Reserach Article

International Journal of Advanced Technology and Engineering Exploration, Vol 11(116)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10102420

Optimized brain tumor analysis in FLAIR-MRI LGG images: leveraging transfer learning and optimization for enhanced diagnosis and localization

P.Santhosh Kumar[1]*, V.P. Sakthivel[2], Manda Raju[3] and P.D. Sathya[4]
Research Scholar, Department of ECE, Annamalai University, Chidambaram, Tamil Nadu, India[1]
Assistant professor, Department of EEE, Government college of Engineering, Tamil Nadu, India[2]
Associate Professor, Department of ECE Kakatiya Institute of Technology and sciences, Warangal, India[3]
Assistant Professor, Department of ECE, Annamalai University, Chidambaram, Tamil Nadu, India[4]

Received: 08-October-2023; Revised: 24-July-2024; Accepted: 26-July-2024

Abstract

This research endeavour conducts a comprehensive exploration of an efficient approach for categorizing and delineating brain tumors in fluid-attenuated inversion recovery magnetic resonance imaging (FLAIR-MRI) low grade glioma (LGG) images. The precise diagnosis and localization of brain tumors are pivotal tasks in the domain of medical imaging, and the proposed approach demonstrates notable advancements in both accuracy and computational efficiency. In the realm of classification, the investigation harnesses the formidable residual network-50 (ResNet-50) architecture by applying transfer learning techniques. Transfer learning facilitates the utilization of pre-trained neural network weights, significantly augmenting the model's ability to generalize from a relatively limited medical image dataset. To further refine the classification model, the grey wolf optimizer (GWO), a biologically inspired optimization algorithm, was employed. This strategic choice enabled meticulous fine-tuning of vital parameters, including learning rate, dense layer configurations, batch size, and the number of training epochs. Subsequently, for the segmentation task, the previously developed ResUNet model was leveraged, specifically tailored for brain tumor segmentation. By seamlessly integrating this model with the classification framework, a comprehensive solution for brain tumor analysis is presented, simultaneously delivering an accurate diagnosis and precise localization of tumor regions within the images. The resulting classification model exhibits remarkable performance metrics, achieving a testing accuracy of 90.7%, a sensitivity of 88.8%, and a specificity of 97.01%. These metrics indicate the model's proficiency in discerning tumor and non-tumor regions within FLAIR-MRI images. This research highlights the potential of advanced deep learning (DL) techniques and optimization strategies to enhance the efficacy and reliability of brain tumor analysis in medical imaging, ultimately aiming to improve patient care and treatment planning.

Keywords

Brain tumor analysis, FLAIR-MRI, ResNet-50, Transfer learning, Grey wolf optimization.

1.Introduction

Texture changes in brain structure are used to analyze different neurological disorders. Among various neurological disorders, brain tumors are the leading cause of death. Irregular cell growth in the brain results in the formation of a mass, leading to the development of brain tumors, which can damage nerve cells and ultimately lead to death [1]. In India, the number of brain tumor cases is steadily increasing. According to "The International Association of Cancer Registries," 28,000 individuals are diagnosed with brain tumors annually in India, and 24,000 reported deaths are attributed to them.

There are different types of brain tumors, initially categorized into two groups: primary tumors and metastatic tumors. Primary tumors originate from brain cells and develop within the brain. They are further classified into Glial and non-Glial tumors based on the type of cells involved. These tumors are then differentiated into benign and malignant forms. The most common intracranial tumors include Glioma, Meningiomas, and Pituitary tumors[2].

Computer-aided diagnosis (CAD) streamlines the detection of brain tumors. Radiologists use magnetic resonance imaging (MRI) to visualize and analyze texture changes in brain tissues, enabling the characterization of brain tumors and neurological disorders [3]. In the traditional approach, the accurate

*Author for correspondence

analysis of MRI images relies on the skills and knowledge of radiologists. Processing a large volume of data manually can lead to increased costs and inaccuracies [4].

Advancements in computational techniques have led to the integration of machine learning (ML) approaches into CAD systems. ML algorithms extract features and perform classification tasks. However, conventional ML algorithms often require human intervention, and the performance of these models is sensitive to expert knowledge [5].

In recent years, deep learning (DL) has emerged as a powerful tool for feature extraction and classification. Convolutional neural networks (CNNs), consisting of multiple hidden layers, excel at feature extraction and accurate classification and segmentation. Various pre-trained DL models, such as GoogleNet, AlexNet, and ResNet, have been employed in brain tumor classification [6, 7]. The classification performance depends on hyperparameters such as epochs, batch size, and the number of filters. Given the large and complex space of parameters, many researchers have used hyperparameter tuning in different applications. For example, [8] used hyperparameter optimization of a CNN model for tomato leaf classification, [9] used hyperparameter tuning of a CNN model for Osteosarcoma detection, and [10] used Unet for brain tumor segmentation, with model parameters tuned using a genetic algorithm (GA). However, finding the optimal parameters for a CNN model is a time-consuming process [11].

There are various traditional hyperparameter tuning algorithms, such as grid search, random search, and Bayesian optimization [12]. However, these algorithms are computationally expensive, especially in high-dimensional spaces, and may not efficiently handle noisy or non-convex optimization problems. Tuning hyperparameters presents a formidable NP-hard optimization challenge due to the expansive search space involved. Metaheuristic algorithms are widely acknowledged for their effectiveness in addressing such complexities [13].

To address these problems, an efficient and scalable hyperparameter tuning optimizer is required for high-dimensional datasets. Nature-inspired metaheuristic optimizers are used to tune the CNN model's hyperparameters, including its structural and training parameters.

The objectives of the proposed work can be summarized as follows:

- To develop an approach for the prediction and segmentation of brain tumors.
- To utilize the grey wolf optimization (GWO) approach to obtain optimal hyperparameter values, thereby enhancing the model's performance.
- To apply a model composed of two sub-modules:

(i)Employing ResNet-50 for classification with an improved approach using GWO,

(ii)Conducting brain tumor segmentation (BraTS) using a novel ResNet-based segmentation model [14].

- To conduct a comprehensive evaluation of the model's performance using various statistical parameters.

The paper is organized as follows: Section 1 provides an introduction to tumor classification and examines various machine learning models. Section 2 reviews previously established frameworks. Section 3 details the methods and dataset used. Section 4 offers an in-depth analysis of the proposed model's results. Finally, Section 5 presents a discussion of the results, and Section 6 concludes the paper.

2.Literature survey

This section explores into various existing frameworks dedicated to hyperparameter tuning for different CNN models primarily focused on brain tumor classification and BraTS. Achieving accurate classification and segmentation in this context is a challenging endeavour. State-of-the-art ML and DL models have been harnessed to advance the field, often coupled with optimization techniques for remarkable results. Different researchers work on hyperparameter optimization to perform CNN parameter tuning.

Kumar et al. [15] employed a two-step approach. First, they selected features through fuzzy c-mean clustering for segmentation, followed by feature extraction using k-nearest neighbors (K-NN) with GA to acquire the most salient features. Subsequently, they employed a deep neural network model for classification. In [16], Geetha and Gomathi focused on BraTS and classification. Fuzzy c-mean clustering was utilized for segmentation, and features were extracted from the segmented regions using the grey-level co-occurrence matrix (GLCM) and grey-level run length matrix (GLRM). Deep belief networks (DBN) were employed for classification, with hyperparameter tuning conducted using the GWO. Sindhu and Radha [17], combined segmentation and classification. K-means clustering was used for segmentation, while features were extracted using GLCM and GLRM. Feature optimization was carried out using particle swarm optimization (PSO) and the whale optimization technique (WOT). Classification was performed using decision trees (DT), support vector machines (SVM), K-NN, and AdaBoost classifiers. Mishra et al. [18] in performed BraTS through a deep convolutional neural network (DCNN). To enhance model accuracy, optimization methods such as GA, GWO, and the whale optimization algorithm (WOA) were applied to fine-tune the network's weights.

Emam et al. used the Improved Hunger Games Search algorithm (I-HGS) to optimize the performance of their model, specifically employing I-HGS-ResNet 50, for brain tumor classification on the BRAST dataset in [19].

Rammurthy and Mahesh [20] introduced whale Harris hawks optimization (WHHO) for segmentation and classification. Segmentation utilized Cellular Automata and Rough Set Theory, while statistical and structural features were extracted from the brain tumor using local optical oriented pattern (LOOP), mean, variance, and kurtosis. CNNs were then employed for classification. Tahir et al. [21], used Ant Colony Optimization for hyperparameter optimization of the VGG-16 transfer learning model to classify brain tumors.

