Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Predicting Brain Tumours Through Transfer Learning and Optimized Hyperparameters Using the Modified Whale's Algorithm

**Sivani Pinnaboina**
KLEF University
**Venkata Sowmya Kambhampati**
KLEF University
**Kodanda Rama Sastry Jammalamadaka**
drsastry@kluniversity.in

KLEF University
**Sasi Bhanu Jammalamadaka**
Gokaraju and Gangaraju Institute of Engineering and Technolgy

**Article**

**Keywords:** Transfer Learning, Pre Training, ResNet18, Whales optimization, Brain Tumors

**Posted Date:** April 23rd, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-9365075/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

**Version of Record:** A version of this preprint was published at Scientific Reports on October 6th, 2026.
See the published version at https://doi.org/10.1038/s41598-026-67395-z.

# Predicting Brain Tumours Through Transfer Learning and Optimized Hyperparameters Using the Modified Whale's Algorithm

Sivani Pinnaboina[1], Venkata Sowmya Kambhampati[2†], Kodanda Rama Sastry Jammalamadaka[3*†], Sasi Bhanu Jammalamadaka[4†]

[1*]Department, Organization, Street, City, 100190, State, Country.
[1*]Electronics and Communication, KLEF University, Green Fields, Vaddeswaram, 522502, Andhra Pradesh, India.
[2*]Electronics and Communication, KLEF University, Green Fields, Vaddeswaram, 522502, Andhra Pradesh, India.
[3*]Computer Science and Engineering, KLEF University, Green Fields, Vaddeswaram, 522502, Andhra Pradesh, India.
[4*]Computer Science and Engineering, Gokaraju GangaRaju Institute of Engineering and Technology, Bachupalli, Hyderabad, 500118, Telangana, India.

*Corresponding author(s). E-mail(s): drsastry@kluniversity.in;
Contributing authors: psivani@kluniversity.in; sowmyakambhampati@kluniversity.in; bhanukamesh1.com@gmail.com;
†These authors contributed equally to this work.

**Abstract**

Brain tumours cause severe morbidity. Needs to be diagnosed in the minimum time and with the Highest accuracy. It takes a long time to learn Deep learning Models that can predict the type and severity of brain tumours. Using values for some selected Hyperparameters often leads to non-convergence, and accuracy cannot be guaranteed. Full-domain-focused adaptive Transfer Learning using models such as ResNet18 reduces learning time and archives 100% accuracy. The learning time can be reduced while maintaining 100% accuracy by selecting appropriate, relevant, and ranked hyperparameters and their values using the Modified Whales algorithm, which focuses on the specified search space. The

proposed method reduced learning and prediction time by 66% (from 3 to 1 second) while achieving 100% accuracy, exceeding the accuracy of other methods reported in the literature.

**Keywords:** Transfer Learning; Pre Training; ResNet18; Whales optimization, Brain Tumors

# 1 Introduction

Brain tumours represent one of the most life-threatening neurological disorders, characterised by the uncontrolled proliferation of abnormal brain cells that can significantly impair cognitive and motor functions if left untreated. Brain tumours are among the most critical and life-threatening neurological disorders, causing severe mortality and morbidity worldwide. They arise from abnormal proliferation of brain cells, which can be either malignant or benign, but in either case may lead to severe neurological impairments if not diagnosed early [1]. Regardless of type, brain tumours can cause life-threatening complications, making early detection and accurate classification crucial for patient survival. The three most common brain tumour types include gliomas, meningiomas, and pituitary tumours, each exhibiting distinct biological and pathological characteristics [2].

The accurate diagnosis and classification of brain tumours are vital for determining appropriate treatment strategies and improving patient survival rates. Early and accurate detection of these tumours is pivotal in determining treatment strategies and improving patient survival rates. CNN-based models have been successfully applied to brain tumour classification, achieving higher accuracy than conventional approaches. CNN-based approaches have achieved impressive accuracy in tumour classification tasks, including multi-grade glioma detection and three-class classification of glioma, meningioma, and pituitary tumours. Data augmentation applied to improve the quality of tumour-related images increases prediction accuracy [3]

Magnetic Resonance Imaging (MRI) has emerged as the most effective imaging modality for brain tumour diagnosis, offering superior spatial resolution and soft-tissue contrast compared to computed tomography (CT) [4] . However, manual interpretation of MRI scans is time-consuming, subject to observer variability, and prone to misdiagnosis, especially in complex cases [5]. Traditional diagnosis largely depends on radiologists' expertise in interpreting medical imaging, particularly magnetic resonance imaging (MRI).

Computer-aided diagnosis (CAD) systems have been developed to assist radiologists by employing automated image segmentation, feature extraction, and classification techniques [6]. These systems reduce the dependency on human expertise and enhance diagnostic consistency. Despite their advantages, existing CAD systems encounter challenges such as varying tumour sizes, irregular shapes, and intensity inhomogeneities, which complicate accurate segmentation and classification.

Moreover, traditional machine learning methods such as k-nearest neighbour (KNN), support vector machines (SVM), and artificial neural networks (ANN) have

achieved moderate success but often struggle with high-dimensional datasets and non-linear feature distributions [7]. Their performance often suffers from overfitting, high dimensionality, and limited generalization capabilities.

Recent advances in deep learning, particularly convolutional neural networks (CNNs), have revolutionized medical image analysis by enabling automatic feature extraction directly from raw images. Convolutional Neural Networks (CNNs) revolutionised medical image analysis by enabling end-to-end learning of hierarchical features directly from raw images [8], [9]. CNN-based methods have achieved promising accuracy in distinguishing tumour types such as gliomas, meningiomas, and pituitary tumours, significantly outperforming classical techniques [10].

Nonetheless, these models demand large training datasets and computational resources, which can limit their practical applicability [11]. To further enhance performance, optimization algorithms have been incorporated into deep learning frameworks for fine-tuning network parameters 12].

Despite these advancements, the performance of CNNs largely depends on the choice of network architecture and hyperparameters. Manual tuning of these hyperparameters—such as learning rate, dropout rate, activation functions, and batch size—remains laborious and may not yield optimal results [13].

To overcome this, researchers have proposed advanced optimisation strategies, such as systematic hyperparameter tuning. The Generalised CNN model, combined with hyperparameter tuning, increases effectiveness and accuracy. This highlights the importance of fine-tuning architectural parameters, optimizers, and dropout strategies to balance accuracy and computational efficiency. The time required to complete training when data sets are huge is enormous [14] and hyperparameter selection via Bayesian optimisation [15].

There are pretrained architectures for processing the images, which include VGG16,[16] ResNet50[17], and InceptionV[18]. These models have been trained using many large datasets. These models, when retrained using transfer learning with tumour-related images, achieve poor accuracy.

Collectively, CNNs enhanced by hyperparameter optimisation and methods represent a promising direction for robust brain tumour classification. They address the shortcomings of traditional CAD systems by providing automated, accurate, and scalable diagnostic tools.

Ongoing research into these hybrid frameworks aims to bridge the gap between experimental studies and clinical applications, ultimately improving patient care and decision-making in neuro-oncology.

Training is expensive and time-consuming when the dataset is large. When hyperparameters are selected randomly, there is no guarantee of convergence, and the model's accuracy suffers. Generalised pre-trained models such as U-Net, ResNet, VGGNet, and InceptionV, when retrained using transfer learning, drastically reduce training time. The Learning added with optimised hyperparameters provides the most accurate models that can be trained in the minimum time.

Selecting appropriate hyperparameters, optimising their values, and using them during adaptive domain-specific learning will drastically reduce training and prediction

times. Many hyperparameters can be used. There should be a way to select the most appropriate hyperparameters scientifically rather than randomly.

Whales' Algorithm is robust and can therefore be effectively employed to find optimal solutions in complex environments. Whales' algorithm, if formulated to find solutions with varying space for each hyperparameter, would provide fast-converging solutions.

