

Article

Automated Detection and Classification of Meningioma Tumor from MR Images Using Sea Lion Optimization and Deep Learning Models

Aswathy Sukumaran ^{1,2,*}  and Ajith Abraham ^{1,3} 

¹ Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, P.O. Box 2259, Auburn, WA 98071-2259, USA; ajith.abraham@ieee.org

² Department of Computer Science and Engineering, Jyothi Engineering College, Thrissur 679531, Kerala, India

³ Centre for Artificial Intelligence, Innopolis University, 420500 Innopolis, Russia

* Correspondence: aswathy.su@gmail.com

Abstract: Meningiomas are the most prevalent benign intracranial life-threatening brain tumors, with a life expectancy of a few months in the later stages, so this type of tumor in the brain image should be recognized and detected efficiently. The source of meningiomas is unknown. Radiation exposure, particularly during childhood, is the sole recognized environmental risk factor for meningiomas. The imaging technique of magnetic resonance imaging (MRI) is commonly used to detect most tumor forms as it is a non-invasive and painless method. This study introduces a CNN-HHO integrated automated identification model, which makes use of SeaLion optimization methods for improving overall network optimization. In addition to these techniques, various CNN models such as Resnet, VGG, and DenseNet have been utilized to give an overall influence of CNN with SeaLion in each methodology. Each model is tested on our benchmark dataset for accuracy, specificity, dice coefficient, MCC, and sensitivity, with DenseNet outperforming the other models with a precision of 98%. The proposed methods outperform existing alternatives in the detection of brain tumors, according to the existing experimental findings.

Keywords: convolutional neural network (CNN); boosted anisotropic diffusion filter; modified K-means clustering; magnetic resonance imaging; meningioma; sealion optimization



Citation: Sukumaran, A.; Abraham, A. Automated Detection and Classification of Meningioma Tumor from MR Images Using Sea Lion Optimization and Deep Learning Models. *Axioms* **2022**, *11*, 15. <https://doi.org/10.3390/axioms11010015>

Academic Editor: Oscar Humberto Montiel Ross

Received: 19 September 2021

Accepted: 20 December 2021

Published: 30 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The brain tumor and its analysis are of tremendous interest because of the growing innovations in medical image processing. An abnormal tissue development within the brain causes a tumor. According to the National Brain Tumor Foundation's (NBTF) global review, the improvement of brain tumor diagnoses among patients and the death rate due to brain tumors are outpacing earlier years' findings clinical experts can give patients more effective e-health care services, thanks to developments in medical imaging for this enhancement [1]. E-health care systems have a wide range of applications in medicine [2]. Due to their high accuracy and efficient results as presented by the radiologist, computer-vision-based biomedical imaging systems have gained appeal among clinical specialists, allowing them to handle treatment-related concerns more efficiently. Positron emission tomography (PET), magnetic resonance imaging (MRI), and computed tomography (CT) are the most popular modalities used to examine a brain tumor [3]. Magnetic resonance imaging (MRI) is well-known medical equipment that can be used to diagnose and study a variety of disorders, including brain tumors, neurological ailments, and epilepsy, among others. Typically, a system that is entirely processed by hardware/computer aids in the automation of this process, in order to produce correct and timely results. Machine learning (particularly deep learning) improvements have made it easier to discover, classify, and measure trends in medical images.

The major focus of this research is identifying tumor growth in the brain through the use of MRI scans. Moosa et al. discovered that these cancerous cells influence the health and functioning of the patient's brain. The most frequent benign intracranial tumor is meningioma [4]. Arachnoid cap cells, which are cells within the thin spiderweb-like membrane that protects the brain and spinal cord, give rise to these types of tumors. The arachnoid is one of three protective layers that surround the brain and spinal cord, commonly known as the meninges. These tumors normally grow inward, forcing on the brain or spinal cord, but they can also grow outward, thickening the skull. The majority of meningiomas are benign tumors that grow slowly. Certain blood vessels have cysts (fluid sacs), calcifications (mineral deposits), and tightly packed groupings of blood veins. Meningiomas are responsible for 30% of all types of brain tumors. They can occur in a variety of areas across the brain [5]. Many image processing technologies and methodologies have been used in the diagnosis and treatment of brain tumors. Artificial intelligence and machine learning, in particular, have played an essential role in the medical industry, as well as providing a key tool for supporting various medical professions. Several machine learning algorithms are employed to construct completely automated CAD systems, particularly for image segmentation and classification, which are used to aid radiologists as a second view [6].