In Chawla et al. [22] used a 2D Gabor filter to extract features from MRI images, and a Deep Neural Network was fine-tuned using the Echo Locator Bat algorithm. Researchers have explored various approaches for brain tumor classification using DL models and optimization algorithms.

ZainEldin et al. [23] employed the adaptive dynamic sine-cosine fitness grey wolf optimizer (ADSCFGWO) to optimize ResNet-V2 for brain tumor segmentation on the BraTS dataset. For binary classification, Kurdi et al. [24] introduced the harris hawks optimized convolutional network (HHOCNN) with tuned parameters on a Kaggle dataset. Ali et al. [25] investigated a combination of wrapper-based metaheuristic (WBM) and deep neural networks (DNNs). They optimized parameters extracted using the DNNs and performed classification with an SVM model.

Joshi and Aziz [26] proposed the Cuckoo Search Algorithm for optimizing hyperparameters of a deep neural network for brain tumor classification based on gene expression data. Arumugam et al. [27] introduced the Crossover Smell Optimizer for a Multilayer perceptron (MLP) used for classification with texture features. The MLP hyperparameters were optimized by the Crossover Smell Agent Optimizer. Geetha et al. [28] utilized a segmentation network for tumor segmentation and feature extraction, followed by classification with DenseNet. Notably, they optimized the segmentation network using the sine cosine Archimedes optimization algorithm (SCAOA), which also tuned DenseNet's hyperparameters. Finally, Xu et al. [29] employed the Contracted Fox Optimization Algorithm to optimize the Mobinet V2 DNNs model for brain tumor classification.

Researchers have utilized various datasets for brain tumor classification. Studies by [14–19] and [24–26] focused on the BraTS dataset, while [10] employed the T1-WCEMRI dataset. Others, particularly [22–24, 27], turned to the Kaggle dataset. MRI images were the primary focus for most researchers. However, [25] used a gene dataset obtained from the Kaggle public repository for brain tumor classification.

Several studies, including those from [13–15], [20, 22], and [25, 26], relied on manually extracted features, primarily texture features derived using techniques like GLCM, GLRM, Gabor filters, and LOOP. These features were then used to train and classify tumors within their models. Conversely, other researchers, such as those in [19, 24] and [23–27], leveraged DNNs like CNNs and pre-trained transfer learning models. Notably, the study in [25] utilized features extracted from a DNNs for classification using an SVM model.

In [15], researchers leveraged GA to reduce the number of features extracted using K-means clustering, before performing classification with a DNNs. Similarly, GWO was utilized in [16] to decrease feature dimensions, followed by classification using a deep belief network. The focus of [17, 18] was on hyperparameter tuning using the WOT. While [17] tuned the hyperparameters of an AdaBoost model, [18] applied it to a CNN model for brain tumor classification. Researchers in [20] combined whale and Harris Hawks optimization models to achieve feature reduction on MRI images before classification with a DNNs. Finally, the study in [24] employed the pre-trained HHOCNN model for brain tumor classification.

In [22], the Bat Optimizer facilitated the reduction of Gabor features before classification with a DNNs. For [25], deep feature extraction was achieved using autoencoders, followed by classification with a CNN model whose hyperparameters were tuned using a WBM.

Beyond manually extracted features, several studies focused on hyperparameter tuning for DL models. Specifically, [19] utilized the I-HGS algorithm, [21] employed the Ant Colony Optimizer, [26] used the Cuckoo Search algorithm, and [29] leveraged the Contracted Fox Optimization algorithm, all for tuning hyperparameters within CNN models. Similarly, [27] employed the Crossover Smell Agent Optimizer for optimizing the hyperparameters of a MLP, while [28] trained a Dense Network and tuned its hyperparameters using the SCAOA.

This section concludes by comparing these existing frameworks, taking into account the dataset, classifier, and optimization approach employed, as presented in *Table 1*.

Table 1 Comparison of various existing frameworks carried optimization approaches

Author name	Features	Classifier	Dataset/Data Source	Optimizer
Kumar et al., [15]	K-NN Features	DL	T1-WCEMRI dataset	GA
Geetha and Gomathi [16]	GLCM, GRLM	DBN	BraTS	GWO
Sindhu and Radha [17]	GLCM, GRLM	Adaboost Classifier	BraTS	PSO and AOT
Rammurthy and Mahesh [20]	LOOP, Mean, Variance, and Kurtosis	DNNs	BraTS	WHHO
Chawla et al., [22]	Gabor filter features	DNNs	BraTS	Bat echo optimizer
Mishra et al., [18]	NA	CNN	BraTS	WOT
Tahir et al., [21]	NA	VGG-16 Transfer Learning	BraTS	Ant Colony optimizer
Emam et al., [19]	DNNs	RESNET-50	BraTS	I-HGS algorithm
ZainEldin et al., [23]	NA	CNN	BraTS	Sine-Cosine Fitness
Kurdi et al., [24]	NA	CNN	Kaggle	Harris Hawks
Ali et al., [25]	Deep features using auto encoder	SVM	Kaggle	WBM optimizer
Joshi and Aziz [26]	Gene features	DNNs	Kaggle	Cuckoo search algorithm
Arumugam et al., [27]	Texture features	MLP	BraTS	Cross over smell agent optimizer
Geetha et al., [28]	Texture features	Densenet	BraTS	SCAOA
Xu et al., [29]	DNNs	Mobinet V2	Kaggle	Contracted Fox Optimization algorithm

In previous studies [15–17], ML models were utilized, while [20, 22] opted for DNNs to perform tumor classification. These models effectively optimize extracted features using different optimization techniques.

Addressing the classification and segmentation of brain tumors presents a range of challenges:

1. The classification of brain tumors from brain images poses formidable challenges due to two main factors. Firstly, brain tumors exhibit substantial variations in contrast, size, intensity, and shape, making their consistent identification difficult. Secondly, the presence of numerous pathological tumor types further complicates the classification process. However, recent advancements in DL models, as demonstrated by [18, 21], have shown promise in automating this process without requiring extensive human intervention. They achieved notable results by training on a limited dataset comprising T1W MRI images collected from the BraTS dataset.
2. The segmentation of brain tumors is a complex task. Many researchers have turned to techniques like k-means clustering [15] and fuzzy c-mean clustering [16] for segmentation, aiming to remove unwanted tissues and extract essential information from MRI images. Nevertheless, improving segmentation accuracy remains an ongoing challenge, necessitating further exploration and refinement of segmentation techniques.

3.Materials and methods

This research utilizes the lower grade glioma (LGG) FLAIR-MRI dataset from Kaggle repository [30]. The dataset consists of data from 90 patients sourced from various healthcare institutions [30]. This dataset contains FLAIR-MRI images, chosen for their ability to detect small, subtle abnormalities in brain tissues. FLAIR-MRI images provide high contrast between lesions and surrounding brain tissues it allowing for the characterization of brain tumor presence. Hyperintensity in FLAIR-MRI images arise from increased water content within the tumor. FLAIR-MRI highlights peritumoral edema, which can indicate the aggressiveness of a brain tumor and its physical impact on the surrounding brain tissue structures.

3.1Algorithms

Step1: Collect Data: Acquire the Kaggle LGG FLAIR-MRI dataset, including images and corresponding tumor masks.

Step 2: Resize all images to a uniform size (e.g., 256×256 pixels), Normalize pixel values to a range suitable for neural network input.

Step 3: Apply data augmentation techniques to increase dataset size and variability

Step 4: Load Pre-trained ResNet-50 Model, apply Transfer learning, Fine-tune later layers using the FLAIR-MRI dataset

Step 5: Set initial values for hyperparameters: learning rate, batch size, number of epochs, and dense layer configurations, optimize hyperparameters using GWO

Step 6: Use the pre-existing ResUNet model designed for image segmentation tasks.

Step 7: Combine the optimized ResNet-50 classification model with the trained ResUNet segmentation model.

Step 8: Evaluate Model Performance using statistical parameters

Step 9: Compare the performance metrics with existing methods to validate improvements.

FLAIR-MRI delineate the brain tissue showing the normal brain parenchyma and abnormal brain tissues that is very much helpful for surgical planning. The dataset includes 3,929 FLAIR-MRI slices, with 2,556 slices depicting brain regions without tumors and the remaining 1,373 containing tumors. The demographic details of the dataset shown in *Table 2*

Table 2 Dataset details

Subject average age	**Gender**	**Number of Patients**
20-75 Years	Male	32
	Female	56
	Unknown	1
	Online resource	1

Both tumor and non-tumor labels are included in the dataset, with their distribution visualized in *Figure 1*. Each MRI slice has dimensions of 256×256×3 pixels. Raw images were used for the classification and segmentation operation.

