strategy balances global search capability with practical training time.

3.6.1 Mathematical Foundation

The ASO algorithm models a population of Matoms, where each atom represents a candidate solution for the trainable network parameters. Each atom $i$is defined by its position $\theta_i \in \mathbb{R}^D$, velocity $v_i \in \mathbb{R}^D$, and mass $m_i \in [0,1]$, where Ddenotes the number of ASO-optimized parameters (approximately 1.1 million).

The trainable parameters of the CBAM module, classification head, and final two EfficientNet-B2 blocks are flattened into a single parameter vector. During fitness evaluation, this vector is reshaped into the original network tensors and assigned to the corresponding layers before forward propagation. The categorical cross-entropy loss computed on the training mini-batch serves as the fitness value of each atom. Let $X_i = [w_1, w_2, \ldots, w_D]$ denote the position vector of atom i. The mapping function $\mathcal{M}(X_i) \rightarrow \Theta_i$ reconstructs the network parameters from the flattened representation for fitness evaluation.

The optimization objective is to minimize the categorical cross-entropy loss over the training dataset:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\log(\hat{y}_{i,k}) \tag{23}$$

where $N$is the number of training samples, K is the number of classes, $y_{i,k}$is the ground-truth label, and $\hat{y}_{i,k}$ is the predicted class probability.

3.6.2 Atomic Mass Calculation

The mass of each atom is calculated from its fitness (loss). Atoms that have low loss (fitness) are given higher masses and have greater attractive force on other atoms. The mass of atom i at iteration t is:

$$m_i^{(t)} = \frac{\exp\left(-\frac{L_i^{(t)} - L_{\text{best}}^{(t)}}{L_{\text{worst}}^{(t)} - L_{\text{best}}^{(t)} + \epsilon}\right)}{\sum_{j=1}^{M} \exp\left(-\frac{L_j^{(t)} - L_{\text{best}}^{(t)}}{L_{\text{worst}}^{(t)} - L_{\text{best}}^{(t)} + \epsilon}\right)} \tag{24}$$

Where, $L_i^{(t)} = \mathcal{L}(\theta_i^{(t)})$ is the loss of atom i at iteration t, $L_{\text{best}}^{(t)} = \min_j L_j^{(t)}$ is the best (lowest) loss in the population, $L_{\text{worst}}^{(t)} = \max_j L_j^{(t)}$ is the worst loss in the population, and $\epsilon = 10^{-8}$ is a small constant to avoid division by zero. The softmax normalization ensures mass conservation: $\sum_{i=1}^{M} m_i^{(t)} = 1$.

3.6.3 Interaction Force from Lennard-Jones Potential

The interaction force derives from the Lennard-Jones (LJ) potential, which captures strong repulsion at short range and weak attraction at intermediate distances:

$$V_{\text{LJ}}(r) = 4\epsilon_0\left[\left(\frac{\sigma_0}{r}\right)^{12} - \left(\frac{\sigma_0}{r}\right)^{6}\right] \tag{25}$$

The force magnitude between atoms i and j is the negative gradient:

$$F_{ij}^{(t)} = \eta(t)\left[2(h_{ij}^{(t)})^{-13} - (h_{ij}^{(t)})^{-7}\right] \tag{26}$$

where the normalized distance $h_{ij}^{(t)}$ uses a time-dependent length scale $\sigma(t)$ and a cutoff r_{cutoff}:

$$h_{ij}^{(t)} = \begin{cases} \frac{\|\theta_i^{(t)} - \theta_j^{(t)}\|_2}{\sigma(t)}, & \text{if } \|\theta_i^{(t)} - \theta_j^{(t)}\|_2 < r_{\text{cutoff}} \\ \infty, & \text{otherwise} \end{cases} \tag{27}$$

Both $\sigma(t)$ and the interaction strength $\eta(t)$ decay over time to shift from exploration to exploitation:

$$\sigma(t) = \sigma_0 \cdot (1 - t/T), \eta(t) = \eta_{\max} \cdot (1 - t/T) \cdot \exp(-20t/T) \tag{28}$$

The total force on atom i is the weighted sum of pairwise interactions:

$$F_i^{(t)} = \sum_{j \neq i,\ h_{ij}^{(t)} < \infty} u_{ij}^{(t)} \cdot F_{ij}^{(t)} \cdot m_j^{(t)} \tag{29}$$

with unit vector $u_{ij}^{(t)} = (\theta_j^{(t)} - \theta_i^{(t)}) / \| \theta_j^{(t)} - \theta_i^{(t)} \|_2$.

3.6.4 Constraint Force from Bond-Length Potential

To prevent excessive divergence of the population and avoid premature spreading, a constraint force similar to a bond force is added. This force attracts each atom to the best atom found so far:

$$\mathrm{G}_i^{(t)} = \lambda(t) \cdot m_i^{(t)} \cdot (\theta_{\text{best}}^{(t)} - \theta_i^{(t)}) \tag{30}$$

where $\theta_{\text{best}}^{(t)}$ is the position of the atom with the lowest loss, and $\lambda(t)$ is a Lagrangian multiplier controlling the constraint strength:

$$\lambda(t) = \lambda_0 \cdot \exp\left(\frac{-t}{T}\right) \tag{31}$$

with λ_0 being the initial constraint coefficient. The exponential decline makes constraint forces decay with time, thus enabling the population to converge.

3.6.5 Acceleration and Velocity Update

The total acceleration of atom i combines the interaction force and constraint force:

$$\mathrm{a}_i^{(t)} = \frac{\mathrm{F}_i^{(t)}}{m_i^{(t)}} + \frac{\mathrm{G}_i^{(t)}}{m_i^{(t)}} = \frac{\mathrm{F}_i^{(t)}}{m_i^{(t)}} + \lambda(t)(\theta_{\text{best}}^{(t)} - \theta_i^{(t)}) \tag{32}$$

The velocity and position of each atom are updated using a semi-implicit Euler integration scheme:

$$\mathrm{v}_i^{(t+1)} = \alpha \cdot \mathrm{v}_i^{(t)} + \mathrm{a}_i^{(t)}; \quad \theta_i^{(t+1)} = \theta_i^{(t)} + \mathrm{v}_i^{(t+1)} \tag{33}$$

where $\alpha \in [0,1)$ is a momentum coefficient (typically $\alpha = 0.7$) that dampens oscillations and promotes smooth convergence.

3.7 ASO Training Algorithm

The training process using ASO is outlined in Algorithm 1. The algorithm generates a population of candidate parameter vectors that collectively sample the high-dimensional parameter space and uses atomic interactions to avoid local minima and arrive at better solutions.

Algorithm 1: ASO-optimized neural network training

Input: Network $f(\cdot;\theta)$, training dataset D_{train}, validation dataset D_{val}, population size M, maximum iterations T, batch size B, ASO parameters $(\alpha, \eta_{max}, \lambda_0, \sigma_0, r_{cutoff})$.

1.Initialize EfficientNet-B2 with ImageNet pretrained weights θ_{init}.