1.1. Research Challenges and Related Problems

Normally, it is found that segmenting and classifying images in a database with a huge number of images needs more calculation time. The MRI images are recorded in numerous sagittal planes so that the database can be enlarged to include all possible planes. Preprocessing is the first step in creating a CAD system, and it is used to avoid overfitting by deleting undesirable areas and improving classification accuracy. It also aids in the removal of noise, which can cause visual blurring. In some circumstances, pixel correlation is ignored for greater speed, which can have an impact on the final result of the CAD system. The goal of this study is to use CNN architecture to solve the difficulty indicated above, as well as to categorize three different tumor types from an uneven database [7]. The efficiency of a system is determined by how effectively it performs in a complex context, as most existing systems are constrained to making minimal use of available resources.

1.2. Key Contribution

This paper focuses on detecting and efficiently classifying meningioma tumors using reformed DL models, with the following highlights:

- Image features are improved using a unique enhancing technique, and objective parameters such as PSNR, MSE, and SSIM are calculated.
- Clustering is a powerful, exploratory data analysis tool for gaining an intuition of the data's structure. The K-means algorithm is an iterative technique that attempts to split a dataset into K distinct non-overlapping subgroups (clusters), each of which has a data point that belongs to a single group. The proposed work made an attempt to make intra-cluster data points as comparable as possible while maintaining clusters as distinct one. The segmentation is performed by K-means and with modified K-means clustering methods. Performance measures from both techniques are calculated, and the best one is selected for further processing.
- After the optimized feature extraction technique, Sealion optimization (SLA12), the brain image is classified using Harris hawks optimization (HHO) [8]. The features are obtained with high accuracy and precision, and with much less computational time.

The paper is organized as follows: Section 1.3 provides a related study on the tumor from other researchers. Section 2 describes the details of the suggested approach and also includes experimental information and findings. Section 3 presents the conclusion and future work.

1.3. Related Works

Numerous steps have been taken to establish a highly accurate and reliable system for brain tumor identification. However, due to high inter and intra shape, texture, and contrast variations, it remains a challenging problem. On the basis of traditional ML and deep learning algorithms, a number of techniques for automatic brain tumor classification from MR images have been presented. Preprocessing is the first step in this process. Despite the numerous technologies available, extracting characteristics and identifying brain tumors remains a bit problematic. Furthermore, a universal approach for categorizing and detecting brain cancers is difficult to come by. A few alternative approaches have been explored using a variety of datasets and based on the methodology used for segmentation and classification. The number of images in the database is the largest issue with using neural networks to classify and segment MRI images. Furthermore, because MRI images are taken in many planes, the possibility of using all accessible planes could expand the database. Preprocessing is essential before inputting the images into the neural network, as this could impact the classification output by overfitting. The main focus of this research is to develop a system that properly recognizes and classifies tumors with a narrow percentage of interpretation error. For the classification of brain tumors in MRI scans, a variety of approaches and methodologies have been proposed. FCM, ANN, SVM, PCA, CNN, MLP, DWT, RF, KNN, K-means, EM, and others are some of the most commonly used algorithms in the classification and segmentation processes to extract vital information from medical imaging modalities. The findings of some of the most significant and well-known studies are summarized in Table 1.

Table 1. Overview of recent work using CNN method used for tumors detection.