In this paper, domain-specific adaptive transfer learning is achieved using RESNET18, with hyperparameters identified using the Modified Whales algorithm, to effectively classify brain tumours in the minimum time with the highest accuracy. A comparative analysis of ResNet-18 with other kinds of models is shown in Table 1. ResNet-18 is fast to Train, requires minimal memory, and achieves moderate accuracy when pre-trained on ImageNet.

**Table 1** Comparison of ResNet18 Models with Other Pre-Trained Models

| Model | Accuracy (ImageNet) | FLOPs (224×224) | Parameters | Memory usage |
|---|---|---|---|---|
| ResNet-18 | 69.8% | 1.8 GFLOPs | 11.7M | Medium |
| VGG-16 | 71.5% | 15.5 GFLOPs | 138M | Very High |
| AlexNet | 57.0% | 0.7 GFLOPs | 60M | High |
| Inception-V3 | 78.0% | 5.7 GFLOPs | 23M | Medium |
| MobileNet-V2 | 68.0% | 0.3 GFLOPs | 3.4M | Very Low |

## 2 Related work

Various CNN architectures were developed for tumour classification. Sultan et al. [19], Abiwinanda et al. [20], and Sajjad et al. [21] demonstrated that CNNs could successfully classify gliomas, meningiomas, and pituitary tumours. The investigation yielded vital revelations, yet they often required substantial computational power, which limited their practical deployment. The advent of deeper CNN architectures and the implementation of transfer learning have substantially improved classification performance. Sajjad et al. introduced a multi-grade classification framework that integrated tumour segmentation, data augmentation, and fine-tuning of the VGG19 architecture, achieving an accuracy of 96.56%.

M. Hanif et al., [22] introduced a strategy grounded in Convolutional Neural Networks (CNNs) to distinguish between benign and malignant tumours. Their research outcomes underscored that the use of pretrained models such as Xception and DenseNet can significantly enhance feature extraction and elevate CNN performance metrics.

They have applied CNNs to distinguish brain tumors into three specific classifications—glioma, meningioma, and pituitary. This study underscored the potential of CNNs in brain tumour research; however, it was constrained by limitations in dataset size and generalisation.

X Jio et al. [23] likewise employed CNN architectures to classify glioblastoma tumours, asserting that deep learning methodologies are reliable for identifying tumour

types. In a related context, O. Özkaraca et al. [24] investigated the classification of meningioma, glioma, and pituitary tumors. This analysis encompassed tumor conditions and grading, utilizing a comprehensive MRI dataset to emphasize the increasingly critical role of CNNs in multi-class classification endeavors.

The idea behind transfer learning has been thoroughly examined. R. Chelghoum. [25] employed pretrained CNN models, such as VGG16 and VGG19, to classify brain tumours. The use of transfer learning enabled shorter training time and better feature representation, though more tweaks were needed to ensure reliable outcomes. Ali Çınar et al. [26] employed a ResNet50-based transfer learning strategy, modifying its terminal layers to achieve an accuracy of 97.2% .

S. Deepak et al., [27] used GoogLeNet, pre-trained on ImageNet, as a model for their tumour classification efforts. Z. A. Swathi et al., [28] undertook block-wise fine-tuning of deep CNNs tailored for contrast-enhanced MRI, revealing that transfer learning could ease training requirements, though it could also sometimes lead to overfitting.

Sinno Jialin Pan et al. advanced this domain further by architecting a deep neural network for multi-class classification, achieving accuracies of 96.13% for tumour type classification and 98.70% for Glioma grading [29].

Despite notable advancements in CNN-based models, numerous challenges persist. The core problem stems from the shortage of detailed, richly annotated medical datasets, which has led to the development of deep learning approaches. The rise of transfer learning as a favoured method to address this issue also introduces the risk of adverse transfer when the source and target domains are inadequately aligned. Notwithstanding the advancements enabled by CNN-based methodologies in the identification of brain tumours, they often rely on extensive hyperparameter optimisation. In response to this predicament, optimization solutions have been investigated with increasing regularity. D. Rammurthy [30] introduced Whale Harris Hawks Optimisation (WHHO) in conjunction with CNNs, demonstrating that bio-inspired optimisation approaches can enhance both efficiency and reliability.

Hyperparameter tuning has not been thoroughly examined in brain tumour classification research. Methodologies such as Bayesian Optimisation show significant promise for automating parameter tuning, thereby further enhancing the efficacy of CNNs [31].

Hyperparameter tuning delivers significant advantages by sharpening the performance of deep learning models for network intrusion detection. Hyperparameters dictate fundamental elements such as learning rate, batch size, and network depth, which are essential to a learning model's functioning. Proper selection of the hyperparameters is crucial for the training workflow, convergence stability, and final detection precision [32]. Opting for inaccurate hyperparameters may result in underwhelming model performance, overfitting to the training dataset, or hindering convergence. Whale Optimization algorithm helps optimize the training of deep CNNs for brain tumor detection. When applied to MRI images, the WOA-based CNN leverages statistical and texture features, including tumour size, mean, variance, kurtosis, and Local Optimal Oriented Pattern (LOOP), to achieve robust classification. Manual hyperparameter selection is both labour-intensive and inefficient, particularly given the extensive parameter space and the inherent variability of network traffic datasets [33].

Desta et al. implemented the Hyperband method, which iteratively allocates resources to promising configurations based on validation loss, thereby facilitating the selection of CNN models [34]

M. Tawfik et al. 35] employed lion swarm optimization, a bio-inspired metaheuristic framework, to calibrate parameters of their OCNN-HMLSTM model, achieving commendable detection accuracy.

The research conducted by Chen and associates illustrates that the use of Neuroevolution alongside a multi-objective evolutionary algorithm (MOEA/D) to autonomously craft CNN architectures has been presented [36]. They have proposed the Poor and Rich Optimization Algorithm (PROA), which emulates wealth distribution behaviour, and applied this technique to optimize CNN-ALSTM hyperparameters, thereby enhancing detection performance

B. Bischl et al. [37] and A. Morales-Hernández, et al. [38] employed Bayesian Optimisation in conjunction with the Tree-structured Parzen Estimator (TPE), which intelligently samples the search space to identify near-optimal configurations.

A. Kaissar et al. [39], P. Geurts [40] reported that Particle Swarm Optimisation (PSO) can efficiently and rapidly determine optimal hyperparameter sets tailored for CNN-based IDS architectures. Selecting the most relevant hyperparameters and their optimal values has proven to be one of the most difficult challenges in deep learning. Unlike model parameters, which are learned during training, hyperparameters must be set before training begins, and their influence on convergence, accuracy, and generalisation can be profound.

Poor hyperparameter choices often lead to overfitting, vanishing gradients, slow training, or underperformance [41]. Traditional approaches, such as grid and random search, provide brute-force solutions but are computationally expensive, particularly for large-scale CNN architectures and high-dimensional search spaces [42]

Most hyperparameter-optimisation techniques treat all hyperparameters equally during the search, which is rarely the case. Some hyperparameters—such as the learning rate or optimizer choice—tend to exert a much stronger influence on model performance than others, such as batch size or weight initialization. Ignoring this relative importance results in wasted computational resources and inefficiency in hyperparameter optimisation pipelines [43]. However, selecting specific hyperparameters from a large set that improve the model's accuracy is key [44].

Metaheuristic algorithms inspired by natural phenomena, such as the Whale Optimisation Algorithm (WOA), have demonstrated strong global search capabilities for solving complex optimisation problems [45]. Evolutionary algorithms, such as Genetic Algorithms (GA) and Particle Swarm Optimisation (PSO), have also been used to optimize hyperparameters for training CNN-based predictive models. [46].

A comprehensive survey [47] of HPO techniques, covering grid search, random search, Bayesian optimisation, evolutionary algorithms, and hybrid methods, has been presented. They concluded that Bayesian methods and metaheuristic approaches strike a balance between computational cost and optimization efficiency, making them suitable for complex deep learning tasks in domains such as medical imaging.