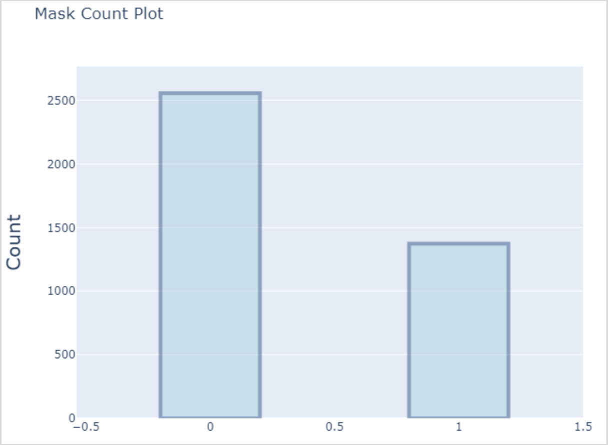
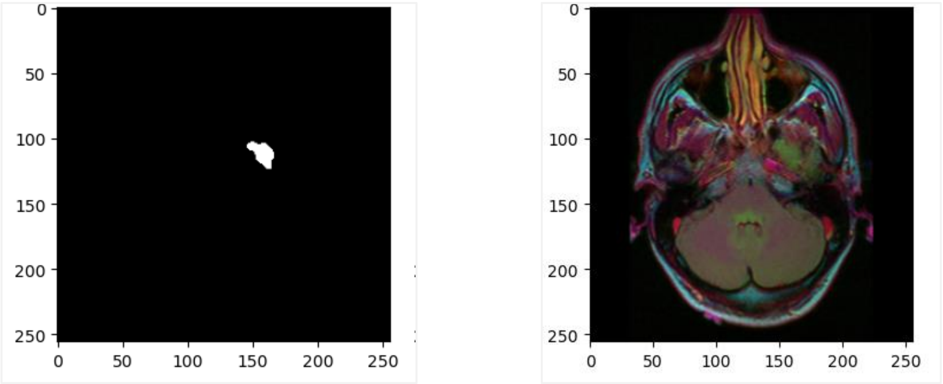


Figure 1 Bar chart of the dataset with respective labels

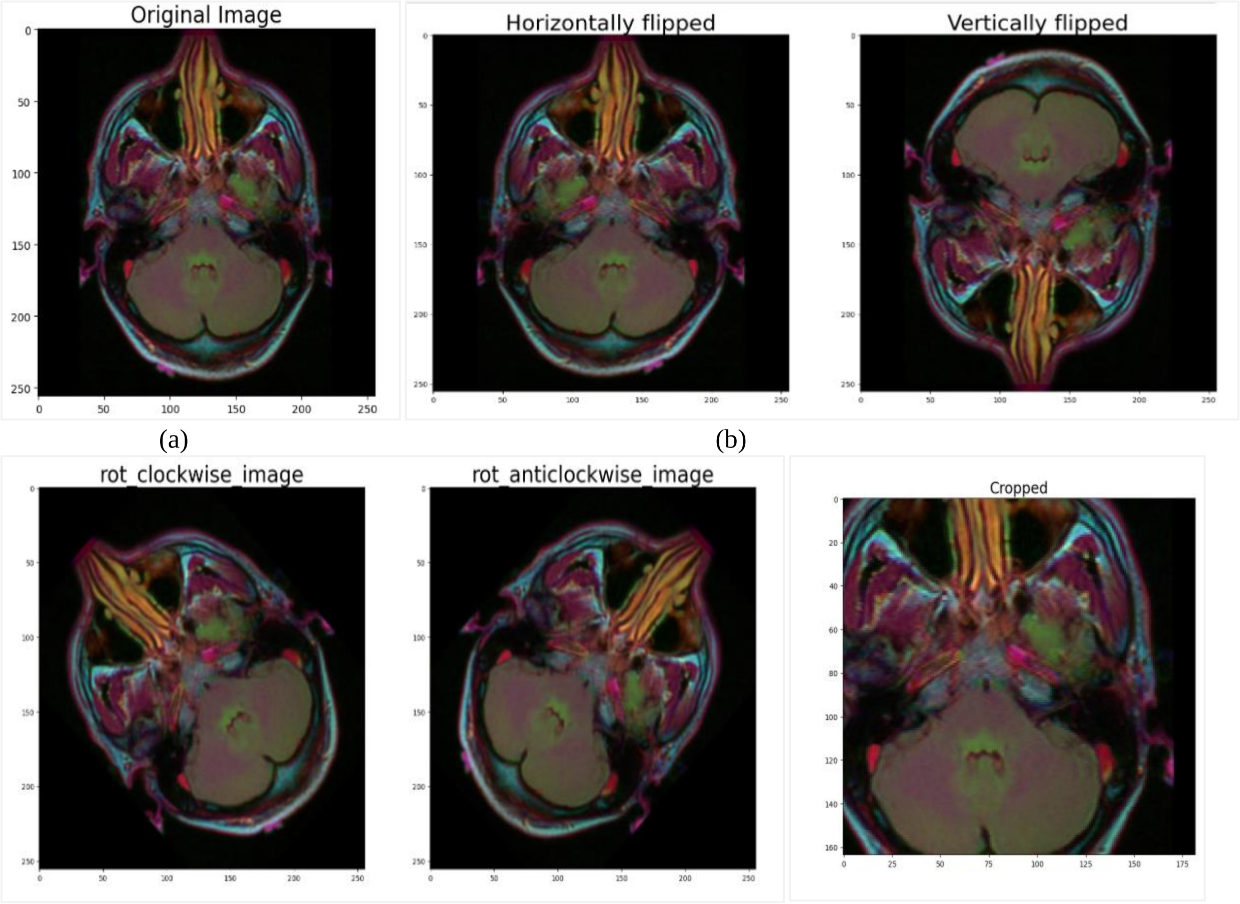
The dataset comprises FLAIR-MRI slices and corresponding masks indicating tumor locations. *Figure 2* displays samples from the dataset, showcasing the original image alongside its respective mask.

The image dataset comprises 3,929 images, is relatively small, posing a risk of overfitting. To address this and enhance the dataset's diversity, data augmentation techniques were applied. These techniques involve various image transformations, including rotation by (-45^o, 45^o), flipping in horizontal and vertical, shearing, and intensity variation with gamma values 0.2 and 2. This augmentation effectively increases the dataset's size and helps mitigate overfitting. *Figure 3* shows the augmentation of an image with various functions.



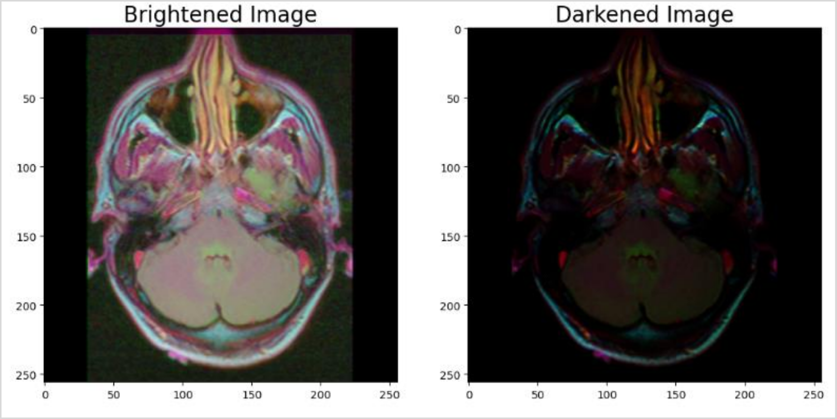
(a) (b)

Figure 2 Samples of the dataset (a) Original FLAIR-MRI image (b) Mask with tumor location



(a) (b)

(c) (d)



(e)

Figure 3 Augmentation of the dataset (a) Original Image, (b) Horizontal and vertical flipping, (c) Clockwise and anti-clockwise rotation by (45°) each side, (d) Sheering the original image (e) Intensity variation

3.2Grey wolf optimizer (GWO)

The GWO is a metaheuristic optimization inspired by the hunting behaviour of grey wolves. Grey wolves typically live in groups comprising 5 to 12 individuals [31–33] and they adhere to a strict social structure [34].