Author	Methodology	Proposed Solution and Approach	Software/Tools/Languages/Simulation/Implementation Libraries	Evaluation	Challenges
Md Ahasan Kabir (2020) [9]	GLCM for feature extraction and ANN for classification	GA for feature selection, ANN classification model	Not mentioned	Accuracy: 99.5% for the BRATS, 98.3% for augmented BRATS Specificity: 97%	High dimensionality of matrix
S. Deepak (2019) [10]	Deep convolutional neural networks using transfer learning	GoogLeNet, deep CNN-SVM	MATLAB 2018b	Accuracy: 97.1%	Poor performances of the transfer learning model, misclassification of samples from the meningioma class.
Banerjee et al., (2018) [11]	Deep convolutional neural networks (ConvNets) trained on multi-sequence MR images.	DenseNet, VGG16	Tenser flow and Python	Accuracy: 97%	Computational complexity is high
Arasi, I P et al., (2019) [12]	GLCM feature extraction method. Lion-optimized boosting SVM	Genetic optimized median, hierarchical fuzzy clustering algorithm, SVM	MATLAB software	Accuracy: 97.69% Specificity: 96.7%	The correlation of pixels is not considered
Khaleda A Sathi et al., (2020) [13]	GLCM for feature extraction and ANN for classification	Gabor, DWT	Not mentioned	Accuracy: 97.99 Specificity: 98.34	Takes more time in image analysis
Zhou et al., (2018) [14]	Convolutional neural networks	DenseNet-RNN, DenseNet-LSTM, DenseNet-DenseNET	Tensor Flow, Nvidia Titan Xp GPU	Accuracy: 92.13%	Time consumption is high

Table 1. Cont.

Author	Methodology	Proposed Solution and Approach	Software/Tools/Languages/Simulation/Implementation Libraries	Evaluation	Challenges
Abdu, Hassan (2019) [15]	PCA-NGIST	PCA-NGIST for feature extraction and RELM for classification	MATLAB	Accuracy: 94.23 Specificity: 96.56	Information loss Overfitting problem
Nyoman et al., (2019) [16]	Convolutional neural network	AlexNet, VGG16, ResNet	MATLAB	Accuracy: 84.19%	Not satisfactory for noisy, nonuniform, and high-intensity images.
Narendra Mohan, (2020) [17]	Modified sea-lion-optimization-based KELA	SGLDM and LESH based feature extraction and MSLO-based KELM for classification	Not mentioned	Accuracy: 94.67% Specificity: 97.78%	Slow process
Roy, (2012) [18]	Modular approach to solve MRI segmentation	Symmetry analysis	Not mentioned	Accuracy: 91%	Time Consuming
Mohsen et al., (2018) [19]	Deep learning neural networks	Feature extraction using DWT and reduction using PCA technique and classification using DNN	MATLAB R2015a	Accuracy: 98 Specificity: 97.45	Time Consuming
Fatih Özyurt et al. (2019) [20]	Convolutional neural network	NS-EMFSE-CNN, SVM and KNN	Not mentioned	Accuracy: 95.62%	Long calculation time
Praveen G.B., Anita Agrawal, 2015 [21]	Machine learning	GLCM for feature extraction & SVM for classification	MATLAB 2014 resolution 256*256 using a 2.2 GHz, I3 windows OS machine.	Accuracy: 96.63 Specificity: 98.63	Takes more time
Mohsen et al., (2018) [19]	Deep neural network	Principal Components Analysis (PCA), Discrete Wavelet Transform (DWT)	Weka 3.9 and MATLAB R2015a	Accuracy: 96.97%	Long calculation time
Meenakshi, 2012 [22]	BPN classifier and orthonormal operators	K-means clustering, BPN classifier	MATLAB	Accuracy: 92%	Accuracy can be improved in less time
S. U. Aswathy et al., (2018) [23]	Wrapper-based genetic algorithm	Texture-based feature extraction, wrapper-based GA for feature selection, and SVM as classifier.	Not mentioned	Accuracy: 98.2 Specificity: 97.90	High dimensionality of matrix
Afshar et al., (2018) [7]	Convolutional neural networks (CNNs)	Capsule networks (CapsNets)	Python 2.7 and Keras library	Accuracy: 86.56%	Applicable for tumor core only
Manorama Sharma, G.N. Purohit and Saurabh Mukherjee, 2018 [24]	K-means and artificial neural network (KMANN)	GLCM is used for feature extraction and tumor classification by (KMANN)	Not mentioned	Accuracy: 98.65 Specificity: 97.89	Information loss
S. Deepak (2019) [10]	Deep convolutional neural networks using transfer learning	GoogLeNet, deep CNN-SVM	MATLAB 2018b	Accuracy: 97.1%	Poor performances of the transfer learning model, misclassification of samples from the meningioma class