Across all the Training models, class imbalance and Interpretability are the two primary issues to address. In response to these challenges, M. N. Ge et al. [48] proposed

a custom CNN architecture specifically designed to enhance brain tumour detection while addressing interpretability and class imbalance. Their approach introduced oversampling and class weighting to mitigate class imbalance, ensuring balanced model training.

In addition, interpretability was enhanced by gradient-based attribution methods and saliency maps, which highlight the most influential regions of the image that contribute to predictions. This combination provided a transparent, clinician-friendly diagnostic framework.

ANN models have been built with different Hyperparameters and their configurations fed as an Example set, and the model's accuracy is assessed. A fixed Parameter set that includes batch size, learning rate, and number of epochs has been considered. An optimal configuration that provides the highest accuracy has been found [49].

Bio-inspired approaches, such as swarm intelligence and evolutionary algorithms, mimic natural behaviours to optimise network parameters and feature selection, thereby enhancing model efficiency [50]. In particular, the Salp Swarm Algorithm (SSA), inspired by the swarming behaviour of salps in oceans, has emerged as a robust method for global optimisation. SSA can fine-tune hyperparameters such as learning rate, batch size, and network depth, leading to faster convergence and improved classification performance. Hybrid bio-inspired models have been used to integrate SSA with Kernel Extreme Learning Machine (KELM) to optimize CNN hyperparameters for brain tumour classification [51].

Differential Evolution (DE), a population-based metaheuristic inspired by natural selection, offers a promising alternative. DE maintains population diversity through its mutation and crossover mechanisms, making it particularly effective in navigating complex, multimodal optimization landscapes [52]. Four key parameters have been fine-tuned using DE: learning rate, dropout, Batch size, and Number of filters.

# 3 Research Gap and Objectives

## 3.1 Research gap

Many have used a pretrained ResNet18-like model, which leads to poor accuracy, and have taken a long time to train when using transfer learning with domain data such as Tumour data. None has attempted to develop a pre-trained ResNet-18-like model trained on tumour data from multiple worldwide datasets that runs faster and is 100

Non have used a method for selecting the most specific hyperparameters and their values using optimised algorithms such as WoA, and then used the same method for training pre-trained models, which can subsequently be used for transfer learning. The woA algorithm uses random values for each hyperparameter, whereas hyperparameters in deep learning have associated parameter spaces that need to be considered. Using WoA significantly reduces learning time.

To address these, a Range-Specific WoA algorithm applied to a domain-specific pre-training and transfer learning method has been proposed, yielding 100% accuracy and reducing training and Prediction time to 1 sec.

### 3.2 Research Obectives

1. To develop domain specific Resnet18 Model that can be adaptively and incrementally trained such that 100
2. To modify WoA Algorithm that takes a range of values for each parameter separately and determines optimum values for each of the parameters, and then use it to retrain the pre-trained domain specific resnet18 Model, that help achieving 100

## 4 Methodology

The methodology used in this research is shown in Figure 1.

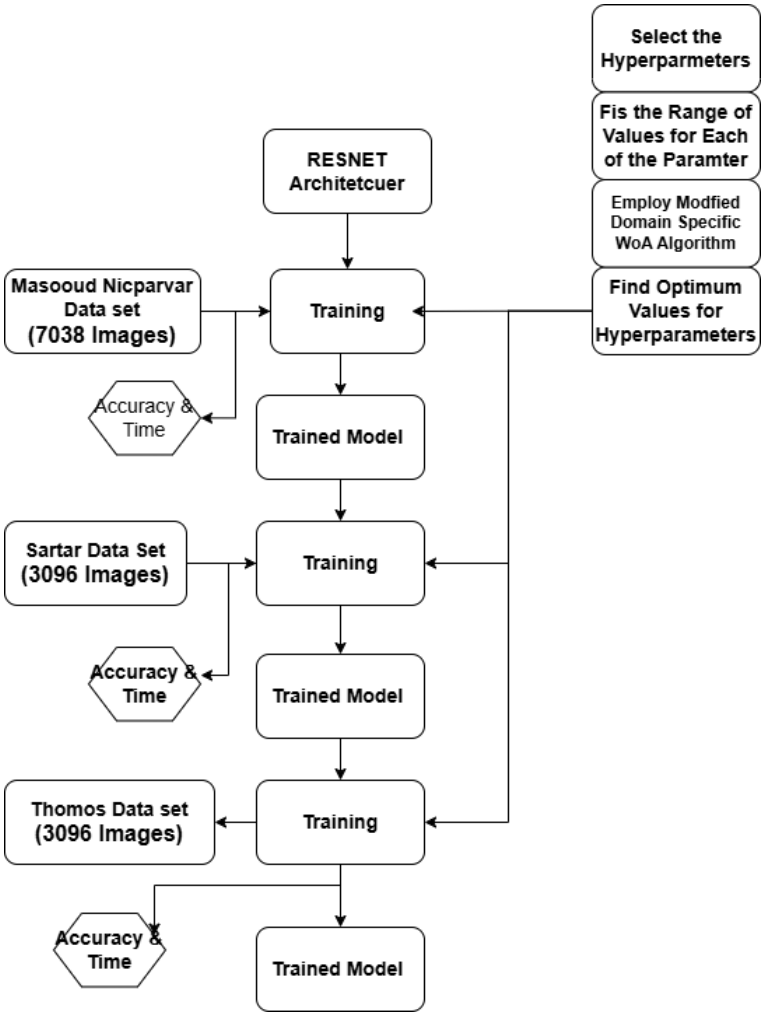



**Fig. 1** Research Methodology

The Resnet Architecture is used to train iteratively using different data sets that include Masooud Nicparvar Data set (7038 Images), Sartar Data Set, (3096 Images). At each iteration, a single data set is used, and the ResNet18 model is trained. Training is performed using optimised Hyperparameter Values fed as parameters. The optimised hyperparameter values are obtained by implementing the modified whale optimisation algorithm, which considers different ranges for each parameter. After completing the training, which considers different ranges of values for different hyperparameters, the response time and accuracy percentage are recorded. The model parameters are stored as checkpoints. The checkpoint is restored, and model training is performed on a different data set, using the same optimised value obtained using the modified whale optimisation algorithm. The process is repeated, considering all available worldwide datasets relating to brain tumours. The final model obtained is a fully pre-trained ResNet-18 model trained on the Brain Tumour datasets. The pretrained ResNet18 Model is used by the healthcare organisation to implement transfer learning using the few images that they have. The re-training and prediction is done in less than 2 seconds of time.

# 5 Material and Methods

## 5.1 Whales Optimisation Emperical Formulation

The Whale Optimisation Algorithm (WOA) is a nature-inspired metaheuristic optimisation algorithm developed by Seyedali Mirjalili in 2016. It mimics the social hunting behaviour of humpback whales, particularly the bubble-net feeding strategy. WOA is widely used in machine learning, engineering design, and feature selection due to its simplicity, exploration-exploitation balance, and competitive performance. Humpback whales feed by creating spiral bubbles to encircle prey (schools of krill or small fish). This technique is known as bubble-net feeding, and WOA abstracts this behaviour into three phases, which include Encircling prey (approximating optimal solution), Bubble-net attacking method (exploitation) and Search for prey (exploration).

**Encircling Prey**

Humpback whales can locate prey and surround it. In WOA, the best solution found so far is assumed to be the target prey or the optimal solution. Other candidate solutions then update their positions based on the best solution.

**Bubble-Net Attacking**

The attacking Mechanism involves shrinking the encircling mechanism

1. Shrinking Encircling Mechanism: As the whales get closer to their prey, they shrink the encircling mechanism by reducing the distance between their current position and the prey.
2. Spiral Updating Position: Whales swim around the prey in a spiral path. This is mathematically modelled by a spiral equation that combines the distance and angle of the whale's position relative to the prey

**Search for Prey**

Whales search for prey randomly, exploring different areas of the search space. This is like the exploration phase in other optimisation algorithms, where the algorithm seeks new solutions by moving away from the current best.