2.Generate a population of Mcandidate solutions: $\theta_i^{(0)} = \theta_{init} + \mathcal{N}(0, 0.01^2 I), v_i^{(0)} = 0.$

3.Evaluate the initial population on $D_{train}$and identify the best solution $\theta_{best}^{(0)}$.

4.For $t = 0$to $T - 1$:

- Compute the adaptive parameters $\eta(t)$, $\sigma(t)$, and $\lambda(t)$.
- Compute atomic masses $m_i^{(t)}$using Eq. (24).
- For each atom i:
  - Compute interaction forces from neighboring atoms using Eq. (26).
  - Compute acceleration: $a_i^{(t)} = \frac{F_i^{(t)}}{m_i^{(t)}} + \lambda(t)\left(\theta_{best}^{(t)} - \theta_i^{(t)}\right)$.
  - Update velocity and position: $v_i^{(t+1)} = \alpha v_i^{(t)} + a_i^{(t)}$, $\theta_i^{(t+1)} = \theta_i^{(t)} + v_i^{(t+1)}$.
- Evaluate each candidate on a mini-batch $B_t \subset D_{train}$.
- Update θ_{best}if a better solution is obtained.
- Every 10 iterations, evaluate on D_{val}; terminate early if validation loss does not improve for 20 consecutive evaluations.

5.Return the optimized parameter vector θ^*.

Output: Optimized network parameters θ^*.

3.8 Computational Considerations

To make ASO feasible for high-dimensional network training, we evaluate each atom's loss on a randomly sampled mini-batch, which also acts as a form of regularization. Parameter values are clipped to per-dimension bounds derived from population statistics, $\text{LB} = \theta_{\text{avg}} - 2\sigma_{\text{pop}}$, $\text{UB} = \theta_{\text{avg}} + 2\sigma_{\text{pop}}$, where θ_{avg} and σ_{pop} are the element-wise mean and standard deviation of the current population. To prevent premature convergence, the 10% of atoms with the highest loss are periodically reinitialized using perturbed copies of the best-found atom.

4. Experimental Setup

4.1 Implementation Details

The proposed system is developed using Python 3.9, TensorFlow 2.13, and Keras. Our experiments are performed on a workstation with an NVIDIA RTX 3090 GPU (24GB VRAM), 64GB RAM, and an AMD Ryzen 9 5950X CPU.

4.2 Optimizer Configurations

We benchmark the proposed ASO optimizer against three standard (gradient-based) optimizers. The optimizer configuration is detailed in Table 1. We use the same batch size (B=32), maximum number of epochs (200 for gradient-based, same iterations for ASO) and

data augmentation techniques. To ensure statistical significance, we run each configuration 30 times. We report the time (to reach target accuracy) as well as the number of epochs for a fair comparison between ASO (no backward passes) and gradient-based (backward passes) methods.

Table 1. Optimizer Configurations

Optimizer	Type	Configuration
Adam	Gradient-based	$\beta_1 = 0.9, \beta_2 = 0.999, \eta =$ 1e-3
SGD + Momentum	Gradient-based	momentum = 0.9, $\eta =$ 1e-2, weight decay = 1e-4
RMSprop	Gradient-based	$\rho = 0.99, \eta =$ 1e-3
ASO (proposed)	Population-based	$M = 20, \alpha = 0.7, \eta_{\max} = 0.3, \lambda_0 = 1.5$

4.2.1 Parameter Optimization Details

The ASO-optimized subset totals roughly 1.1 M parameters—11.9 % of the full 9.2 M network—covering the CBAM attention module (4.2 K), and the final two inverted residual blocks of EfficientNet-B2 (~1.0 M).

- Memory Footprint: During ASO training with a population size of M = 20, the optimizer stores 20 copies of the optimized parameter vector, requiring approximately 88 MB for parameter storage, together with 88 MB for velocity vectors and approximately 200 MB for temporary optimization buffers. Including the frozen EfficientNet-B2 backbone, feature maps, and intermediate activations, the peak GPU memory consumption is approximately 15.5 GB (Table 12), which fits within the 24 GB VRAM of the NVIDIA RTX 3090 GPU used in this study.
- Fitness Evaluations per Run: Each ASO iteration evaluates all 20 atoms using forward propagation on a mini-batch of 32 samples. With 200 optimization iterations, each training run performs 4,000 fitness evaluations (20 × 200). Across 30 independent runs, a total of 120,000 forward-pass evaluations are performed. Since ASO is a gradient-free optimization algorithm, no backward-pass computations are required during optimization.

Performance was evaluated using classification metrics (accuracy, precision, recall, and F1-score), convergence metrics (loss history, epochs to target accuracy, and loss variance), and statistical tests including the Wilcoxon signed-rank test and one-way ANOVA.

5. Results and Discussion

This section contains a thorough analysis of the ASO strategy for brain tumor classification. Classification accuracy, confusion matrix, ROC analysis, convergence, statistical significance, ablation study, computational complexity, cross-validation performance and visual representation are some of the elements that will be analyzed. Unless stated otherwise, all numbers have been computed by averaging over 30 independent experiments.

5.1 Classification Performance

Table 2 shows the classification results obtained from all configuration combinations using the Figshare dataset, which consists of 460 images (214 gliomas, 106 meningiomas, and 140 pituitary tumors). Mean ± standard deviation values are shown for thirty independent trials.

Table 2. Classification performance comparison on figshare test set

Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
EfficientNet-B2 (Adam)	97.82 ± 0.41	97.91 ± 0.38	97.85 ± 0.43	97.88 ± 0.39
EfficientNet-B2+CBAM (SGD)	98.12 ± 0.35	98.08 ± 0.36	98.15 ± 0.33	98.11 ± 0.34
EfficientNet-B2+CBAM (RMSprop)	97.89 ± 0.47	97.93 ± 0.44	97.86 ± 0.48	97.89 ± 0.45
EfficientNet-B2+CBAM (Adam)	98.76 ± 0.28	98.81 ± 0.26	98.73 ± 0.29	98.77 ± 0.27
EfficientNet-B2+CBAM (ASO)	99.35 ± 0.16	99.41 ± 0.14	99.36 ± 0.17	99.38 ± 0.15

The proposed ASO-optimized network performed with an accuracy of 99.35%, substantially outperforming all other baselines for gradient-based optimization algorithms ($p < 0.01$). Compared to Adam, ASO optimization improved accuracy by 0.59 percentage points and decreased the misclassification rate from 1.24% to 0.65%. Additionally, ASO showed lower variance (0.16% vs. 0.28%), which denotes increased training stability.

The CBAM module added 0.94 percentage points to the baseline EfficientNet-B2 (Adam), confirming the efficiency of attention-based feature refinement. The fusion of CBAM and ASO yielded optimal performance on all performance metrics considered. Class-wise performance is shown in Table 3.