1. α wolf is the dominating among the pack (Group of 5-12 Gray wolfs) the decision taken by the α is final and it dictates the total pack
2. β is the subordinate level (second level) of the pack they help α in making decisions
3. δ Wolf dominates the ω wolf and δ come next in the chain with αandβ. δ look after the security of the pack.
4. ω is the last positioning in pack they fulfil the total pack predominantly.

Among these agents α is the fittest solution, β is the second fittest solution, δ is the third fittest solution and the remaining all are considered as ω. Mathematical modelling of the grey wolf optimization explains in the following equations in three stages.

Stage1: Hunting process:

The important approach is searching for prey. Where searching means distance between grey wolf and the prey. During the iteration a, the wolf position at iteration 't' is represented as $G(t)$, and the prey is at $G_P(t)$

The distance between the prey and wolf is calculated using Equation 1 and Equation 2.

$D = |HG_P(t) - G(t)|$ (1)

$H = 2r_2$ (2)

$r_2 = rand(0,2)$ (3)

H is the randomness used to tune the model to get optimum value the value decreases from 2 to 0 as shown in Equation 3.

Stage2: Roundup Process

In second step the wolf is encircling the prey. The mathematical model of the prey encircling is formulated as Equation 4, 5, 6, 7, 8, 9.

$G_i^d(t+1) = G_p^d(t) - B_i^d . D_i^d$ (4)

$D_i^d = \left|H_i^d G_p^d(t) - G_i^d(t)\right|$ (5)

$B_i^d = 2br_1 -$ b (6)

$H_i^d = 2r_2$ (7)

b $= 2 -$ t$/t_{max}$ (8)

r$_1 r_2 = round(0.2)$ (9)

$B_i^d . D_i^d$ denotes the enclosed steps; t_{max} denotes the maximum iterations; t represents the current number of iterations; b denotes the convergence factor its value decreases linearly from 2 to 0. To move to optimal global position B_i^d and D_i^d chosen randomly

Step 3: Attack Process

Updating the alpha, beta, Gama, and delta wolves, the algorithm accurately determines the optimum global position and achieves the required values are calculated using Equation 10, 11, 12, 13.

$W_1 = W_\alpha - B_1 D_\alpha$ (10)

$W_2 = W_\beta - B_2 D_\beta$ (11)

$W_3 = W_\gamma - B_3 D_\gamma$ (12)

$W = W_1 + W_2 + W_3/3$ (13)

Position of the alpha, beta, Gama, wolves represented asW_1, W_2, W_3. B_1, B_2, B_3 represent the random numbers. Wolfs encircling prey is denoted as$B_1 D_\alpha, B_2 D_\beta, B_2 D_\beta$.

3.3ResNet-50

ResNet is a conventional CNN model commonly employed for image recognition and classification [7]. Notably, deeper neural networks can be challenging to train effectively, but ResNet simplifies the training process and optimizes the network to achieve higher accuracy. This transfer learning model offers the advantage of mitigating the vanishing gradient problem through the use of skip connections.

Skip connection: ResNet employs an addition operation that combines the input with the output of a convolution block.

There are two distinct types of blocks within ResNet:

(i) Identity Block: When the input size matches the output size of the convolution block, the identity block is utilized. It simply adds the input and output together.

(***ii) Convolution Block:*** In cases where the input size differs from the output size, the convolution block is employed. This block adjusts the input size to match the output size and then adds them together. The modification of input size can be achieved through techniques like padding and 1×1 convolution, apart from using a standard convolution block.

In this context, the ResNet50 transfer learning model is utilized for feature extraction. The final dense layers of the model are removed, and custom dense layers are added for classification. To optimize the transfer learning model and achieve optimal accuracy, the GWO algorithm is used to fine-tune the hyperparameters.

For the segmentation of brain tumors, a previously published model is employed [14]. This model utilizes the novel ResUnet architecture to perform tumor segmentation.

3.4Proposed model

The proposed model is designed to carry out both the classification and segmentation of FLAIR-MRI brain tumors. Optimized brain tumor analysis in FLAIR-MRI LGG images diagram of the model shown in the *Figure 4*. Raw images are used for training and testing purpose. No enhancement and filtration techniques are applied over the images.

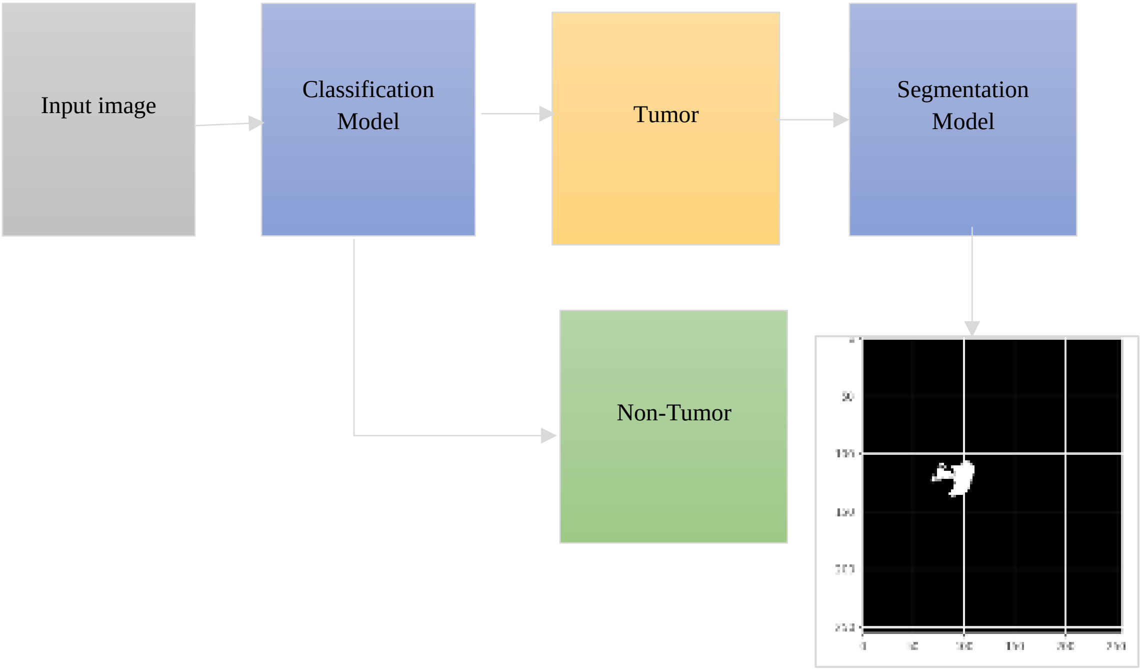


Figure 4 Overall model performs classification and extracting the tumor region

Figure 5, illustrates the utilization of the ResNet50 transfer learning model. Additional dense nodes are incorporated to facilitate the classification of FLAIR-MRI images into tumor and non-tumor categories. The number of nodes within the dense network, along with the activation function employed in the network, is considered as a critical hyperparameter. This hyperparameter is fine-tuned to achieve the highest possible accuracy during both the training and validation phases, as depicted in *Figure 6*.