**Empirical formulation Encircling Prey (Approximating Optimal Solution)**

The encircling Behaviour is exhibited through Equation 1.

$$D = |CX^*(t) - X(t)| \tag{1}$$

where

1. X*(t) is the position vector of the best solution found so far
2. X(t) is the current position vector
3. A and C are coefficient vectors that help control the behaviour of the algorithm. C is the coefficient that randomly perturbs the distance
4. D is the vector that shows the distance between the current whale and the best whale (Pray)

Coefficient Vector A is computed using equation 2

$$A = a(2r - 1) \tag{2}$$

where

The Range of A is expressed as per equation 3

$$A \in [-a, a] \tag{3}$$

When
— | $A$ | ¡1 — Exploitation (move toward best solution) —
— | $A$ | ¿ 1— Exploration (search globally) —

When Parameter a decrease linearly from $2 \rightarrow 0$ during iterations the same is expressed as per euation 4..

$$a = 2 - 2t/T \tag{4}$$

where

1. t= current iteration
2. T= maximum iterations

The Purpose is to search a large search area in the Early Iterations, which leads to exploration and in the later iterations, fine-tuning near the optimum (Exploration).

Random Vector r is computed as per equation 5.

$$r \in [0, 1] \tag{5}$$

where

1. Random number generated for each whale and each dimension.
2. Used to add stochastic behavior to the search

The Variable C is computed as per equation 6

$$C = 2 * r \tag{6}$$

The Coefficient C is computed first, and then the distance D, then A is computed, and the whale's position is updated

**Empirical formulation for Bubble-Net Attacking (Exploitation Phase) is shown in Equation 7**

**Shrinking**

$$(t + 1) = X^*(t) - A * D \tag{7}$$

Where: —A— ¡ 1

**Spiral Updating can be done using Equation 8**

$$X(t + 1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \tag{8}$$

Where

1. X(t+1) = New Whale Position
2. X* (t) Best Solution found so far
3. D' Distence Between Whale and Best Solution where
4. $D' = [X^*(t) - X(t)]$
5. B - Constant Defining Spiral Shape
6. I - Random Number in [-1, 1]
7. t - Current Iterations

**Exploration (Searching for Prey)**

When — A— ¿ 1, Whales search randomly around another whale and around the pray. The distence D and the new location of the whale are computed using the equations 9 and 10

$$D = [C * X_(rand-)X(t)] \tag{9}$$

$$X(t + 1) = X_(rand) - A * D \tag{10}$$

## 5.2 Modified whales optimisation method

In the case of deep learning, several hyper parameters exists, and the value of each hyperparameter has to be chosen from a Range of values. One has to find optimum values considering all the hyperparameters so that when those values are chosen highest accuracy of CNN model can be achieved, and also the learning is done in minimum time. Suppose Each Whale Variable has limits

$$X_{\min} \leq X_i \leq X_{\text{Max}} \tag{11}$$

Where each X will have (x1, x2, x3, ... xd) where d stands for number of dimensions

The distance computation, the new location computation equations, and the spiral equation remain the same and are shown again in equations 12, 13, and 14.

$$D = |CX * (t) - X(t)| \tag{12}$$

$$X(t+1) = X^*(t) - A * D \tag{13}$$

$$X(t+1) = D' * e^{bl} * \cos(2\pi l) + X^*(t) \tag{14}$$

Now the boundary conditions are applied, which are called as clipping and projection as shoen in Equation 15.

$$X_i(t+1) = \begin{cases} X_{\min} & \text{if } X_i < X_{\min} \\ X_{\max} & \text{if } X_i > X_{\max} \\ X_i & \text{otherwise} \end{cases} \tag{15}$$

Every whale when has a separate range to search, then define max and min values for each Variable

$$X_min^k, X_max^k \tag{16}$$

The new distance and location areocation are computed by using equations 17 and 18

$$X_i^k(t+1) = min(max(X_i^k(t+1), X_(min,)^k), X_imax^k) \tag{17}$$

$$D = [CX^(*)(t) - X(t)] * ((X_(max) - X_Min)/R \tag{18}$$

Where R defines the predefined search space and the initialization of the whales can be affected through equation 20.

$$X_(i) = r(X_max - X_min) where r belongsto [0, 1] \tag{19}$$

# 6 The Modified Whales Algorithms based on Modified Whales empirical formulation

**Algorithm 1** Modified Whale Optimisation Algorithm (MWOA)

**Require:** Search space bounds $LB$, $UB$, population size $N$, maximum iterations $T$

**Ensure:** Best solution $X^*$

Initialize whale population $X_i$ $(i = 1, 2, \ldots, N)$ within $[LB, UB]$
Evaluate fitness of each whale
Identify best whale $X^*$
$t \leftarrow 1$
**while** $t \leq T$ **do**
  $a = 2 - \frac{2t}{T}$
  **for** each whale $i = 1$ to $N$ **do**
    Generate $r_1, r_2 \in [0, 1]$
    $A = 2ar_1 - a$
    $C = 2r_2$
    Generate $p \in [0, 1]$
    **if** $p < 0.5$ **then**
      **if** $|A| < 1$ **then**
        $D = |C \cdot X^* - X_i|$
        $X_i^{new} = X^* - A \cdot D$
      **else**
        Select random whale $X_{rand}$
        $D = |C \cdot X_{rand} - X_i|$
        $X_i^{new} = X_{rand} - A \cdot D$
      **end if**
    **else**
      $D' = |X^* - X_i|$
      Generate $l \in [-1, 1]$
      $b =$ constant
      $X_i^{new} = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*$
    **end if**
    **if** $X_i^{new} < LB$ **then**
      $X_i^{new} = LB$
    **else if** $X_i^{new} > UB$ **then**
      $X_i^{new} = UB$
    **end if**
    $X_i = X_i^{new}$
  **end for**
  Evaluate fitness
  Update $X^*$
  $t \leftarrow t + 1$
**end while**
**return** $X^*$

# 7 Data Sets

Many have collected brain tumour data. Their sizes, types, and colours vary. Table 3 displays Keras data. The images are categorised as glioma, meningioma, pituitary, or no tumour. Training and testing photos are provided separately. Globally, many data sets must be collected, converted, and processed to create a fully trained (pre-trained) model.

**Table 2** Details of the Brain tumour-related data sets

| Srl. Num. | Data Set Name | Total Images | Description | Image Size | Format | Colour |
|---|---|---|---|---|---|---|
| 1 | Masoud Nickparvar | 7022 | Classification into 4 Classes | 512 X 512 Pixels | Jpeg | Gray Scale |
| 2 | Navneel Data Set | 253 | Binary Classification | 224 X 225 Pixels | Jpeg | Gray Scale |
| 3 | Sartar Data set | 3096 | Classification into 4 Classes | 398 X 387 Pixels | Jpeg | Gray Scale |
| 4 | Brats Data set | 319 | Image segmentation | 240 X 240 Pixels | Jpeg | Gray Scale |
| 5 | Thomos Data set | 3096 | Same dataset as 3 | 398 X 397 Pixels | Jpeg | Gray Scale |
| 6 | Jakesh Bohaju | 3764 | Binary Classification | 259 X 269 Pixels | Jpeg | Gray Scale |

# 8 Training Algorithms

Three algorithms have been developed, that include the following:

1. Retraining with Brain tumour data using the pretrained RESNet18 model with the ImageNet data set
2. Initial pre-training using the first brain tumour data using the Resnet18 Architecture
3. Subsequent Training using different brain tumour-related data sets, using successively generated model parameters