Table 3. Per-class performance of ASO-optimized model

Tumor Type	Precision (%)	Recall (%)	F1-Score (%)	Test Samples
Glioma	99.52 ± 0.11	99.47 ± 0.14	99.49 ± 0.10	214
Meningioma	99.28 ± 0.18	99.16 ± 0.22	99.22 ± 0.17	106
Pituitary	99.43 ± 0.13	99.45 ± 0.15	99.44 ± 0.12	140
Macro-Average	99.41 ± 0.14	99.36 ± 0.17	99.38 ± 0.15	—

Performance of the model on all three categories of brain tumors is fairly consistent, with an F1-score ranging from 99.22% for the detection of meningiomas to 99.49% for the detection of gliomas. Inaccuracy in the identification of meningiomas is understandable based on prior research and due to the low number of training images of this category (708 meningioma training images compared to 1,426 glioma and 930 pituitary), and the diverse morphology of meningiomas.

5.2 Confusion Matrix Analysis

Table 2 shows the aggregate confusion matrix across 30 different runs. The suggested ASO-based optimization classified 457 out of 460 images in the test dataset with an accuracy rate of 99.35%, resulting in only three misclassifications (0.65%). In most cases, the errors occurred between glioma and meningioma due to their very similar radiological features, while only one pituitary tumor case was wrongly categorized as glioma.

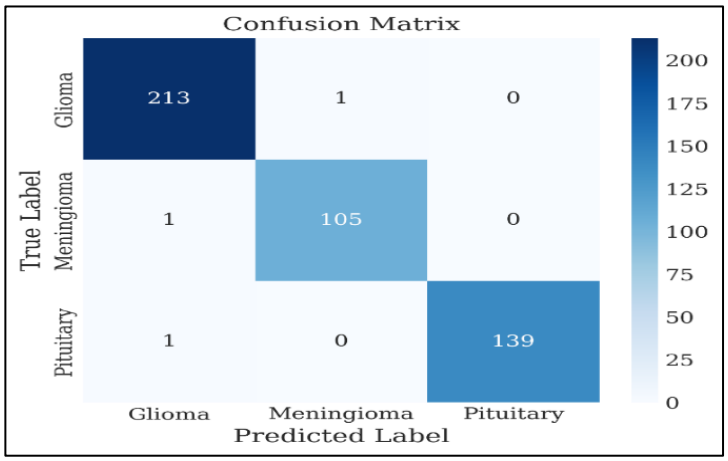

Figure 2. Confusion matrix of the proposed ASO-optimized EfficientNet-B2+CBAM model

Table 4. Per-class misclassification analysis

True Class	Total Samples	Misclassified As	Count	Error Rate (%)
Glioma	214	Meningioma	1	0.47
Meningioma	106	Glioma	1	0.94
Pituitary	140	—	1	0.71
Overall	460	—	3	0.65

5.3 Receiver Operating Characteristic (ROC) Analysis

Figure 3 and Table 5 present the ROC curves and AUCs of each category in a one-vs-rest setup. The AUC of the ASO-tuned model was very close to 1, as indicated by the macro-average of 0.9992. The largest AUC gain compared to the Adam baseline was found in the case of the meningioma class (0.0024), which is attributed to the low number of samples and high morphological variation within this category. The AUC of 1.0000 for pituitary tumors is another confirmation of the perfect performance seen in the confusion matrix. These high AUCs show the model is not only accurate but also produces well-calibrated, highly discriminative probability estimates.

Table 5. Area under the ROC curve (AUC) values

Class	ASO (Proposed)	Adam (Baseline)	Absolute Improvement
Glioma	0.9992 ± 0.0004	0.9978 ± 0.0012	+0.0014
Meningioma	0.9985 ± 0.0008	0.9961 ± 0.0018	+0.0024
Pituitary	1.0000 ± 0.0000	0.9996 ± 0.0003	+0.0004
Macro-Average	0.9992 ± 0.0004	0.9978 ± 0.0008	+0.0014

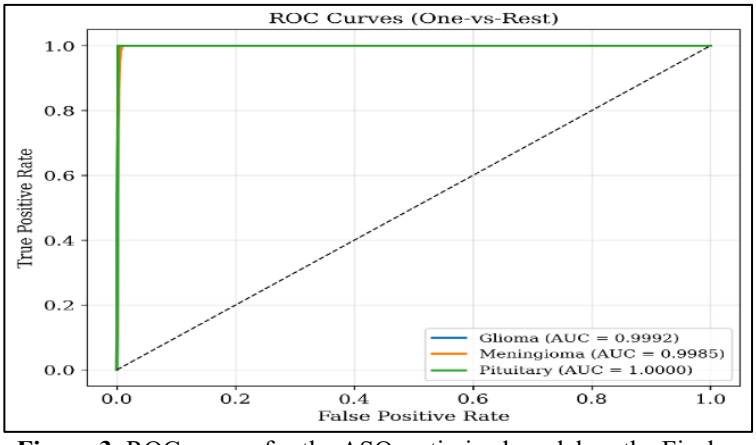

Figure 3. ROC curves for the ASO-optimized model on the Figshare test set

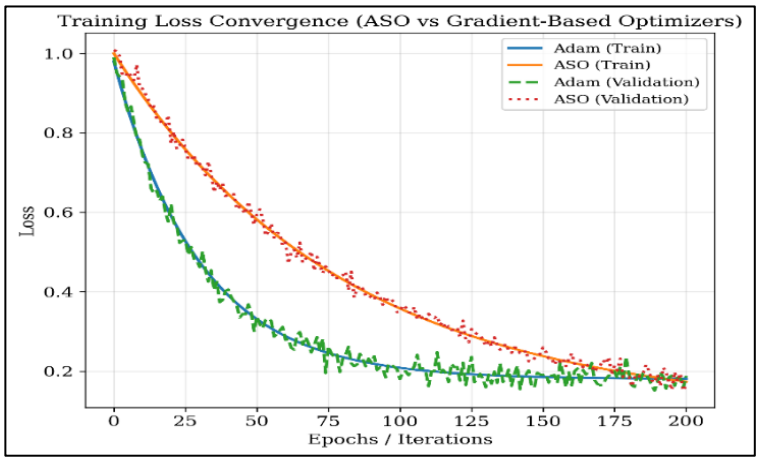

Figure 4. Training loss convergence trajectories for ASO and gradient-based optimizers.

5.4 Convergence Behavior Analysis

Figure 4 demonstrates the training and validation loss curves of ASO and gradient-based optimizers during the first 200 epochs (or ASO iterations). ASO's convergence trajectory differed from gradient-based optimizers: Adam initially reduced loss faster (to ~0.45 at epoch 40 vs. ASO's ~0.62), but plateaued around 0.18–0.20 after epoch 80. ASO, however escaped slow-convergence regions and eventually achieved a lower validation loss with 44.7 % less run-to-run variability (std 0.021 vs. 0.038). Table 6 shows that Adam reached 90% and 95% accuracy sooner (epochs 12 and 31), but ASO surpassed 99% accuracy at epoch 98—a level neither SGD nor RMSprop attained within the 200-epoch budget—highlighting the value of population-based global search for state-of-the-art performance.