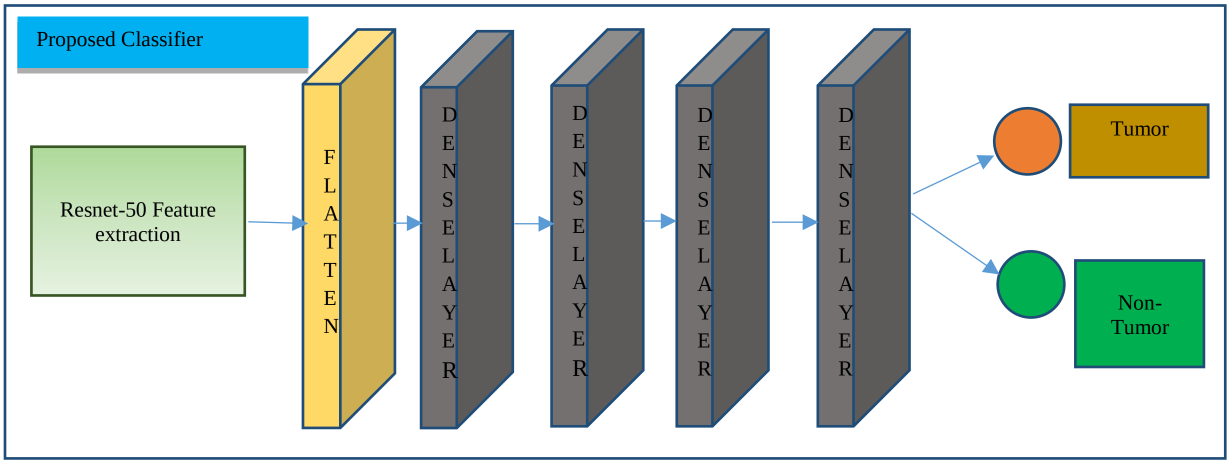


Figure 5 ResNet-50 transfer leaning classifier model

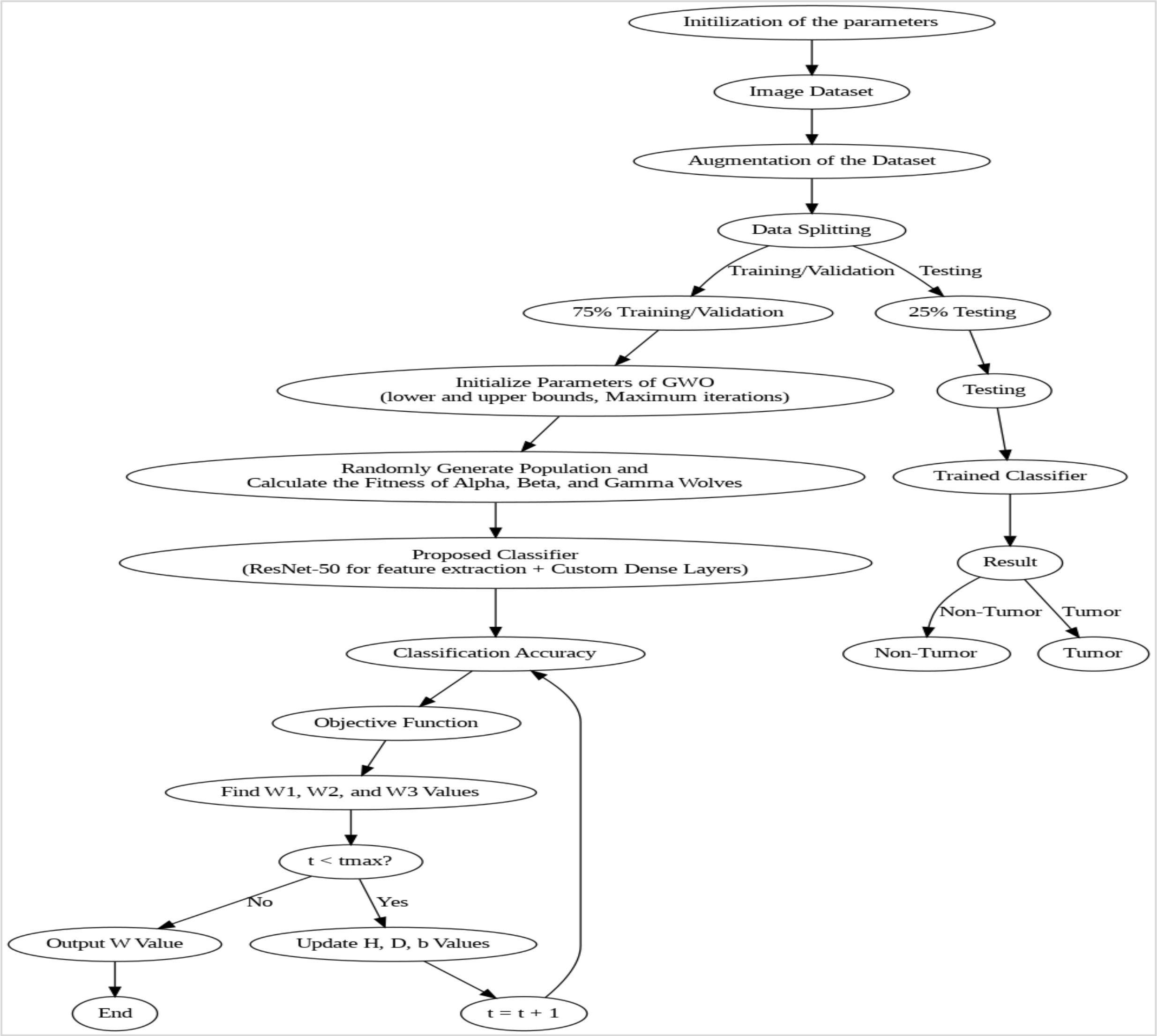

Initilization of the parameters
Image Dataset
Augmentation of the Dataset
Data Splitting
Training/Validation
Testing
75% Training/Validation
25% Testing
Initialize Parameters of GWO (lower and upper bounds, Maximum iterations)
Testing
Randomly Generate Population and Calculate the Fitness of Alpha, Beta, and Gamma Wolves
Trained Classifier
Proposed Classifier (ResNet-50 for feature extraction + Custom Dense Layers)
Result
Non-Tumor
Tumor
Classification Accuracy
Non-Tumor
Tumor
Objective Function
Find W1, W2, and W3 Values
t < tmax?
No
Yes
Output W Value
Update H, D, b Values
End
t = t + 1

Figure 6 Transfer learning ResNet-50 CNN hyperparameter tuning using GWO

Figure 6 represents the integration of the ResNet-50 transfer learning model with the GWO to fine-tune hyperparameters and improve classification accuracy.

3.4.1Flow diagram explanation

- **Start**: Initialize parameters
- **Image dataset**: Input the raw FLAIR-MRI images.
- **Augmentation of the dataset**: Apply data augmentation techniques to increase the dataset size.
- D**ata Splitting**: Divide the dataset into:
- **75% for training/validation**: Used to train and validate the model.
- **25% for Testing**: Used to evaluate the final model performance.
- **Initialize parameters of GWO**: Set the lower and upper bounds and the maximum number of iterations for the GWO.
- **Randomly generate population**: Generate an initial population of wolves and calculate their fitness based on classification accuracy.
- **Proposed Classifier**: Utilize ResNet-50 for feature extraction, replace final dense layers with custom dense layers for classification.
- **Classification accuracy**: Evaluate the accuracy of the classifier.
- **Objective function**: Define the objective function to optimize the classification accuracy.
- **Find W1, W2, and W3 Values**: Optimize these weights/parameters using GWO.
- **t < tmax?** Check if the maximum number of iterations has been reached.
- **No**: If the maximum iterations are reached, output the optimized weight values and end the optimization process.
- **Yes**: If not, update the positions of the wolves.
- **Update H, D, b Values**: Update the positions based on the objective function.
- t **= t + 1**: Increment the iteration counter.
- **Repeat**: Loop back to the classification accuracy step and repeat until the maximum iterations are reached.

3.4.2Hybridization of Residual Network-50 (ResNet-50) and GWO

- **Integration of GWO**: Integrate GWO into the training process for hyperparameter optimization.
- **Feature extraction**: Use ResNet-50 for extracting features from input images.
- **Dense layers**: Custom dense layers for enhanced classification.
- **Fine-tuning**: GWO optimizes parameters such as the number of kernels, learning rate, batch size, and number of epochs.
- **Enhanced accuracy**: Achieve high classification accuracy by avoiding local minima.

3.5Segmentation model

Following the classification of the image into tumor and non-tumor categories, the image is then passed to the segmentation model as proposed in the previous work [14]. This segmentation model is employed to achieve precise and accurate segmentation of the brain tumor.

3.5.1Experiment setup

This research was conducted on Google Colab using Python 3.7.1 with TensorFlow and Keras as the backend. The GPU used for training and validation was a Tesla K80 with 2496 CUDA cores, compute capability 3.7, and 24 GB of GDDR5 VRAM (23.439 GB usable).

In this research, the initial step involves the classification of images into either tumor or non-tumor categories. The study is conducted on the Kaggle FLAIR-MRI LGG dataset, which comprises a total of 3,929 images. These images are split into training, validation, and testing sets.

To enhance the dataset's diversity and meet the requirements of DL models, image augmentation techniques are applied initially. The ResNet-50 transfer learning model is then trained using 75% of the dataset, while the remaining 25% is used for testing.

Hyperparameter tuning is a crucial aspect of the training process and is carried out in two phases:
(i) Tuning of hyperparameters related to the network structure.
(ii)Tuning of hyperparameters used during the network training.

Network and hyperparameters of the network are trained using following steps:

Initializing: The initialization process involves creating a population of potential neural networks. This population represents different combinations of hyperparameters such as the number of nodes in the network architecture, learning rate, batch size, and number of epochs. In this work, a population size of 7 is used.

Fitness function: The performance of the neural networks is evaluated using a fitness function. In this work, use validation accuracy as the fitness metric.

Updating positions: The positions of the alpha, beta, and delta wolves (elements in the population) are updated during the optimization process. In this work, 1000 iterations are used for this process.