2. Methodology

The goal of this research is to use deep learning techniques to implement various stages of detection and segmentation techniques in MRI brain imaging. To determine abnormality in the brain image, the proposed method employs an augmenting approach, as well as segmentation and classification techniques. The system's suggested framework is depicted in Figure 1. BRATS 2015 is the database used by the system. These input images are first preprocessed to eliminate noise and adjust the image's brightness and contrast for preprocessing, an adaptive diffusion filter and a boosted anisotropic diffusion filter were used; this outperforms the existing anisotropic diffusion filter and is compared to other filters to validate the performance. The ROI is retrieved from the preprocessed image using the modified K-means technique. Swallow swarm optimization (SSO) and adaptive swallow swarm optimization (ASSO) are used in identifying and extracting features from the segmented image, and SLA12 is used to optimize the extracted features. This improved feature is fed to the classifier, which divides images into normal and abnormal categories. CNN is the classifier utilized in this case.

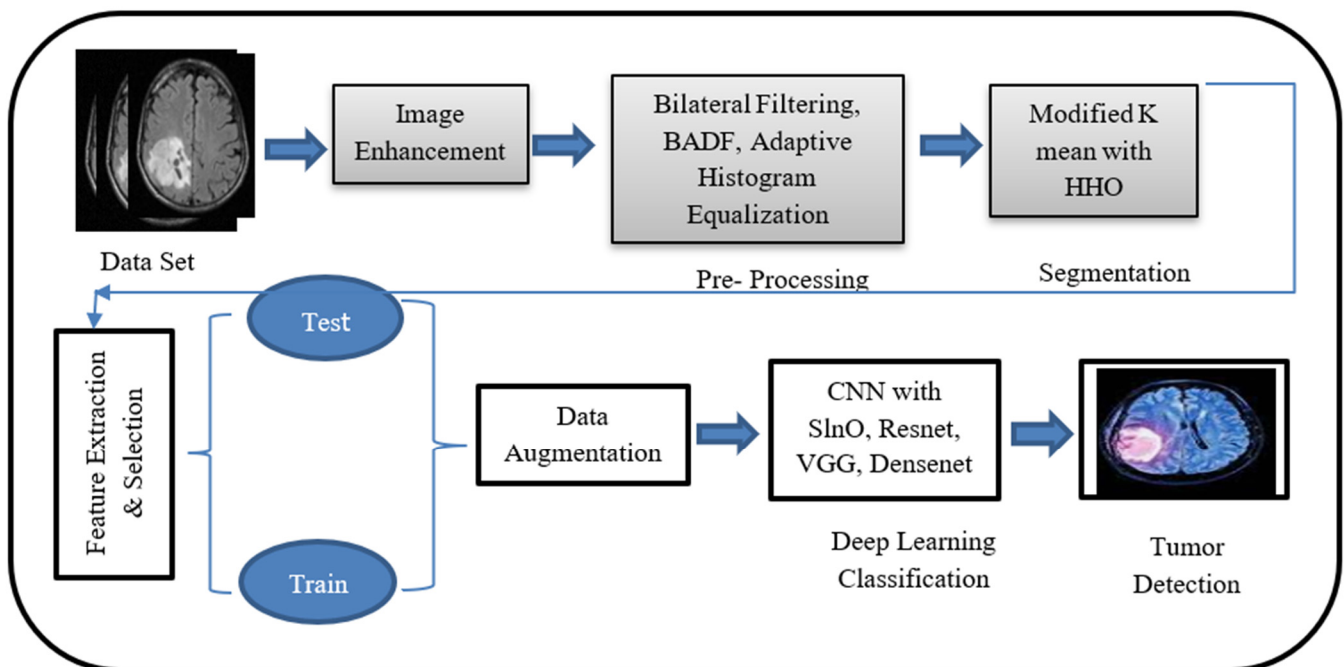


Figure 1. Schematic representation of the proposed research study.