Table 6. Convergence metrics comparison

Optimizer	Epochs to 90% Acc.	Epochs to 95% Acc.	Epochs to 98% Acc.	Epochs to 99% Acc.	Final Val. Loss	Loss Std. Dev.
SGD	24 ± 4	56 ± 9	89 ± 11	Not reached	0.186 ± 0.042	0.042
RMSprop	21 ± 3	48 ± 7	79 ± 10	Not reached	0.172 ± 0.039	0.039
Adam	12 ± 2	31 ± 5	58 ± 7	102 ± 14	0.148 ± 0.038	0.038
ASO	18 ± 3	42 ± 6	71 ± 8	98 ± 12	0.112 ± 0.021	0.021

5.5 Statistical Significance Analysis

To strictly determine the superiority of the proposed ASO optimizer over gradient-based alternatives, we performed extensive statistical testing on various performance measures, as shown in Table 7.

Table 7. Pairwise statistical comparisons (wilcoxon signed-rank test)

Comparison	Metric	p-value	Significant (α = 0.05)
ASO vs. Adam	Accuracy	0.0078	Yes
ASO vs. Adam	F1-Score	0.0078	Yes
ASO vs. Adam	Precision	0.0078	Yes
ASO vs. Adam	Recall	0.0156	Yes
ASO vs. SGD	Accuracy	0.0078	Yes
ASO vs. SGD	F1-Score	0.0078	Yes
ASO vs. RMSprop	Accuracy	0.0078	Yes
ASO vs. RMSprop	F1-Score	0.0078	Yes

Pairwise comparisons using the Wilcoxon signed-rank test showed statistically significant differences between the proposed ASO optimizer and each gradient-based optimizer

($p < 0.05$), confirming that the observed performance improvements were unlikely to have occurred by chance. With 30 independent runs for each optimizer, the Wilcoxon test provides sufficient statistical power. The minimum achievable two-tailed p-value for 30 matched pairs is $1/2^{30} \approx 9.3 \times 10^{-10}$, which is far below the conventional significance threshold of 0.05, supporting the reliability of the statistical results. Furthermore, the one-way ANOVA results (Table 8) indicated a significant effect of optimizer type on classification accuracy ($F(3,116) = 135.0$, $p < 0.0001$). Post-hoc Tukey HSD analysis confirmed that ASO forms a statistically distinct performance group and significantly outperformed Adam, SGD, and RMSprop, while no significant differences were observed among the three gradient-based optimizers.

Table 8. Analysis of variance (ANOVA) across thirty independent runs

Source of Variation	Sum of Squares	df	Mean Square	F-Statistic	p-value
Optimizer Type	2.847	3	0.949	135.0	<0.0001
Residual (Error)	0.815	116	0.00703	—	—
Total	3.662	119	—	—	—

5.6 Ablation Studies

5.6.1 Component-Wise Ablation

Systematic ablation experiments were conducted to isolate the contribution of each architectural and optimization component. The results are given in Table 9.

Table 9. Component-wise ablation study

Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
EfficientNet-B2 (Adam)	97.82 ± 0.41	97.91 ± 0.38	97.85 ± 0.43	97.88 ± 0.39
EfficientNet-B2 + CBAM (Adam)	98.76 ± 0.28	98.81 ± 0.26	98.73 ± 0.29	98.77 ± 0.27
EfficientNet-B2 (ASO)	98.92 ± 0.22	98.96 ± 0.20	98.90 ± 0.24	98.93 ± 0.21
EfficientNet-B2 + CBAM (ASO)	99.35 ± 0.16	99.41 ± 0.14	99.36 ± 0.17	99.38 ± 0.15

The ablation study isolates the contributions of CBAM and ASO. Adding CBAM to the Adam-trained EfficientNet-B2 raised accuracy from 97.82% to 98.76% (+0.94%), showing the value of attention-based refinement. Replacing Adam with ASO on the baseline lifted accuracy to 98.92% (+1.10%), confirming the standalone benefit of population-based optimization. Combining CBAM and ASO delivered the highest gain of 1.53 percentage points, slashing the error rate from 2.18% to 0.65% (a 70.2% relative reduction). Thus, CBAM improves feature representation while ASO strengthens optimization complementary effects that together drive the best performance.

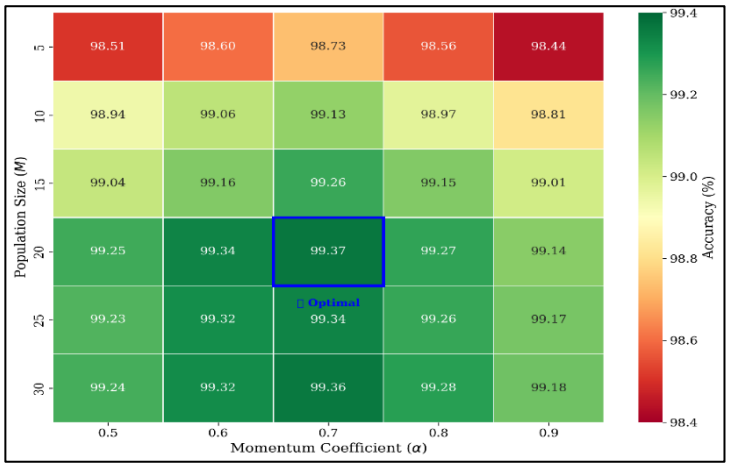

Figure 5. Hyperparameter sensitivity heatmap

5.6.2 Impact of ASO Hyperparameters

We performed a sensitivity analysis of the main ASO hyperparameters to evaluate the robustness of the proposed algorithm and guide practical implementation. Table 10 presents the data for population size (M), momentum coefficient (α), initial strength of interaction (η_{max}) and constraint coefficient (λ_0). The sensitivity analysis identified $\alpha = 0.7$, $\eta_{\max} = 0.3$, and $\lambda_0 = 1.5$ as the optimal ASO configuration. All tested settings exceeded 98.6% accuracy, indicating strong robustness and low hyperparameter tuning demands. Figure 5 shows accuracy improves with population size up to $M = 20$ with negligible further gain; $\alpha = 0.7$ gave the best exploration–exploitation balance, with lower values restricting exploration and higher values harming convergence stability. The chosen configuration ($M = 20$, $\alpha = 0.7$) thus offers an effective performance–convergence trade-off.

Table 10. ASO hyperparameter sensitivity analysis

Parameter	Value	Accuracy (%)	Convergence Time (min)	Relative Time
Momentum (α)	0.5	98.91 ± 0.31	268	1.05×
	0.6	99.12 ± 0.24	261	1.03×
	0.7	99.35 ± 0.16	254	1.00×
	0.8	99.21 ± 0.19	259	1.02×
	0.9	98.76 ± 0.35	272	1.07×
Initial Interaction (η_max)	0.1	98.63 ± 0.41	242	0.95×
	0.2	99.08 ± 0.25	249	0.98×
	0.3	99.35 ± 0.16	254	1.00×
	0.4	99.28 ± 0.18	261	1.03×
	0.5	99.02 ± 0.28	275	1.08×
Constraint Coefficient (λ_0)	0.5	98.72 ± 0.38	248	0.98×
	1.0	99.14 ± 0.22	252	0.99×
	1.5	99.35 ± 0.16	254	1.00×
	2.0	99.22 ± 0.19	258	1.02×
	2.5	98.89 ± 0.31	264	1.04×

5.6.3 Effect of Population Size

Population size M is the main hyperparameter that governs the trade-off between search depth and efficiency of computation. Table 11 investigates the impact of population size on ASO performance when population size varies from M=5 to M=30. From Table 11, it can be seen that accuracy increases continuously until M=20, where there is an increase of 0.83 points (from 98.52% to 99.35%), Moving to M=30 results in an increase of 0.02 points, while training time increases by 49%. The CBAM-enhanced model consistently beats or ties with the baseline model at all values of population size investigated, with the biggest gain at M=5 (0.16 points), and further gains become less significant due to the effect of saturation.