Termination criteria: The optimization process terminates when a specific termination criterion is met. In this work, early stopping is used with a patience of 15 epochs.

Both types of hyperparameter tuning are performed using the GWO algorithm. Detailed information about the specific hyperparameters can be found in *Table 3.*

Table 3 Hyperparameters tuned with lower limit and upper limit

Hyperparameter	Lowe limit	Upper limit
Learning rate	0.001	0.1
No. of nodes in 1st dense layer	5	512
No. of nodes in 2st dense layer	5	512
No. of nodes in 3rd dense layer	5	512
No. of nodes in 4th dense layer	5	512
Number of epochs	2	30
Batch size	20	50

4.Results

In an optimization approach, the objective function is the key component used to evaluate fitness and obtain the optimal solution for the hyperparameters. In this work, validation accuracy is obtained by training the model using the specified hyperparameters and evaluating its performance on a validation dataset.

The results are obtained by using early stopping, where parameter updates no longer improve the learning parameters, and the current best parameters are stored. In this algorithm, early stopping of the GWO algorithm is adopted; if the model does not improve its performance after 15 iterations, the parameters are frozen. Early stopping is used to prevent overfitting.

To optimize the hyperparameters of the classifier, parameters such as the learning rate, the number of nodes in the 1st dense layer, the number of nodes in the 2nd dense layer, and the number of epochs were considered. These hyperparameters were fine-tuned using the GWO algorithm, employing 50 epochs and 7 wolves.

After the hyperparameter tuning process with GWO, the following values were obtained:
Learning rate: 0.00842281957
Number of nodes in the 1st dense layer: 256
Number of nodes in the 2nd dense layer: 256
Number of nodes in the 3rd dense layer: 256
Number of nodes in the 4th dense layer: 256
Number of epochs: 5
Batch size: 35

Table 4 shows the custom dense layer after performing hyperparameter tuning. The model produces 25,829,890 number of trainable parameters, 53,120 number of non-trainable parameters, total 25,883,010 number of parameters are generated.

The dataset is split into training, validation, and test sets as shown in *Table 5*. The dataset is divided into two groups: 75% of the non-tumor and tumor data is used for validation, and the remaining 25% is used for testing. Using 75% of the data for training ensures that a substantial amount of data is available for training and validation, while 25% of unbiased data is reserved for testing. This approach mitigates the risk of overfitting.

The model is trained with early stopping, where validation loss is considered as the checkpoint to tune the model's performance. When the model's performance ceases to improve, a callback operation is performed with a delta rate of 0.0001, and the learning rate is reduced by a factor of 0.2.

The optimized model was trained and validated using 75% of the dataset, and its performance was analyzed through accuracy and loss curves. *Figure 7* displays the training and validation loss and accuracy curves. Following the hyperparameter tuning process, the model achieved an impressive validation accuracy of 95.5%. The trained model was tested with the remaining 25% of the dataset, and the results were used to generate the confusion matrix, as illustrated in *Figure 8.*

Table 4 Customized dense layer parameters and extended formation of every layer

Type	Output shape	Parameters	Connected to
dense_5	(None, 256)	2097408	['Flatten [0][0]']
dropout_3	(None, 256)	0	['dense_5[0][0]']
dense_6	(None, 256)	65792	['dropout_3[0][0]']
dropout_4	(None, 256)	0	['dense_6[0][0]']

Type	Output shape	Parameters	Connected to
dense_7	(None, 256)	65792	['dropout_4[0][0]']
dropout_5	(None, 256)	0	['dense_7[0][0]']
dense_8	(None, 256)	65792	['dropout_5[0][0]']
dropout_6	(None, 256)	0	['dense_8[0][0]']
dense_9	(None, 2)	514	['dropout_6[0][0]']
Total params: 25,883,010			
Trainable params: 25,829,890			
Non-trainable params: 53,120			

Table 5 Splitting the dataset for training, validation and testing

Class	Training and validation	Testing	Total
Non-Tumor	1917	639	2556
Tumor	1030	343	1373
Total	2947	982	3929

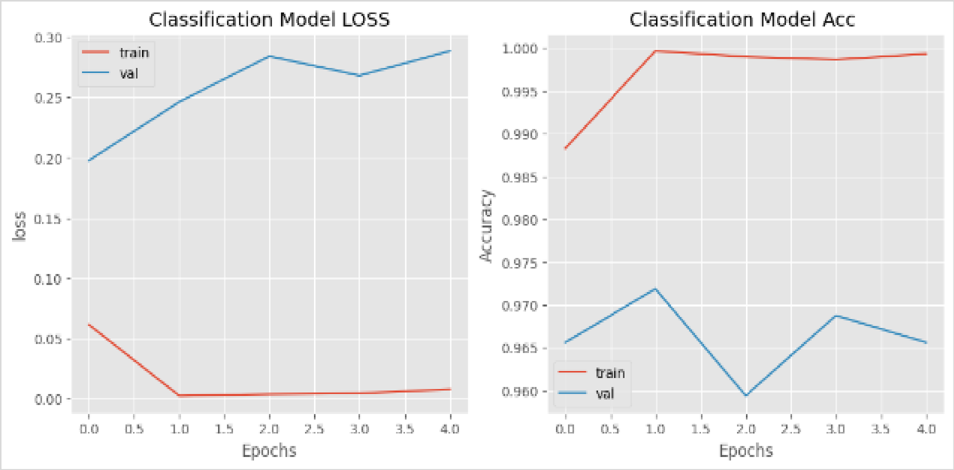


Figure 7 Model performance

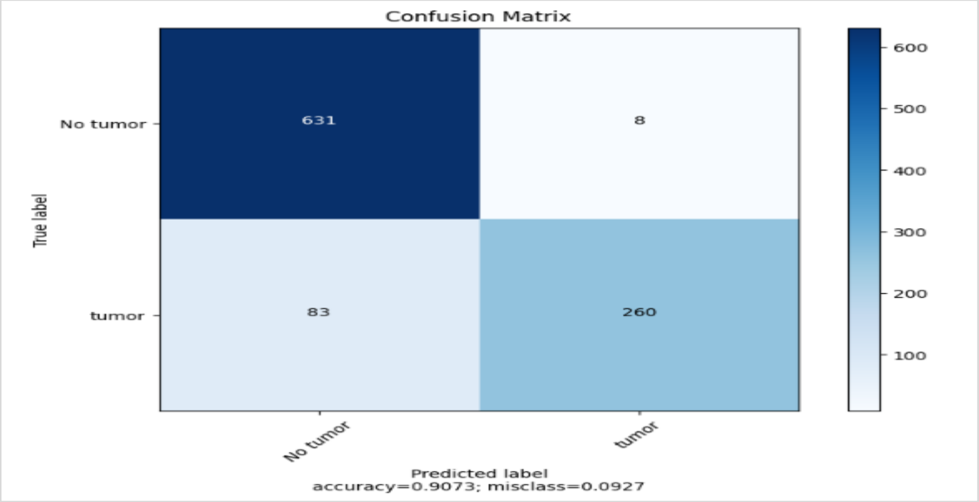


Figure 8 Confusion matrix performs binary classification

Various performance measuring parameters for the classification model are calculated using the confusion matrix, employing the following Equation 14, Equation 15, Equation 16 and Equation 17.

$$\text{Sensitivity}=\frac{True\ Positive}{True\ Positive+False\ Negative} \tag{14}$$

$$\text{Specificity}=\frac{True\ Negative}{False\ Positive+True\ Negative} \tag{15}$$

$$\text{Precision}=\frac{True\ Positive}{True\ Positive+False\ Positive} \tag{16}$$

$$\text{Accuracy}=\frac{True\ Positive+True\ Negative}{False\ Positive+False\ negative} \tag{17}$$

The resulting parameters are listed the *Table 6*.

Table 6 Performance measuring parameters generated using confusion matrix

Parameter	Value
Sensitivity	88.38
Specificity	97.01
Precision	98.75
Accuracy	90.73

Following the classification process, the images undergo further processing through the segmentation model [14]. The segmentation model is developed using a novel residual block to construct a ResUnet, which performs segmentation with encoding, decoding, and a bottleneck for schematic segmentation. The encoding and decoding layers are skip-connected to retain features and perform effective segmentation. The resulting images are presented in *Figure 9*. Segmentation is used to identify and label the brain tissues into normal and abnormal. It strengthens the decision taken by the radiologist. The classification approach tested with number of previous existing frame works as shown in the *Table 7*.