## 8.1 Algorithm-A: Transfer learning using the Pre-Trained Resnet18 model on ImageNet

**Require:** Pre-trained ResNet18 parameters, Parvar Brain Tumour Dataset

**Ensure:** Classification Accuracy, Processing Time

1: **Dataset Preparation:**

Download dataset and split into training and testing sets
**Training Data Transformation:**
Resize, apply horizontal flip, random rotation
Convert to tensors and normalize
**Testing Data Transformation:**
Resize, convert to tensors, normalize
**Define Hyperparameter Search Space:**
$lr \in [10^{-5}, 10^{-1}]$, $bs \in [16, 512]$
$ep \in [10, 200]$, $pt \in [5, 20]$, $gc \in [0.1, 5]$
**Initialize Whale Population:**
**for** each whale $X_i$ **do**
    $X_i = [lr, bs, ep, pt, gc]$
**end for**
**Evaluate Fitness:**
**for** each whale $X_i$ **do**
    Train model for few iterations
    Compute validation accuracy
    $fitness_i = 1 - accuracy$
**end for**
Identify best whale $X^*$
**for** $t = 1$ to Max_Iterations **do**
    $a = 2 - \frac{2t}{Max_Iterations}$
    **for** each whale $X_i$ **do**
        Generate $r_1, r_2 \in [0, 1]$
        $A = 2ar_1 - a$
        $C = 2r_2$
        Generate $p \in [0, 1]$
        **if** $p < 0.5$ **then**
            **if** $|A| < 1$ **then**
                $D = |C \cdot X^* - X_i|$
                $X_i^{new} = X^* - A \cdot D$
            **else**
                Select random whale $X_{rand}$
                $D = |C \cdot X_{rand} - X_i|$
                $X_i^{new} = X_{rand} - A \cdot D$
            **end if**
        **else**
            $D' = |X^* - X_i|$, $l \in [-1, 1]$
            $X_i^{new} = D' e^{bl} \cos(2\pi l) + X^*$
        **end if**
        Apply boundary constraints
        Update $X_i = X_i^{new}$
    **end for**
    Evaluate fitness and update $X^*$
**end for**

Extract optimal hyperparameters
**Dataset Batching:**
Shuffle, batch, and parallelize into four streams
Load pretrained ResNet18
Modify final fully connected layer to 4 outputs
Set loss = Cross Entropy, optimizer = Adam
**for** each epoch **do**
    **for** each batch **do**
        Forward propagation
        Compute loss
        Backward propagation
        Update weights
        Apply gradient clipping
    **end for**
    Compute loss and accuracy
    Apply early stopping
**end for**
Compute confusion matrix
Plot loss and accuracy curves
Save model checkpoints
Compute total processing time

## 8.2 Algorithm-B: Transfer learning using the Pre-Trained Resnet18 model on ImageNet

**Require:** NickParvar Dataset, ResNet18 Architecture
**Ensure:** Accuracy, Training and Validation Time
**Dataset Preparation:**
Download dataset and organize into Training and Testing directories
**Training Data Transformation:**
Resize images, apply horizontal flip and random rotation
Convert images to tensors and normalize
**Testing Data Transformation:**
Resize images, convert to tensors, normalize
**Define Hyperparameter Search Space:**
$lr \in [10^{-5}, 10^{-1}]$, $bs \in [16, 512]$
$ep \in [10, 200]$, $pt \in [5, 20]$, $gc \in [0.1, 5]$
**Initialize MWOA Population:**
**for** each whale $X_i$ **do**
    $X_i = [lr, bs, ep, pt, gc]$
**end for**
Evaluate fitness of each whale
Identify best whale $X^*$
**for** each iteration **do**
    $a = 2 - \frac{2 \times iteration}{MaxIterations}$

**for** each whale $X_i$ **do**
Generate $r_1, r_2 \in [0, 1]$
$A = 2ar_1 - a$
$C = 2r_2$
Generate $p \in [0, 1]$
**if** $p < 0.5$ **then**
**if** $|A| < 1$ **then**
$D = |C \cdot X^* - X_i|$
$X_i^{new} = X^* - A \cdot D$
**else**
Select random whale $X_{rand}$
$D = |C \cdot X_{rand} - X_i|$
$X_i^{new} = X_{rand} - A \cdot D$
**end if**
**else**
$D' = |X^* - X_i|$
Generate $l \in [-1, 1]$
$X_i^{new} = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*$
**end if**
Apply boundary constraints
Update $X_i = X_i^{new}$
**end for**
Evaluate fitness of whales
Update best whale $X^*$
**end for**
Extract optimal hyperparameters from $X^*$
**Dataset Batching:**
Create batches and shuffle dataset
**Model Construction:**
Load ResNet18 and modify final layer
**Training Configuration:**
Use Cross Entropy loss and Adam optimizer
**for** each epoch **do**
**for** each batch **do**
Forward propagation
Compute loss
Backward propagation
Apply gradient clipping
**end for**
Compute epoch loss and accuracy
**end for**
Compute confusion matrix
Plot training curves
Save model
Compute total time

## 8.3 Algorithm-c: subsequent re-training using the RESNET18 Architecture

**Require:** Navneel, Sartar, Brats, Thomos, Jakesh Bohaju Datasets

**Ensure:** Accuracy and Processing Time for each dataset

**Define Hyperparameter Search Space:**
$lr \in [10^{-5}, 10^{-1}]$, $bs \in [16, 512]$
$ep \in [10, 200]$, $pt \in [5, 20]$, $gc \in [0.1, 5]$
**for** each dataset $D$ **do**
    **Dataset Preparation:**
    Download dataset and split into Training and Testing directories
    **Training Data Transformation:**
    Resize images, apply horizontal flip and random rotation
    Convert images to tensors and normalize
    **Testing Data Transformation:**
    Resize images, convert to tensors, normalize
    **Initialize MWOA:**
    **for** each whale $X_i$ **do**
        $X_i = [lr, bs, ep, pt, gc]$ randomly initialized
    **end for**
    Evaluate fitness of whales using validation performance
    Identify best whale $X^*$
    **for** each iteration **do**
        $a = 2 - \frac{2 \times iteration}{MaxIterations}$
        **for** each whale $X_i$ **do**
            Generate $r_1, r_2 \in [0, 1]$
            $A = 2ar_1 - a$
            $C = 2r_2$
            Generate $p \in [0, 1]$
            **if** $p < 0.5$ **then**
                **if** $|A| < 1$ **then**
                    $D = |C \cdot X^* - X_i|$
                    $X_i^{new} = X^* - A \cdot D$
                **else**
                    Select random whale $X_{rand}$
                    $D = |C \cdot X_{rand} - X_i|$
                    $X_i^{new} = X_{rand} - A \cdot D$
                **end if**
            **else**
                $D' = |X^* - X_i|$, $l \in [-1, 1]$
                $X_i^{new} = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*$
            **end if**

38: Apply boundary constraints
39: Update $X_i = X_i^{new}$
40: **end for**
41: Evaluate fitness of whales
42: Update best whale $X^*$
43: **end for**
44: Extract optimal hyperparameters from $X^*$
45: **Dataset Batching:**
46: Create batches using optimized batch size
47: Shuffle dataset and parallelize into four streams
48: **Model Preparation:**
49: Load pretrained model weights
50: Modify fully connected layer for 4 output classes
51: **Training Configuration:**
52: Loss = Cross Entropy
53: Optimizer = Adam (optimized learning rate)
54: Gradient Clipping = optimized value
55: Early Stopping Patience = optimized value
56: Epochs = optimized value
57: **for** each epoch **do**
58: **for** each batch **do**
59: Forward propagation through network
60: Compute loss and cumulative loss
61: Backward propagation using Adam optimizer
62: Apply gradient clipping
63: Perform prediction
64: Update processed and correctly classified counts
65: **end for**
66: Compute epoch loss and accuracy
67: **end for**
68: Compute confusion matrix
69: Plot loss and accuracy curves
70: Save model parameters and checkpoints
71: Compute total processing time for dataset $D$
72: **end for**

# 9 Results and Discussion

## 9.1 Optimised Hyperparameter Values

in ResNet18, the following Hyperparameters used, for which Optimised Values are to be found using the Modified Whales Algorithm. The Hyperparameters used in the ResNet Architecture and the ranges chosen for those parameters are shown in Table 2. The hypermeters and the space options of each parameter are fed to the Modified Whales optimisation algorithm, and the optimised values for the parameters achieved through the implementation of the algorithm is shown in Table 3.