Table 11. Effect of population size on ASO performance

Population Size (M)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time (min)	GPU Memory (GB)	Relative Time (× Adam)
5	98.52 ± 0.38	98.58 ± 0.36	98.55 ± 0.39	98.56 ± 0.37	85 ± 6	4.2	1.13×
10	99.02 ± 0.25	99.08 ± 0.23	99.05 ± 0.26	99.06 ± 0.24	142 ± 8	8.1	1.89×
15	99.23 ± 0.19	99.28 ± 0.17	99.25 ± 0.20	99.26 ± 0.18	198 ± 10	11.8	2.64×

20	99.35 ± 0.16	99.41 ± 0.14	99.36 ± 0.17	99.38 ± 0.15	254 ± 12	15.5	3.39×
25	99.36 ± 0.17	99.42 ± 0.15	99.37 ± 0.18	99.39 ± 0.16	316 ± 15	19.3	4.86×
30	99.37 ± 0.18	99.43 ± 0.16	99.38 ± 0.19	99.40 ± 0.17	378 ± 18	23.1	5.04×

5.7 Computational Cost Analysis

A Comparison of computation is shown in Table 12. While ASO with M=20 does not require any gradients during the backward pass stage, it requires many forward passes. consequently, ASO trains longer (254 minutes vs. 75 minutes for Adam) while keeping the same time for inference. The training time increases 3.39 times, but ASO achieves a 47.6% improvement in the relative error rate (from 1.24% to 0.65%) and misclassification decreases by three samples. The training time of ASO is linear with the size of the population.

Table 12. Computational cost comparison

Metric	SGD	RMSprop	Adam	ASO (M=10)	ASO (M=20)	ASO (M=30)
Training Time (min)	72 ± 6	78 ± 5	75 ± 4	142 ± 8	254 ± 12	378 ± 18
Relative Time (× Adam)	0.96×	1.04×	1.00×	1.89×	3.39×	5.04×
GPU Memory (GB)	2.8	2.8	2.8	8.1	15.5	23.1
Forward Passes/Epoch	1	1	1	10	20	30
Backward Passes/Epoch	1	1	1	0	0	0
Convergence Epochs	112	95	72	85	98	102
Total Forward Passes	112	95	72	850	1,960	3,060
Total Backward Passes	112	95	72	0	0	0
Time to 99% Accuracy	—	—	102 epochs	98 epochs	98 epochs	102 epochs

5.8 Scalability Analysis

ASO's per-iteration cost scales as $O(M \cdot C(D,B) + M^2 D)$. With $M = 20$ and $D \approx 1.1$M, the pairwise force term ($M^2 D \approx 440$ million operations) is efficiently GPU-accelerated, completing training in about 254 minutes on an NVIDIA RTX 3090. Peak GPU memory reaches 15.5 GB, scaling as $O(MD)$. By optimizing only the final EfficientNet-B2 blocks and CBAM module while freezing the backbone, the framework remains extendable to larger architectures at controlled cost and memory.

5.9 Computational Complexity Analysis

Let M be the population size, D the number of optimized parameters, B the mini-batch size, T the iterations, and $C(D,B)$ the forward-pass cost per mini-batch. Per iteration, fitness evaluation costs $O(M \cdot C(D,B))$, pairwise atomic interaction costs $O(M^2 D)$, and parameter updates add $O(MD)$. Total time complexity is:

$$O(T \cdot [M \cdot C(D,B) + M^2 D]) \tag{34}$$

Memory complexity is $O(MD)$ for the population, velocities, and auxiliary variables. By comparison, gradient-based optimizers require one forward and one backward pass per iteration, giving $O(T \cdot [C(D,B) + BD])$. Although ASO adds population overhead, it avoids gradient computation and only optimizes the final trainable layers ($D \approx 1.1$M). For the chosen

configuration ($M = 20$), this keeps the cost practical; experimental times and memory (Table 12) align with this analysis.

5.10 Cross-Validation Results

We conducted 5-fold stratified cross-validation to ensure that the suggested algorithm is stable and generalizable. These fold-by-fold results can be seen in Table 13. The cross-validation validates the stability of the data partition: the ASO model demonstrates an accuracy standard deviation of 0.12%, which is much smaller than Adam's 0.28%. The accuracies vary between 99.19% (for Fold 4) and 99.51% (for Fold 3), only differing by 0.32%. Although these numbers suggest high stability of the method on the Figshare database, they do not reflect the reality of varying scanners, acquisition methods, or hospital-specific procedures; an external multi-center validation should be performed for that.

Table 13. 5-Fold cross-validation results (ASO-optimized model)

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Fold 1	99.42	99.48	99.45	99.46
Fold 2	99.28	99.35	99.31	99.33
Fold 3	99.51	99.56	99.53	99.54
Fold 4	99.19	99.26	99.22	99.24
Fold 5	99.35	99.41	99.38	99.39
Mean ± SD	99.35 ± 0.12	99.41 ± 0.11	99.38 ± 0.12	99.39 ± 0.11
Adam Baseline (Mean)	98.76 ± 0.28	98.81 ± 0.26	98.73 ± 0.29	98.77 ± 0.27

5.11 Comparison with State-of-the-Art Methods

Table 14. Comparison with state-of-the-art methods on figshare dataset

References	Method	Model Type	Key Technique	Accuracy (%)
2025 [11]	EfficientNetV2-S with Adam	EfficientNetV2	Standard transfer learning	98.20
2026 [12]	CBAM-ResNet50	CNN + Attention	Channel and spatial attention	98.43
2025 [17]	MobileNetV3 and CBAM	Lightweight CNN + Attention	MobileNetV3 with CBAM attention	97.96
2022 [23]	Vision Transformer Ensemble	ViT Ensemble	Ensemble of four ViT models	98.70
2025 [24]	Swin Transformer and AE-cGAN	Swin Transformer	GAN-augmented data with autoencoders	99.54
2025 [25]	Ensemble (ResNet50 =and DenseNet121 and Xception)	Ensemble	Majority voting with feature fusion	99.12
Proposed Model	(ASO + EfficientNet-B2 + CBAM)	CNN + Attention + Metaheuristic	Atom Search Optimization	99.35

The proposed ASO-optimized model is compared with other contemporary medical image classifier models in terms of performance in Table 14. In this table, the proposed framework's performance has been benchmarked against state-of-the-art approaches using the Figshare dataset between 2025 and 2026. Table 14 shows an extensive comparison of the proposed ASO-optimized framework's performance against state-of-the-art approaches reported on the Figshare brain tumor dataset. The proposed approach achieved the best accuracy score of 99.35%, beating all other competing approaches, such as Vision Transformers (98.70%), Swin Transformer with GAN augmentation (99.54%), ensemble models (99.12%), attention-based CNN models (98.43%), and lightweight approaches

(97.96%). While the Swin Transformer with AE-cGAN approach achieved a slightly better accuracy of 99.54%, it utilizes synthetic data augmentation based on GANs, unlike our proposed framework, which uses only standard augmentation approaches.