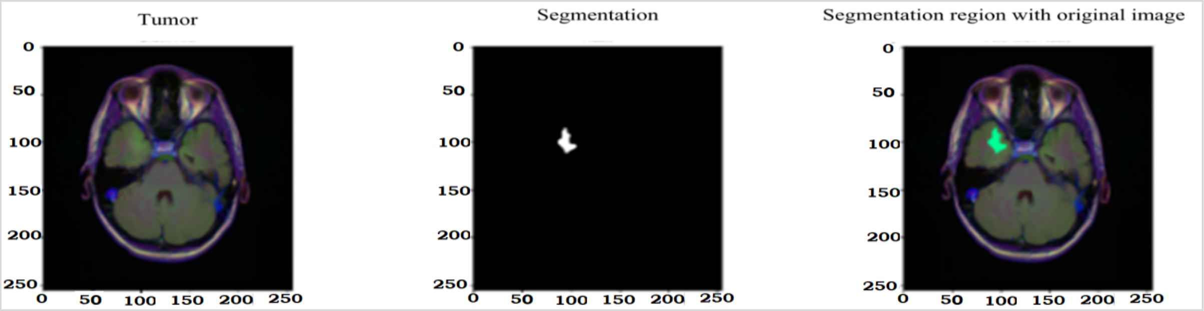


Figure 9 Classification result, segmentation and overlapping of original image with segmentation region

5.Discussion

Optimized brain tumor analysis in FLAIR-MRI LGG images encompasses both the classification and segmentation of brain tumor regions. Initially, the model is trained using 75% of the dataset, while the remaining 25% of an independent dataset is used for testing. To evaluate the performance of the proposed model, it is compared with various models proposed by different research scholars.

Table 7 shows a comprehensive overview of various studies and their corresponding results in the domain of medical image classification. The studies primarily focus on brain tumor classification using different feature extraction techniques, classifiers, datasets, optimizers, and performance metrics such as accuracy, sensitivity, and specificity. It is observed that the model provides better results compared to the existing models.

Several feature extraction techniques are employed across the studies, including K-NN features, GLCM, GRLM, Gabor filter features, LOOP, mean, variance, kurtosis, deep features using autoencoder, and texture features [13–16], [18–20, 19], [21–27]. These diverse techniques highlight the importance of extracting meaningful and relevant features for effective classification.

Classifiers employed range from traditional ML algorithms like MLP, SVM, and Adaboost to DL architectures such as DNNs, CNN, and transfer learning models like VGG-16 and ResNet-50 [13–16], [18–20, 19], [21–27]. The increasing adoption of DL architectures indicates their superiority in handling complex data and extracting intricate patterns, especially in medical image classification tasks.

The datasets used in these studies primarily include BraTS, T1-WCEMRI dataset, and Kaggle datasets [13–16], [18–20, 19] and [21–27]. BraTS is a commonly used dataset for brain tumor classification due to its comprehensive collection of MRI images with varying tumor types and characteristics. Kaggle repository offers a broader range of medical images datasets, allowing researchers to validate their models on diverse datasets.

Optimizers play a crucial role in fine-tuning the model parameters and enhancing classification performance. Various optimization techniques such as GA, GWO, PSO, WOT, ant colony optimizer, and cuckoo search algorithm have been employed [13–16], [18–20, 19] and [21–27].]. These optimizers aid in improving the convergence rate, avoiding local minima, and optimizing the model parameters effectively.

Comparing the results across the studies, it is evident that the choice of feature extraction technique, classifier, and optimizer significantly impacts classification performance. For instance, studies utilizing DL architectures like DNNs, CNNs, and transfer learning models tend to achieve higher accuracy scores compared to traditional ML algorithms [13–16], [18–20, 19], [21–27]. Similarly, optimization techniques like WOT, bat echo optimizer, and I-HGS demonstrate superior performance in optimizing model parameters and enhancing classification accuracy.
In the proposed approach, ResNet-50 is utilized with modifications made to the bottom layers. Four dense layers are incorporated at the bottom of the ResNet-50 model. Using the GWO algorithm, parameters such as the number of kernels, learning rate, batch size, and number of epochs are finely tuned. Subsequently, the proposed model is rigorously tested with a completely independent dataset, distinct from the training and validation sets. The trained model is assessed using a set of 982 images, achieving an impressive accuracy of 90.7%. The model achieves a balanced sensitivity of 88.38% and specificity of 97.01%, ensuring reliable identification of both positive and negative cases, which is crucial for clinical applications. Notably, this high level of accuracy is attained with just 5 epochs of training. Furthermore, the images classified by the model are subjected to segmentation using a previously published method, yielding precise and accurate region segmentation results.

Table 7 Comparison of previous models with proposed model

Author name	Features	Classifier	Dataset/ Data Source	Optimizer	Accuracy	Sensitivity	Specificity
Kurdi et al., [24]	NA	CNN	Kaggle	Harris Hawks	98%	NA	NA
Ali et al., [25]	Deep features using auto encoder	SVM	Kaggle	WBM optimizer	95.7%	NA	NA
Joshi and Aziz [26]	Gene features	DNNs	Kaggle	Cuckoo search algorithm	98.7	97.8	96.7
Proposed Model	NA	Resnet 50 Model	FLAIR-MRI LGG images	Grey wolf optimizer	90.73	88.38	97.01

5.1 Limitations of the model

While the proposed model achieves a sensitivity of 88.38%, there is room for improvement to ensure the accurate identification of all positive cases, which is crucial for the early detection and treatment of brain tumors. The performance of the proposed model heavily relies on the GWO. Exploring alternative optimization techniques and hybrid approaches could potentially enhance the model's robustness and generalization capabilities. The performance can be evaluated on larger and more diverse datasets to ensure its applicability across different clinical settings and populations. A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

This study achieves a significant breakthrough by accomplishing both brain tumor classification and segmentation with high accuracy. The GWO algorithm plays a crucial role, acting as an expert tuner that meticulously adjusts the hyperparameters of the ResNet-50 model. Remarkably, GWO fine-tuned the CNN model in just 15 epochs.

The results are truly impressive. By fine-tuning the classifier's parameters, the model achieved a phenomenal validation accuracy of 96.5% in just 5 epochs. This exceptional performance translated into a testing accuracy of 90.7%, demonstrating the model's generalizability to unseen data. This signifies a significant reduction in training time while maintaining exceptional accuracy. More importantly, the model achieves this high performance with a minimal number of trainable parameters. This is a significant advantage, as it reduces the risk of overfitting and improves the model's efficiency for real-world applications. In essence, this model stands out for its remarkable ability to perform both classification and segmentation of brain tumors with higher accuracy.

To further enhance the system's robustness, additional clinical parameters can be incorporated. Various optimization algorithms can also be explored to improve the optimization of the parameters.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Dataset availability

The dataset used for the experimentation is available at https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation.

Author's contribution statement

P.SanthoshKumar: Conceptualize, investigate, design model, data curation, coding, interpreting results, write and review the paper. **V.P. Sakthivel:** Investigate, analyse and interpret results, supervising and draft the paper. **Manda Raju:** Conceptualize, investigate, interpreting results and supervising. **P.D. Sathya:** Conceptualize, investigate, interpreting results and supervising.

References

[1] Louis DN, Perry A, Reifenberger G, Von DA, Figarella-branger D, Cavenee WK, et al. The 2016 world health organization classification of tumors of the central nervous system: a summary. Acta Neuropathologica. 2016; 131:803-20.

[2] https://www.hopkinsmedicine.org/brain-tumor. Accessed 04 June 2024.

[3] Wulandari A, Sigit R, Bachtiar MM. Brain tumor segmentation to calculate percentage tumor using MRI. In international electronics symposium on knowledge creation and intelligent computing 2018 (pp. 292-6). IEEE.

[4] Yang Y, Yan LF, Zhang X, Han Y, Nan HY, Hu YC, et al. Glioma grading on conventional MR images: a deep learning study with transfer learning. Frontiers in Neuroscience. 2018; 12:804.

[5] https://www.neuralconcept.com/post/the-future-of-cad-technology-innovations-and-implications. Accessed 04 June 2024.

[6] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In proceedings of conference on computer vision and pattern recognition 2015 (pp. 1-9). IEEE.

[7] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In proceedings of conference on computer vision and pattern recognition 2016 (pp. 770-8). IEEE.

[8] Alkaff AK, Prasetiyo B. Hyperparameter optimization on CNN using hyperband on tomato leaf disease classification. In international conference on cybernetics and computational intelligence (CyberneticsCom) 2022 (pp. 479-83). IEEE.