**Table 3** ResNet Included Hyperparameter and the related Value Ranges

| Hyperparameter | Description | Range Start | Range End |
|---|---|---|---|
| Learning Rate ($\eta$) | Step size used by optimizer to update weights | 1e-5 | 1e-1 |
| Batch Size | Number of samples processed before updating weights | 16 | 512 |
| Number of Epochs | Number of passes through the dataset | 10 | 200 |
| Early Stopping Patience | Epochs to wait before stopping | 5 | 20 |
| Gradient Clipping | Prevent exploding gradients | 0.1 | 5 |

**Table 4** Best Hyperparameters Found after running the modified Whale's Algorithm

| Hyperparameter | Optimum Value Fetched |
|---|---|
| Learning Rate | 0.064 |
| Batch size | 122 |
| Epoch | 62 |
| Early stopping | 10,355 |
| Gradient Clip | 1.123 |
| Best Fitness | 0.05 |

## 9.2 Transfer Learning using Resnet18 Trained on ImageNet with Whale Optimisation Included I

In the first instance, ResNet-18, pretrained on ImageNet and optimised for whales, was used to predict brain tumours on the Parvar brain tumour dataset, achieving a maximum accuracy of 94.13% after noise filtering and data augmentation. It has taken 6 Hours and 17 minutes to complete the training and validation. The retraining has taken an enormous amount of time, which is not acceptable at local diagnostic centres. Table 4 shows the performance of retraining the pre-trained ResNet-18 on ImageNet. The training accuracy and training loss are shown in Figures 2 and 3. It can be seen that a maximum accuracy of 94.13% could be achieved at the possible Minimum loss, considering 100 Epochs

**Table 5** Performance Analysis – Transfer Learning using pretrained Resnet18 Using ImageNet included with whales Optimisation

| Type of training | Data Set used | Number of Images | Architecture used | Feature used | Epochs Used | Accuracy | Time Taken for Training and Validation |
|---|---|---|---|---|---|---|---|
| Transfer Learning | Parvar | 7022 | RESNET18 trained on ImageNet | Regular | 150 | 94.77 % | 6Hours 55 Minutes |
| | | | | Noise filtering | 150 | 94.37% | 6Hours 32 Minutes |
| | | | | Data Augmentation | 150 | 94.13 % | 6Hours 17 Minutes |

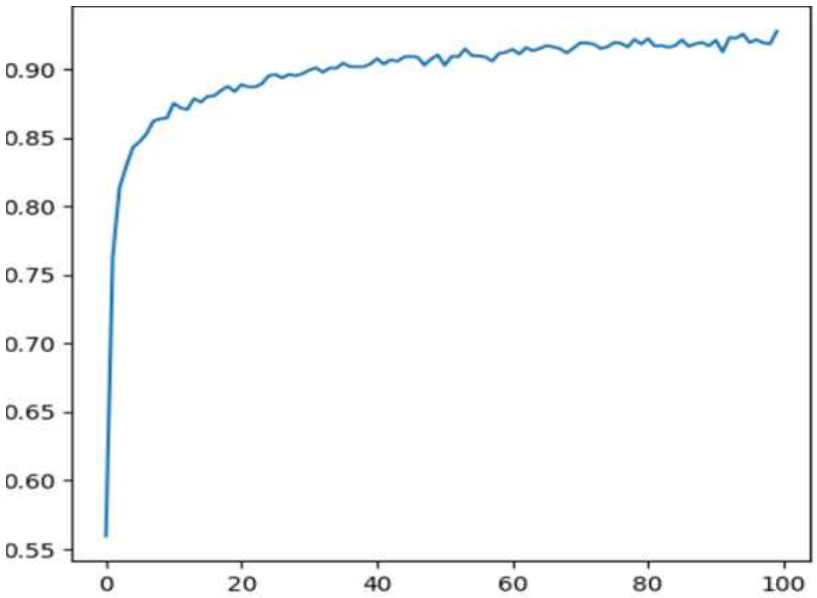



**Fig. 2** Training Accuracy – Pretrained ResNet18 Model

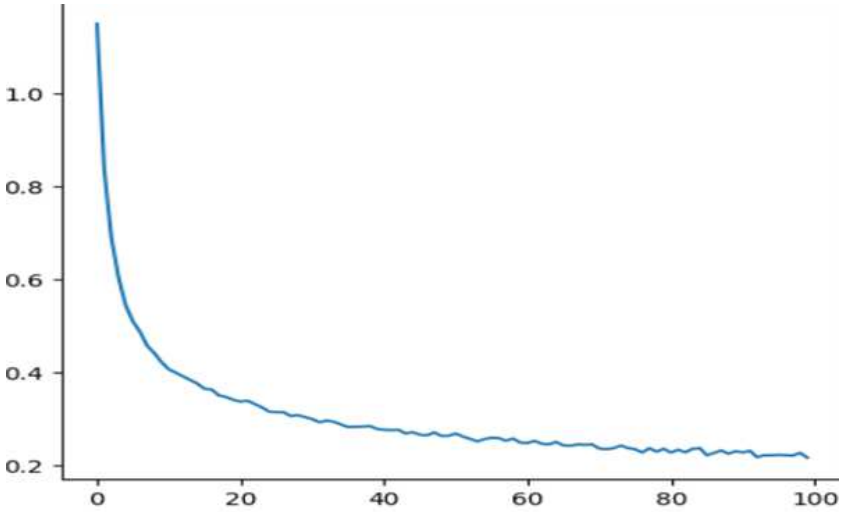



**Fig. 3** Training Loss - Pretrained ResNet18 Model

The classification report is shown in Table 6. From the classification report, it can be seen that a Maximum precision of 93 % could only be achieved. So, the inference is that Transfer learning using a pretrained RSNET18 with whale optimisation did not lead to much improvement. The primary reason is that ImageNet images are different from Brain Tumour MRI images.

**Table 6** Performance of ResNet18 trained using Paravar Data set and retraining using other data sets

| Type of training | Data Set used | Number of Images | Architecture used | Epochs Used | Accuracy | Time Taken for Training and Validation |
|---|---|---|---|---|---|---|
| Pre-Training | Parvar | 7022 | RESNET18 | 150 | 99.00% | 1Hour 7 Minutes |
| Re-Training | Sartar | 3096 | RESNET18 | 150 | 1000.00% | 50 Minutes |
| | Brats | 319 | RESNET18 | 150 | 100.00% | 55 Seconds |
| | Thomos | 3096 | RESNET18 | 150 | 100.00% | 45 Seconds |
| | Bohaju | 3764 | RESNET18 | 150 | 100.00% | 15 Seconds |
| Local Re-training | Navneel | 253 | RESNET18 | 150 | 100.00% | 2 Seconds |

## 9.3 Pre-training Resnet18 with whale optimisation included using Brain Tumour Data and then Transfer Learning using Brain Tumour Data sets

The next approach is to pre-train ResNet18 with whale optimisation to tune hyperparameters across different brain tumour datasets, thereby developing a domain-specific pre-trained model that can be retrained on other domain-specific datasets, thereby making the model well-established. The model can be trained using local, small-sized data quite quickly, and the same is used for immediate predictions

Table 7 details the pre-training of the ResNet-18 architecture, including whale optimisation, using the Parvar dataset. It took 2 Seconds to complete training on the Local data after 150 Epochs, achieving an accuracy of 100%, which is much better than the 94.13% we achieved with pure transfer learning using whale optimisation.