2.1. Image Acquisition

The BRATS 2015 data set was employed in this analysis, and it contains 325 high-grade (HGG) and 54 low-grade (LGG) images scanned for meningioma patients. In the 2015 BRATS multimodal MRI dataset, there were also four scanning sequences available for each patient, using weighted T1, gadolinium-enhancing contrast (T1C), weighted T2, and FLAIR. Each subject was scanned with T1. The HGG picture from BRATS 2015 is used in this report. Table 2 shows the most commonly used dataset for brain tumors. Figure 2 gives the most frequently used dataset details, and Table 2 gives the link for the online data repository.

Table 2. Different sources of data and techniques of acquisition.

Sl.No	Dataset Sources	Link (Access on: 6 April 2020)
1	Medical images from BRAINIX	https://www.medicalimages.com/search/brain.html
2	BraTS 2018	https://www.med.upenn.edu/sbia/brats2017/data.html
3	Harvard Medical School website	http://med.harvard.edu/AANLIB/
4	TCGA-GBM, TCGA-LGG	https://wiki.cancerimagingarchive.net/display/Public/TCGA-LGG
5	BraTS-2018 ISLES-2018	http://www.isles-challenge.org/
6	IBSR dataset Cyprus	http://www.medinfo.cs.ucy.ac.cy/
7	BraTS 2015 and BraTS 2017	https://www.med.upenn.edu/sbia/brats2017/data.html
8	Brain web (simulated brain database)	http://brainweb.bic.mni.mcgill.ca/brainweb/
9	Brain MRI	https://figshare.com/articles/dataset/brain_tumor_dataset/1512427
10	Harvard Medical School	http://www.med.harvard.edu/aanlib/home.html

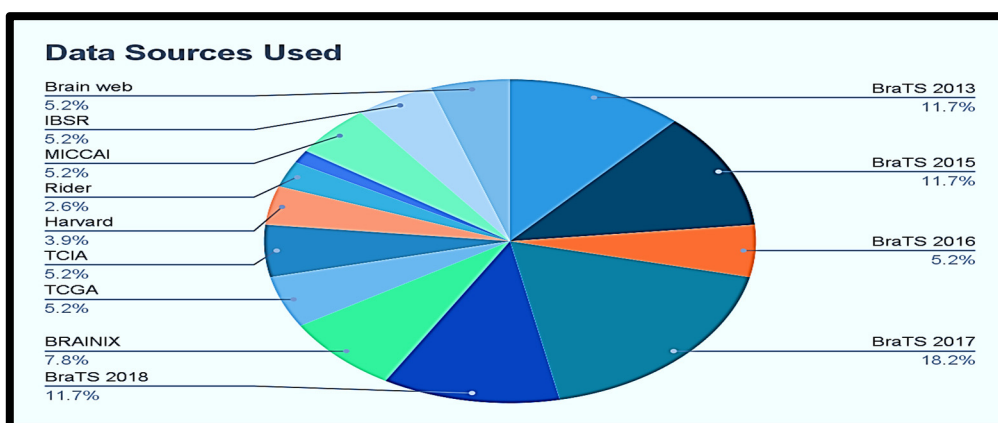


Figure 2. Pictorial representation of online data sources most frequently used by world-wide researchers.