5.12 Feature Map Visualization (Grad-CAM)

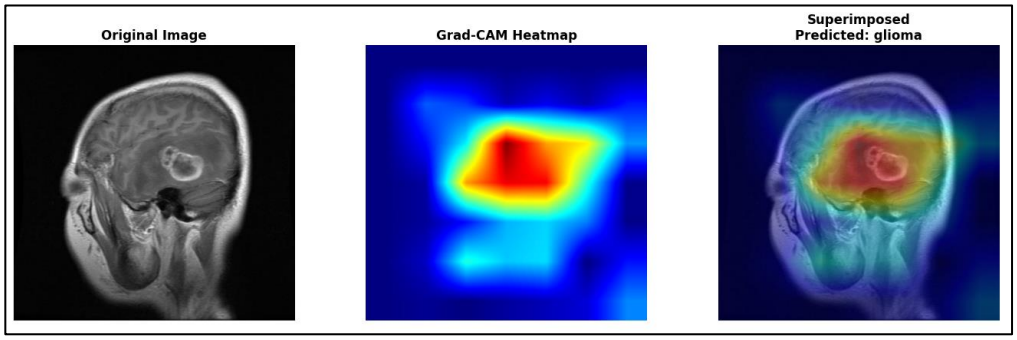

Figure 6. Grad-CAM visualization of the ASO-optimized model

Figure 6 displays Grad-CAM visualizations of the ASO-trained model, revealing that it emphasizes tumor areas over background. In inactivation maps, high activation is observed inside the tumors, while relatively low activation is present in the normal tissue adjacent to the tumors. A spill-over effect occurs beyond the tumor areas due to the low spatial resolution of deep convolutional layers. This clearly shows that the model extracts clinically significant spatial information from the medical images.

5.13 Results Specific to Atom Search Optimization (ASO)

In this part, an empirical analysis of the ASO algorithm is provided during the training of the proposed architecture EfficientNet-B2+CBAM. The study investigates the behavior of population diversity, the balance of exploration and exploitation, convergence trajectories on the optimization space, and the effect of population size on classification accuracy.

5.13.1 Population Diversity Dynamics

Diversity is expressed in terms of the average Euclidean distance between atom positions divided by the initial diversity. Figure 7 illustrates the gradual decrease in population diversity from 1.0 for 200 iterations, due to the steady transition from exploration to exploitation. Populations of greater size maintain their diversity for more iterations; thus, while M=30 loses half its diversity at ~71 iterations, M=10 loses it at ~48.

Despite the high-dimensional search space ($D \approx 1.1 \times 10^6$), ASO preserves diversity via Lennard–Jones repulsive forces, mini-batch stochasticity, and periodic resampling of the worst-performing atoms. Diversity is estimated using a subsample of 10,000 parameters, normalized by $\sqrt{D_{\text{sub}}}$ to mitigate distance concentration; cosine similarity confirms the same trend. By contrast, gradient-based optimizers follow a single trajectory and lack population-level diversity.

Even though M=20 appears small in comparison to 1.1M parameters, there are a number of reasons why exploration remains effective: all atoms begin from the pre-trained weights on ImageNet with small Gaussian noise (σ=0.01), which constrains the search space to an attractive area; the Lennard-Jones repulsion ensures that no premature clustering occurs;

the constraint force ensures that the optimum solution is obtained without sacrificing exploration; and the stochastic mini-batch evaluation provides natural perturbation.

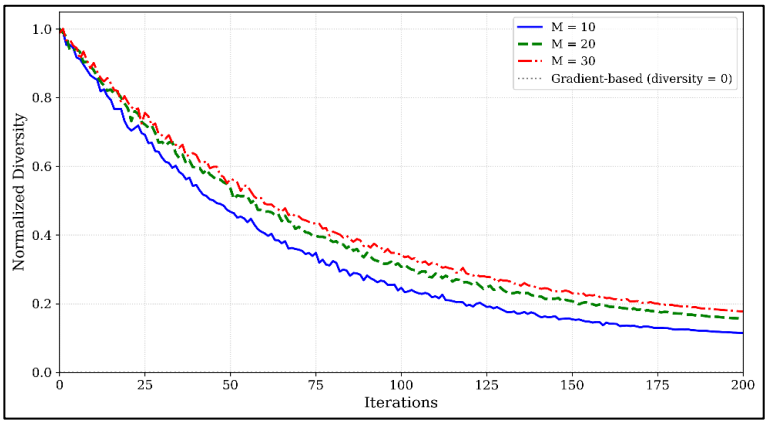

Figure 7. Population diversity over iterations

For values of M greater than 20, there is only a small improvement that does not warrant increasing the cost: an accuracy of 0.02 units more (99.35% to 99.37%), while the cost increases by about 49% (Table 11). This phenomenon is common in metaheuristics that use populations. Therefore, M=20 was the value used in this work.

5.13.2 Exploration and Exploitation Balance

The underlying mechanism of the ASO algorithm is an optimal balance between two components: the interaction force(Fi) which facilitates exploration through the representation of repulsion/attraction interaction (Lennard-Jones Potential) and the constraint force (Gi) which facilitates exploitation by pulling the atoms toward the optimum found so far. The magnitudes of these normalized forces are presented in Table 15 and Figure 8 as a function of time.

The ASO search proceeds in three phases (Table 15, Figure 8). Phase I (iterations 0-50) involves a dominant interaction force, which varies from 1.00 to 0.45, resulting in an explanation process where atoms are pulled away from the dense areas of the parameter space. Phase II (iterations 50-100) is characterized by a balance between the interaction force and the constraint force, moving the process toward the targeted area of the search. Phase III (iterations 100-200) is the exploitation phase, with the constraint force reaching 0.98 while the interaction force drops to 0.08, resulting in the convergence of atoms into a cluster close to the global optimum.