**Table 7** Performance of ResNet18 trained using Paravar Data set and retraining using other data sets

| Type of training | Data Set used | Number of Images | Architecture used | Epochs Used | Accuracy | Time Taken for Training and Validation |
|---|---|---|---|---|---|---|
| Pre-Training | Parvar | 7022 | RESNET18 | 150 | 99.00% | 1Hour 7 Minutes |
| Re-Training | Sartar | 3096 | RESNET18 | 150 | 1000.00% | 50 Minutes |
| | Brats | 319 | RESNET18 | 150 | 100.00% | 55 Seconds |
| | Thomos | 3096 | RESNET18 | 150 | 100.00% | 45 Seconds |
| | Bohaju | 3764 | RESNET18 | 150 | 100.00% | 15 Seconds |
| Local Re-training | Navneel | 253 | RESNET18 | 150 | 100.00% | 2 Seconds |

The training time decreases while accuracy remains constant as more training datasets from the same domain are used for retraining, clearly demonstrating the model's optimisation. When data sets have fewer images, the response time is 2 seconds with 100

Table 8 shows the classification results when trained on the Nanvn data set, which has just 253 Images. It can be seen that accuracy and precision are 100

The accuracy and loss graphs are shown in Figure 4 and Figure 5. From the figures, it can be seen that the loss is zero. Accuracy is 100

**Table 8** Revised Classification Report when trained locally using a small data set

| Type of tumour | Precision | recall | F1-Score | Support |
|---|---|---|---|---|
| Glioma | 1.00 | 1.00 | 1.00 | 33 |
| Meningioma | 1.00 | 1.00 | 1.00 | 33 |
| No tumour | 1.00 | 1.00 | 1.00 | 33 |
| Pituitary | 1.00 | 1.00 | 1.00 | 33 |
| Accuracy | | | 1.00 | 132 |
| Macro Average | 1.00 | 1.00 | 1.00 | 132 |
| Weighted Average | 1.00 | 1.00 | 1.00 | 132 |

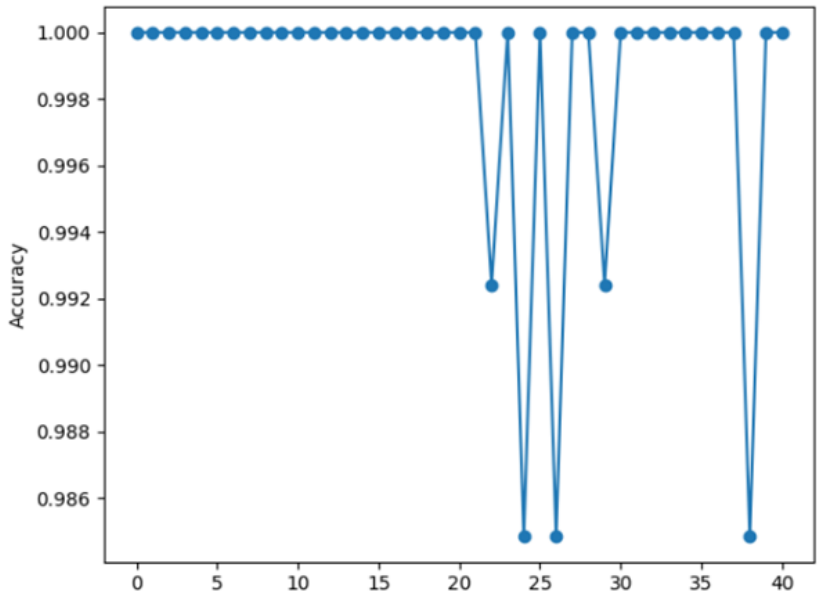



**Fig. 4** Training Accuracy – Training Accuracy – Local Data

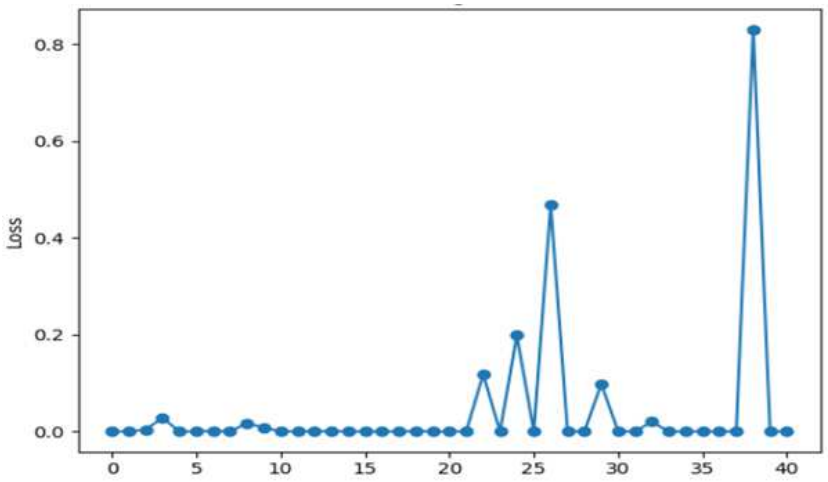



**Fig. 5** Local data

sectionExperimental Setup

All the algorithms have been run on a laptop with an 8-CPU and 2-GPU model. The performance could be much better when tried on a Cloud with a greater number of resources

## 10 Conclusions

A pretrained model, such as ResNet-18 trained on ImageNet data, will take a lot of time and achieve lower accuracy when used for Transfer learning on brain tumour data. The performance and time taken to train slightly improve when whale optimisation is used for determining the optimised values of hyperparameters, which are fed to train the model A pretrained model trained on an architecture like RESNET18, using a domain-specific dataset and optimised hyperparameters, yields the most appropriate models for transfer learning. Transfer learning can be implemented with less time and with 100The research can be extended to test recursive training on domain-specific data using the VCGNet, U-Net, and MobileNet architectures to reduce training time and determine whether 100% accuracy can be achieved.

## References

[1] Lisa M. DeAngelis, “Brain tumors,” New England Journal of Medicine, vol. 344, no. 2, pp. 114–123, 2001.

[2] David N. Louis, Arie Perry, Guido Reifenberger, Andreas von Deimling, Dominique Figarella-Branger, Webster K. Cavenee, Hiroko Ohgaki, Otmar D. Wiestler, Paul Kleihues, and David W. Ellison, “The 2016 World Health Organization classification of tumors of the central nervous system: a summary,” Acta Neuropathologica, vol. 131, no. 6, pp. 803–820, 2016

[3] Muhammad Sajjad, Salman Khan, Khan Muhammad, Wanqing Wu, Amin Ullah, and Sung Wook Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," Journal of Computer Science, vol. 30, pp. 174–182, 2019

[4] Hamed Khotanlou, Olivier Colliot, Jamal Atif, and Isabelle Bloch, "3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models," Fuzzy Sets and Systems, vol. 160, no. 10, pp. 1457–1473, 2009.

[5] Gouenou Coatrieux, Hui Huang, Huazhong Shu, Limin Luo, and Christian Roux, "A watermarking-based medical image integrity control system and an image moment signature for tampering characterization," IEEE Journal of Biomedical and Health

[6] Siva Polepaka, Chandra Sekhar Rao, and M. C. Mohan, "IDSS-based two stage classification of brain tumor using SVM," Health Technology, pp. 1–10, 2019.

[7] Eirini I. Zacharaki, Shun M. Wang, Sanjay Chawla, Soo Mee Yoo, Ronny Wolf, Elias R. Melhem, and Christos Davatzikos, "Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme," Magnetic Resonance in Medicine, vol. 62, no. 6, pp. 1609–1618, 2009.

[8] Deepak Ravi, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang, "Deep learning for health informatics," IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 4–17, 2017

[9] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud A. A. Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez, "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017

[10] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," Computers in Biology and Medicine, vol. 111, p. 103345, 2019.

[11] Zunairah N. K. Swati, Muhammad Sajjad, Muhammad Haris Khan, Shahid Raza, Arif Mehmood, Khurram Khurshid, and Sung Wook Baik, "Brain tumor classification for MR images using transfer learning and fine-tuning," Computerized Medical Imaging and Graphics, vol. 75, pp. 34–46, 2019

[12] Muhammad Sajjad, Salman Khan, Khan Muhammad, Wanqing Wu, Amin Ullah, and Sung Wook Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," Journal of Computer Science, vol. 30, pp. 174–182, 2019.

[13] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," Proceedings of the IEEE, vol. 104, no. 1, pp. 148–175, 2016.