[9] Alsubai S, Dutta AK, Alghayadh F, Gilkaramenthi R, Ishak MK, Karim FK, et al. Group teaching optimization with deep learning-driven osteosarcoma detection using histopathological images. IEEE Access. 2024.

[10] Arif M, Jims A, Geman O, Craciun MD, Leuciuc F. Application of genetic algorithm and U-net in brain tumor segmentation and classification: a deep learning approach. Computational Intelligence and Neuroscience. 2022; 2022(1):5625757.

[11] Wang Y, Zhang H, Zhang G. cPSO-CNN: an efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. Swarm and Evolutionary Computation. 2019; 49:114-23.

[12] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research. 2012; 13(2):281-305.

[13] Venkatesh PK, Cohen MH, Carr RW, Dean AM. Bayesian method for global optimization. Physical Review E. 1997; 55(5):6219.

[14] Kumar PS, Sakthivel VP, Raju M, Sathya PD. Brain tumor segmentation of the FLAIR MRI images using novel ResUnet. Biomedical Signal Processing and Control. 2023; 82:104586.

[15] Kumar VV, Krishna KS, Kusumavathi S. Genetic algorithm based feature selection brain tumour segmentation and classification. International Journal of Intelligent Engineering & Systems. 2019; 12(5):214-23.

[16] Geetha A, Gomathi N. A robust grey wolf-based deep learning for brain tumour detection in MR images. Biomedical Engineering/Biomedizinische Technik. 2020; 65(2):191-207.

[17] Sindhu A, Radha V. An optimal feature selection with whale algorithm and adaboost ensemble model for pancreatic cancer classification in PET/CT images. Bioscience Biotechnology Research Communications. 2020; 13(4):18861894.

[18] Mishra PK, Satapathy SC, Rout M. Segmentation of MRI brain tumor image using optimization based deep convolutional neural networks (DCNN). Open Computer Science. 2021; 11(1):380-90.

[19] Emam MM, Samee NA, Jamjoom MM, Houssein EH. Optimized deep learning architecture for brain tumor classification using improved hunger games search algorithm. Computers in Biology and Medicine. 2023; 160:106966.

[20] Rammurthy D, Mahesh PK. Whale Harris hawks optimization based deep learning classifier for brain tumor detection using MRI images. Journal of King Saud University-Computer and Information Sciences. 2022; 34(6):3259-72.

[21] Tahir AB, Khan MA, Alhaisoni M, Khan JA, Nam Y, Wang SH, et al. Deep learning and improved particle swarm optimization based multimodal brain tumor

classification. CMC Computers, Materials & Continua. 2021; 68:1099-116.

[22] Chawla R, Beram SM, Murthy CR, Thiruvenkadam T, Bhavani NP, Saravanakumar R, et al. Brain tumor recognition using an integrated bat algorithm with a convolutional neural network approach. Measurement: Sensors. 2022; 24:100426.

[23] Zaineldin H, Gamel SA, El-kenawy ES, Alharbi AH, Khafaga DS, Ibrahim A, et al. Brain tumor detection and classification using deep learning and sine-cosine fitness grey wolf optimization. Bioengineering. 2022; 10(1):1-19.

[24] Kurdi SZ, Ali MH, Jaber MM, Saba T, Rehman A, Damaševičius R. Brain tumor classification using meta-heuristic optimized convolutional neural networks. Journal of Personalized Medicine. 2023; 13(2):1-18.

[25] Ali MU, Hussain SJ, Zafar A, Bhutta MR, Lee SW. WBM-DLNets: wrapper-based metaheuristic deep learning networks feature optimization for enhancing brain tumor detection. Bioengineering. 2023; 10(4):1-16.

[26] Joshi AA, Aziz RM. Deep learning approach for brain tumor classification using metaheuristic optimization with gene expression data. International Journal of Imaging Systems and Technology. 2024; 34(2):e23007.

[27] Arumugam M, Thiyagarajan A, Adhi L, Alagar S. Crossover smell agent optimized multilayer perceptron for precise brain tumor classification on MRI images. Expert Systems with Applications. 2024; 238:121453.

[28] Geetha M, Srinadh V, Janet J, Sumathi S. Hybrid archimedes sine cosine optimization enabled deep learning for multilevel brain Tumor classification using MRI images. Biomedical Signal Processing and Control. 2024; 87:105419.

[29] Xu L, Mohammadi M. Brain tumor diagnosis from MRI based on Mobilenetv2 optimized by contracted fox optimization algorithm. Heliyon. 2024; 10(1).

[30] Buda M, Saha A, Mazurowski MA. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Computers in Biology and Medicine. 2019; 109:218-25.

[31] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software. 2014; 69:46-61.

[32] Mirjalili S, Saremi S, Mirjalili SM, Coelho LD. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Systems with Applications. 2016; 47:106-19.

[33] Daniel E, Anitha J, Kamaleshwaran KK, Rani I. Optimum spectrum mask based medical image fusion using gray wolf optimization. Biomedical Signal Processing and Control. 2017; 34:36-43.

[34] Muro C, Escobedo R, Spector L, Coppinger RP. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. Behavioural Processes. 2011; 88(3):192-7.

P.Santhosh Kumar is a Research Scholar in the Department of ECE at Annamalai University, Chidambaram, Tamil Nadu, India. He is currently pursuing a Ph.D. He obtained his M.Tech in Embedded Systems from JNTU Hyderabad in 2011 and his B.Tech from JNTU Hyderabad in 2009. He has 13 years of teaching experience and has published more than 6 papers in international and national journals and conferences.
Email: santhoshguptha507@gmail.com

V.P. Sakthivel is an Assistant Professor in the Department of Electrical and Electronics Engineering at Government College of Engineering, Dharmapuri, Tamil Nadu, India. He obtained his Ph.D. degree from Annamalai University, Chidambaram, Tamil Nadu in 2012. He has published more than 70 research papers in reputed international journals, including IET, Elsevier, and Taylor & Francis, and has presented more than 30 research papers at various international conferences**.**
Email: vp.sakthivel@yahoo.com

Manda Raju is an Associate Professor and Head of the Department of Electronics and Communication Engineering at Kakatiya Institute of Technology & Science, Warangal, Telangana, India. He completed his Ph.D. in Signal Processing in Communications from Kakatiya University, Warangal, India, in 2017. His research interests include signal processing in communications, image processing, wireless networks, VLSI, and embedded systems. He has published more than 50 papers in international and national journals and conferences.
Email: m.raju2002@gmail.com

P.D. Sathya is an Assistant Professor in the Department of Electronics and Communication Engineering at Annamalai University, Chidambaram, Tamil Nadu, India. She obtained her Ph.D. degree from Annamalai University in 2012. Her research interests include signal processing, image and video processing, and optimization techniques applied to various image processing applications. She has published more than 100 research papers in reputed international journals, including Elsevier and Inderscience, and has presented over 30 papers at various international conferences.
Email: pd.satya@yahoo.in

Appendix 1

S. No.	Abbreviation	Description
1	ADSCFGWO	Adaptive Dynamic Sine-Cosine Fitness Grey Wolf Optimizer
2	BraTS	Brain Tumor Segmentation
3	CAD	Computer-Aided Diagnosis
4	CNNs	Convolutional Neural Networks
5	DBN	Deep Belief Networks
6	DCNN	Deep Convolutional Neural Network
7	DL	Deep Learning
8	DNNs	Deep Neural Networks
9	DT	Decision Trees
10	FLAIR-MRI	Fluid-Attenuated Inversion Recovery Magnetic Resonance Imaging
11	GA	Genetic Algorithm
12	GLCM	Grey-Level Co-occurrence Matrix
13	GLRM	Grey-Level Run Length Matrix
14	GWO	Grey Wolf Optimizer
15	HHOCNN	Harris Hawks Optimized Convolutional Network
16	I-HGS	Improved Hunger Games Search algorithm
17	K-NN	K-Nearest Neighbors
18	LGG	Low Grade Glioma
19	LOOP	Local Optical Oriented Pattern
20	ML	Machine Learning
21	MLP	Multilayer Perceptron
22	MRI	Magnetic Resonance Imaging
23	PSO	Particle Swarm Optimization
24	ResNet-50	Residual Network -50
25	SCAOA	Sine Cosine Archimedes Optimization Algorithm
26	SVM	Support Vector Machines
27	WBM	Wrapper-Based Metaheuristic
28	WHHO	Whale Harris Hawks Optimization
29	WOA	Whale Optimization Algorithm
30	WOT	Whale Optimization Technique