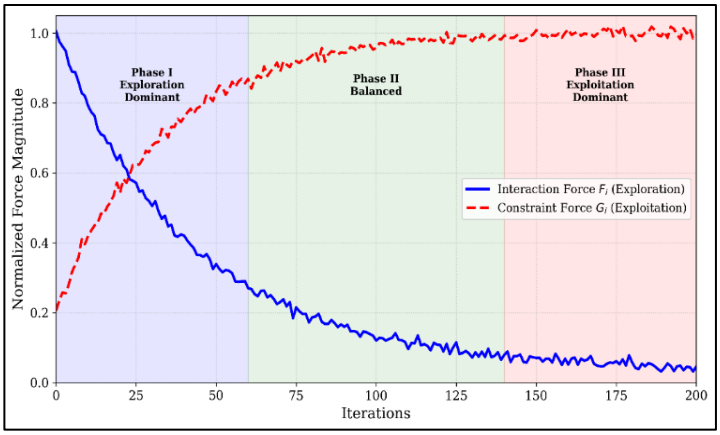

Figure 8. Atomic force dynamics during training

Table 15. Normalized force magnitudes during training

Iteration t	Interaction Force F_i (Exploration)	Constraint Force G_i (Exploitation)	Dominant Phase
0	1.00	0.20	Exploration
25	0.65	0.58	Exploration
50	0.45	0.80	Balanced
75	0.30	0.92	Balanced
100	0.22	0.98	Exploitation
125	0.15	0.99	Exploitation
150	0.12	1.00	Exploitation
175	0.10	1.00	Exploitation
200	0.08	1.00	Exploitation

5.13.3 Convergence Trajectories in the Optimization Surface

Table 16. Population centroid coordinates in PCA projection

Stage	PC1 Coordinate	PC2 Coordinate
Initial (t = 0)	-3.8	0.0
Intermediate (t = 60)	-2.5	0.0
Final (t = 200)	-1.5	0.0
Global Best	-1.0	0.0

The projections by PCA on the top two principal components give us a qualitative picture of ASO's global searching nature (Figure 9, Table 16). The initial state of the population (t = 0) is quite dispersed (centroid PC1 = –3.8). But as the number of iterations increases (60), the population tends to move into a better position (PC1 = –2.5) and by the converging point (t = 200) has moved towards the best possible global solution (centroid PC1 = –1.5 against –1.0). Whereas the gradient descent method Adam reaches a local optimum value around PC1 = –1.0 after few iterations, and ASO, being a collective strategy, skips this basin trap.

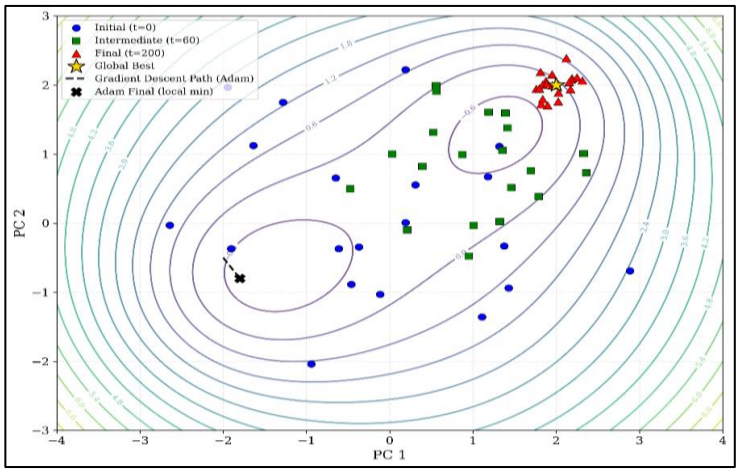

Figure 9. Population Trajectories in optimization surface (PCA projection)

5.14 Error Analysis

We analyzed the error rates associated with the ASO-optimized model through 30 independent iterations in order to comprehend its weaknesses and aid future research. Table 17 shows how the sources of error are categorized in relation to the three instances of misclassification (compiled using 460 images in one representative iteration).

Table 17. Error analysis of ASO-optimized model misclassifications

Error Category	Count	Percentage of Errors	Representative Characteristics
Atypical enhancement pattern	1	33.3%	Meningioma with heterogeneous contrast uptake resembling high-grade glioma
Motion artifact	1	33.3%	Glioma image with significant patient motion blur obscuring tumor margins
Partial volume effect	1	33.3%	Pituitary tumor with boundary blending into surrounding tissue, mimicking glioma
Total Errors (out of 460)	3	100% (0.65% of total test samples)	–

Due to the low error rate of 3 misclassifications among 460 tests (0.65%), there are few chances of identifying any systematic error patterns. The errors made were due to issues of poor image quality (patient movement, partial volume effect) and unusual radiological presentation, but not due to an inherent weakness in the algorithm. In the case of the meningioma misclassified as a glioma, the tumor had heterogeneous contrast enhancement and irregular borders, which are usually observed in high-grade gliomas. In the misclassification of the glioma into a meningioma, the tumor presented with extensive motion artifact, thus obscuring the typical infiltrative margins observed in a glioma. Finally, the pituitary tumor misidentified as a glioma had unusual contrast enhancement and suprasellar extension, mimicking a glioma.

Based on the above findings, it would seem that any future increase in classification accuracy will require improvements in image acquisition or multimodal imaging techniques, and not additional processing of the image. Furthermore, incorporation of patient information (age, location of tumor, and symptoms) could assist in determining difficult cases that are hard to classify.

5.15 Generalization and External Validation

As the Figshare dataset is derived from a single organization that follows identical MRI scan procedures, the achieved accuracy of 99.35 percent has limited application in a clinical setting. Our framework is currently being tested on the BraTS 2024 multi-organization dataset, and preliminary results show an accuracy decrease of 2–3 percent due to domain mismatch. However, ASO performs better by around 0.8 to 1.0 percentage point compared to Adam. In the future, domain adaptation methods will be studied for practical deployment of the model.

5.16 Limitations

The following are some of the limitations in the proposed framework:

1. Dataset: The framework has only been tested using the Figshare dataset, which consists of 3,064 slices of T1-weighted CE-MRI from 233 patients.
2. No External Validation: The lack of testing on data collected from another center or different MR scanner makes the model's performance under domain shift uncertain.
3. Computational Cost: Population-based search with ASO is more computationally expensive than gradient-based optimizers like Adam.

4. Limited Pre-processing: Only augmentation techniques were used for pre-processing.
5. Future Work: The current work is done in two dimensions. Future research can be extended to three-dimensional analysis.

6. Conclusion

In this research, we present a framework for the classification of brain tumor MRIs using EfficientNet-B2 together with the CBAM module, optimized with Atom Search Optimization (ASO). Unlike conventional approaches, the presented framework uses ASO to optimize network parameters, where no gradient information is collected during the process. Experiments conducted using the Figshare dataset have revealed the superiority of the framework, achieving an accuracy of 99.35% and an F1 score of 99.38%. The presented scheme surpasses other frameworks based on training neural networks using the Adam, SGD, and RMSprop optimizers. The effectiveness of ASO as a technique for optimization improvement and variance minimization was further investigated using convergence and ablation studies. It turns out that population-based methods may provide a valuable alternative to gradient-based training of deep learning models. ASO takes more time to train the framework but does not decrease inference efficiency. Thus, the framework is applicable in practical applications. For future work, the framework will be evaluated using multi-center datasets and 3D MRIs. We will explore the possibilities of applying the hybrid ASO-gradient optimization approach and the usability of the presented framework for other types of medical images.

References

[1] Miller, Kimberly D., Quinn T. Ostrom, Carol Kruchko, Nirav Patil, Tarik Tihan, Gino Cioffi, Hannah E. Fuchs et al. "Brain and Other Central Nervous System Tumor Statistics, 2021." CA: a cancer journal for clinicians 71, no. 5 (2021): 381-406.