[14] Ainur R. Minarno, Moh. H. C. Mandiri, Yufis Azhar Munarko, and Heru Hariyady, "Convolutional neural network with hyperparameter tuning for brain tumor classification," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol. 6, no. 2, pp. 127–132, May 2021. doi:10.22219/kinetik.v6i2.1219.

[15] Mohamed Ait Amou, Kun Xia, Samir Kamhi, and Mohamed Mouhafid, "A novel MRI diagnosis method for brain tumor classification based on CNN and Bayesian optimization," Healthcare, vol. 10, no. 3, p. 494, Mar. 2022. doi:10.3390/healthcare10030494.

[16] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1409.1556

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 770–778

[18] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9.

[19] H. Sultan, A. El-Bendary, and A. Rafea, "Brain tumor classification using deep CNNs on MRI images," Journal of Healthcare Engineering, vol. 2019, pp. 1–12, 2019.

[20] Abiwinanda, B. D. Nugroho, I. Santoso, and E. J. Kim, "Brain tumor classification using convolutional neural network," Indonesian Journal of Electrical Engineering and Computer Science, vol. 12, no. 1, pp. 1–6, 2019

[21] M. Sajjad, S. Khan, I. Lee, S. W. Baik, and J. Y. Lee, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," Journal of Medical Systems, vol. 43, no. 11, p. 314, 2019

[22] M. Hanif, N. Abiwinanda, S. T. Hesaputra, A. Handayani, and T. R. Mengko, "Brain Tumor Classification Using Convolutional Neural Network," Proc. ResearchGate, 2017.

[23] X. Gu, Z. Shen, J. Xue, and Y. Fan, "Brain Tumor MR Image Classification Using Convolutional Dictionary Learning With Local Constraint," Frontiers in Neuroscience, vol. 15, 2021.

[24] O. Özkaraca, O. İ. Bağrıaçık, H. Gürüler, F. Khan, J. Hussain, and U. Laila, "Multiple Brain Tumor Classification with Dense CNN Architecture Using Brain MRI Images," Soft Computing, 2022.

[25] R. Chelghoum, K. Allab, L. Meziane, and B. Beddiar, "Transfer Learning Using Convolutional Neural Network Architectures for Brain Tumor Classification from MRI Images," arXiv preprint arXiv:341728934, 2020.

[26] Ali Çınar and Mehmet Yıldırım, "Brain Tumor Classification with Transfer Learning Using ResNet50," Medical Hypotheses, vol. 139, p. 109684, 2020

[27] S. Deepak and P. Ameer, "Brain tumor classification using deep CNNs with transfer learning," Multimedia Tools and Applications, vol. 79, no. 21–22, pp. 15467–15484, 2020.

[28] Z. A. Swati, Q. Zhao, M. Kabir, S. Ali, Z. Ali, S. Ahmed, and J. Lu, "Content-based brain tumor retrieval using multimodal deep CNNs," Neurocomputing, vol. 364, pp. 374–386, 2019.

[29] Sinno Jialin Pan and Qiang Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[30] D. Rammurthy and P. K. Mahesh, "Whale Harris hawks optimization based deep learning classifier for brain tumor detection using MRI images," Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 6, pp. 3259–3272, 2022

[31] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in Proceedings of Neural Information Processing Systems (NIPS), pp. 2951–2959, 2012.

[32] D. Kilichev and W. Kim, "Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO," Mathematics, vol. 11, no. 17, art. 3724, Aug. 2023, doi:10.3390/math11173724.

[33] Y. Chen, R. S. S. S. Yadav and J. Li, "Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in fog computing," Knowledge-Based Systems, vol. 248, p. 108505, 2022, doi:10.1016/j.knosys.2022.108505.

[34] A. K. Desta, S. Ohira, I. Arai and K. Fujikawa, "Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots," Vehicular Communications, vol. 35, p. 100470, 2022

[35] M. Tawfik, A. H. Abdelrahman, M. E. El-Halees and A. H. Fahmy, "Optimized intrusion detection in IoT and fog computing using stacked autoencoders,

transformer-CNN-LSTM ensemble and hyperparameter tuning," Computers, Materials & Continua, vol. 74, no. 3, pp. 1823–1841, 2024.

[36] R. Khalid, A. Gul and A. H. Malik, "A survey on hyperparameter optimization algorithms and their applications," Journal of King Saud University — Computer and Information Sciences 2020

[37] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix, D. Deng and M. Lindauer, "Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2023.

[38] A. Morales-Hernández, I. Van Nieuwenhuyse and S. Rojas-González, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," Applied Intelligence / Artificial Intelligence Review (survey/arXiv), 2021–2022

[39] A. Kaissar, "Enhancing CNN-based Network Intrusion Detection through Four Hyperparameter Optimization Methods (Grid Search, GA, PSO, GWO)," Elsevier / ScienceDirect, 2025.

[40] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning,vol.63, no.1, pp.3–42,2006.

[41] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuß, "Sequential parameter optimization," in Proc.CEC,2005,pp.773–780

[42] F. Hutter, H. Hoos, and K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance," in Proc. ICML, 2014, pp. 754–762.

[43] F. Hutter, L. Kotthoff, and J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges.Springer,2019

[44] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, vol. 95, pp. 51–67, 2016.

[45] A. A. Heidari et al., "Harris hawks optimization: algorithm and applications," Future Generation Computer Systems, vol. 97, pp. 849–872, 2019.

[46] D. Rammurthy and P. K. Mahesh, "Whale Harris hawks optimization based deep learning classifier for brain tumor detection using MRI images," Journal of King Saud University, Computer and Information Sciences, vol. 34, pp. 3259–3272, 2022.

[47] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. Al Mamun, M. A. Rahman, S. Shatabda, and M.S. H. Mukta, "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," Decision Analytics

Journal, vol. 11, p. 100470, Apr. 2024, doi:10.1016/j.dajour.2024.100470

[48] M. N. Gea, Wanayumini, and R. Rosnelly, "Impact of hyperparameter tuning on CNNbased algorithm for MRI brain tumor classification," J. Tek. Inform., vol. 18, no. 1, pp. 87–100, Apr. 2025, doi: 10.15408/jti.v18i1.44147.

[49] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., Perth, WA, Australia, 1995, pp. 1942–1948.

[50] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," Adv. Eng. Softw., vol. 114, pp. 163–191, Dec. 2017.

[51] M. O. Lawrence, "A hybrid bio-inspired augmented with hyper-parameter deep learning model for brain tumor classification," J. Electr. Syst. Inf. Technol., vol. 12, no. 35, pp. 1–29,2025, doi: 10.1186/s43067-025-00207

[52] Karen Simonyan; Andrew Zisserman Very deep convolutional networks for large-scale image recognition ICLR, 2022, 10, 1–16

[53] Christian Szegedy; Wei Liu; Yangqing Jia; Pierre Sermanet; Scott Reed; Dragomir Anguelov; Dumitru Erhan; Vincent Vanhoucke; Andrew Rabinovich Rethinking the inception architecture for computer vision CVPR 2024 24 1 1–22

[54] Kaiming He; Xiangyu Zhang; Shaoqing Ren; Jian Sun Deep residual learning for image recognition CVPR 2012, 25, 1097–1105

[55] Mark Sandler; Andrew Howard; Menglong Zhu; Andrey Zhmoginov; Liang-Chieh Chen MobileNetV2: Inverted residuals and linear bottlenecks CVPR 2015

[56] Mingxing Tan; Quoc V. Le EfficientNet: Rethinking model scaling for CNNs, ICML 2016, 2818–2826

- Funding: As of now, we do not have funding for this work. The Authors of this paper will contribute to the Expenses
- Conflict of interest: The Authors have no conflict of interest with anybody
- Ethics Approval and consent to participate: We do not have an ethics committee in our University. We follow general guidelines to ensure adherence to ethical standards such as citation, quotation, and plagiarism.
- Consent of all Authors for publishing the paper: We have the consent of all authors for publishing this research work
- Availability of Data: We have used publicly available data only https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset
- Materials availability
- Code availability: The code developed for proving the results is available
- Author contribution: All Authors have contributed equally to this paper