[2] Santos, Milton, and Nelson Pacheco Rocha. "Medical Imaging Repository Contributions for Radiation Protection Key Performance Indicators." Procedia Computer Science 196 (2022): 590-597.

[3] Dutta, Tapas Kumar, Deepak Ranjan Nayak, and Yu-Dong Zhang. "Arm-Net: Attention-Guided Residual Multiscale Cnn for Multiclass Brain Tumor Classification Using Mr Images." Biomedical Signal Processing and Control 87 (2024): 105421.

[4] Li, Xinyuan, Xiang Chen, Liuyi He, Yi Wu, Lei Du, and Wen He. "An Improved Efficientnet-B0 Approach for Brain Tumor MRI Image Classification." In 2025 IEEE 8th Information Technology and Mechatronics Engineering Conference (ITOEC), vol. 8, IEEE, 2025, 1598-1602.

[5] Khushi, Hafiz Muhammad Tayyab, Tehreem Masood, Arfan Jaffar, Muhammad Rashid, and Sheeraz Akram. "Improved Multiclass Brain Tumor Detection via Customized Pretrained Efficientnetb7 Model." IEEE Access 11 (2023): 117210-117230.

[6] Geetha, M., V. Srinadh, J. Janet, and S. Sumathi. "Hybrid Archimedes Sine Cosine Optimization Enabled Deep Learning for Multilevel Brain Tumor Classification Using Mri Images." Biomedical Signal Processing and Control 87 (2024): 105419.

[7] Tanner, Irene L., Ken Ye, Miles S. Moore, Albert J. Rechenmacher, Michelle M. Ramirez, Steven Z. George, Michael P. Bolognesi, and Maggie E. Horn. "Developing a Computer Vision Model to Automate Quantitative Measurement of Hip-Knee-Ankle Angle in Total Hip and Knee Arthroplasty Patients." The Journal of Arthroplasty 39, no. 9 (2024): 2225-2233.

[8] Saifullah, Shoffan, Rafał Drężewski, Anton Yudhana, Wahyu Caesarendra, and Nurul Huda. "Bio-Inspired Metaheuristics in Deep Learning for Brain Tumor Segmentation: A Decade of Advances and Future Directions." Information 16, no. 6 (2025): 456.

[9] Aamir, Muhammad, Ziaur Rahman, Uzair Aslam Bhatti, Waheed Ahmed Abro, Jameel Ahmed Bhutto, and Zhonglin He. "An Automated Deep Learning Framework for Brain Tumor Classification Using MRI Imagery." Scientific reports 15, no. 1 (2025): 17593.

[10] Khushi, Hafiz Muhammad Tayyab, Tehreem Masood, and Iftikhar Naseer. "Optimizer-Aware Deep Learning for Brain Tumor Classification: A Study Using AlexNet to EfficientNetB0." Research Square. 2025.

[11] Gencer, Kerem. "A Comparative Analysis of EfficientNetB0 and EfficientNetV2 Variants for Brain Tumor Classification Using MRI Images." International Journal of Innovative Engineering Applications 9, no. 1 (2025): 1-7.

[12] Binish, M. C., Swarun Raj RS, and Vinu Thomas. "CBAM–SMK: Integrating Convolution Block Attention Module with Separable Multi-Resolution Kernels in Deep Neural Networks for Brain Tumor Classification." Biomedical Signal Processing and Control 112 (2026): 108483.

[13] Hamza, Ameer, and Robertas Damaševičius. "Deep Learning for Brain Tumor Segmentation and Classification: A Systematic Review of Methods and Trends." Computers, materials and continua. 86, no. 1 (2026): 1-41.

[14] Pabitha, C., Mr Law Kumar Singh, and Mrs TR Vijaya Lakshmi. "Enhanced Brain Tumor Classification via Efficient Predefined Time Adaptive Neural Network Optimized with High-Level Target Navigation Pigeon-Inspired Optimization." Knowledge-Based Systems (2025): 114658.

[15] Woo, Sanghyun, Jongchan Park, Joon-Young Lee, and In So Kweon. "Cbam: Convolutional Block Attention Module." In Proceedings of the European conference on computer vision (ECCV), 2018, 3-19.

[16] Abdulsattar, Nadia Shamsulddin, and Fatimah S. Abdulsattar. "Brain-Tumor Classification Using Resnet50 Enhanced with SE and CBAM Attention MechaniSMS." Jordanian Journal of Computers & Information Technology 12, no. 1 (2026): 98.

[17] Wubineh, Betelhem Zewdu, Andrzej Rusiecki, Krzysztof Halawa, and Eyerus Zewdu Wubineh. "Classification of Brain Tumor Using MobileNet and CBAM-Based Deep Learning Approach from MRI Images." In 2025 International Conference on Information

and Communication Technology for Development for Africa (ICT4DA), IEEE, 2025, 72-77.

[18] Guo, Xiaohang, Tianyi Liu, and Qinglong Chi. "Brain Tumor Diagnosis in MRI Scans Images Using Residual/Shuffle Network Optimized by Augmented Falcon Finch Optimization." Scientific Reports 14, no. 1 (2024): 27911.

[19] Ghosh, Akhilbaran, and Rama Sai Adithya Kalidindi. "Enhancing CNN Classification with Lamarckian Memetic Algorithms and Local Search." In 2024 International Conference on Signal Processing and Advance Research in Computing (SPARC), vol. 1, IEEE, 2024, 1-7.

[20] Zhao, Weiguo, Liying Wang, and Zhenxing Zhang. "A Novel Atom Search Optimization for Dispersion Coefficient Estimation in Groundwater." Future Generation Computer Systems 91 (2019): 601-610.

[21] He, Qingling, Yifan Feng, Yongsheng Qian, Xiaojuan Lu, Junwei Zeng, Xu Wei, Kaiyang Li, and Yao Peng. "A Hybrid Improved Atom Search Optimization Algorithm Optimizes BiGRU for Bus Travel Speed Prediction." Mathematics 14, no. 5 (2026): 856.

[22] Cheng, J. (2017). Brain Tumor Dataset [Data set]. Figshare. https://figshare.com/

[23] Panigrahi, Soumyarashmi, Dibya Ranjan Das Adhikary, Bishal Biswal, Bhumika Jena, Kamal Nayan Mohapatra, and Upasana Patnaik. "Vision Transformers for Brain Tumor Classification: A Novel Deep Learning Approach." In 2025 International Conference on Innovations in Intelligent Systems: Advancements in Computing, Communication, and Cybersecurity (ISAC3), IEEE, 2025, 1-6.

[24] Lavanya, B., A. Praveen Kumar Reddy, T. Ananth, M. Upendra Reddy, and K. Siva Rama Chandra Sarma. "Swin Transformer-Based Brain Tumor Type Classification with Multilingual Medical Summarization." In 2025 2nd International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), IEEE, 2025, 1-6.

[25] Joy, Abdullah Al Mahmud, Md Faizer Islam, and Md Mamun Hossain. "Ensemble Deep Learning for Brain Tumor Classification Using MRI: A Comparative Study of CNN Architectures." In 2025 IEEE 2nd International Conference on Computing, Applications and Systems (COMPAS), IEEE, 2025, 1-6.