2.2. Preprocessing: Image Enhancement

Preprocessing of the image is done to enhance the image quality. The dynamic range of the selected characteristics is increased so that they may be easily spotted. To extract the relevant information, the MRI images of the brain are first obtained and preprocessed. The quality of photographs might decline during the image production process due to a variety of factors, such as being out of focus, noise, the optical system, distortion, relative motion between the camera and the scene, and so on. A variety of preprocessing approaches is used. The goal of image enhancement is to make information in images more interpretable or to give better input for other automated image processing techniques. Film artefacts such as labels and marks on the MRI image, as well as the high-frequency components, are removed during the enhancing stage. The MRI picture is turned into a standard image without noise, film artefacts, or labels after enhancing techniques are applied. MRI images are frequently contaminated by noise, such as Gaussian and Poisson noise [25]. The great majority of de-noising methods assume white Gaussian noise is additive. Edge-preserving bilateral filter, total variation, and non-local means are some of the strategies meant to remove Gaussian noise. Various image enhancing techniques are used in this study, including bilateral filtering (BF), curvelet transform (CT), adaptive histogram equalization (AHE), and boosted anisotropic diffusion filter (BADF).

- **Bilateral filtering**

Bilateral filtering is a technique for smoothing images while keeping the edges. The use of bilateral filtering has rapidly advanced, and it is now employed in image processing applications such as image de-noise, picture enhancement, and so on [26]. The pixel of each point in conventional low-pass filtering is assumed to be equal to that of the nearby points:

$$k_d^{-1}(x) = \int_{-\infty}^{\infty} f(\delta)c(\delta, x)d\delta \tag{1}$$

where $(c(\delta, x))$ is the arithmetic contact between the region centre x and a nearby point δ . It is possible that both input (f) and output (h) images are multiband. Furthermore,

$$k_d(x) = \int_{-\infty}^{\infty} c(\delta, x)d\delta \tag{2}$$

Nevertheless, the pixels of points at borders are essentially identical to the closing points. Hence, the lines are blurred. This filter blends gray rates based on both numerical proximity and multispectral similarities, thus allowing values that are identical in both domains. Similarly, the filtering range is specified as follows:

$$h(x) = k_r^{-1}(x) = \int_{-\infty}^{\infty} f(\delta)s(f(\delta), f(x))d\delta \tag{3}$$

where $s(f(\delta), f(x))$ computes the photographic similarity between the region center x and a nearby point δ , and the kernel assesses pixel similarity in this situation. In this instance, the ongoing standardization is as follows:

$$k_r^{-1}(x) = \int_{-\infty}^{\infty} s(f(\delta), f(x))d\delta \tag{4}$$

The bilateral filtering is described as given below:

$$h(x) = k^{-1}(x) = \int_{-\infty}^{\infty} f(\delta)c(\delta, x)s(f(\delta), f(x))d\delta \tag{5}$$

where $k(x) = \int_{-\infty}^{\infty} c(\delta, x)f(\delta)d\delta$. Each region combines filtration area, and their distance will be defined as bilateral filtration. It substitutes the pixel value at a related and near-average pixel value. In flat regions, pixel values in a small neighborhood are related to each other, and the bilateral filter effectively acts as a normal domain filter, averaging the weak, inadequately associated variations between the pixel values produced by the noise. Figure 3a depicts the clinical database input, while Figure 3b depicts the bilateral filtering image for all of those inputs.

• **Curvelet Transform**

The curvelet transform is a multi-scale transform that is based on the wavelet transform. The basic elements of the curvelet transform are dimension, location, and orientation. It has excellent orienting characteristics [27]. In comparison to the wavelet transform, the curvelet transform is the best method for removing noise. It accurately depicts curved objects, allowing them to be used in graphical applications [28]. The main applications are edge detection and picture de-noising. The bilateral filtering image for all of those inputs is shown in Figure 3c.

A curvelet coefficient $t(a, b, c)$ is given by the following:

$$t(a, b, c) = \langle f, \varphi_{a,b,c} \rangle \tag{6}$$

where $a = 0, 1, \dots$ —scale limit; $b = 0, 1, \dots$ —orientation limit; $c = (c_1, c_2)$, $c_1, c_2 \in \mathbb{Z}$ —translation limit.

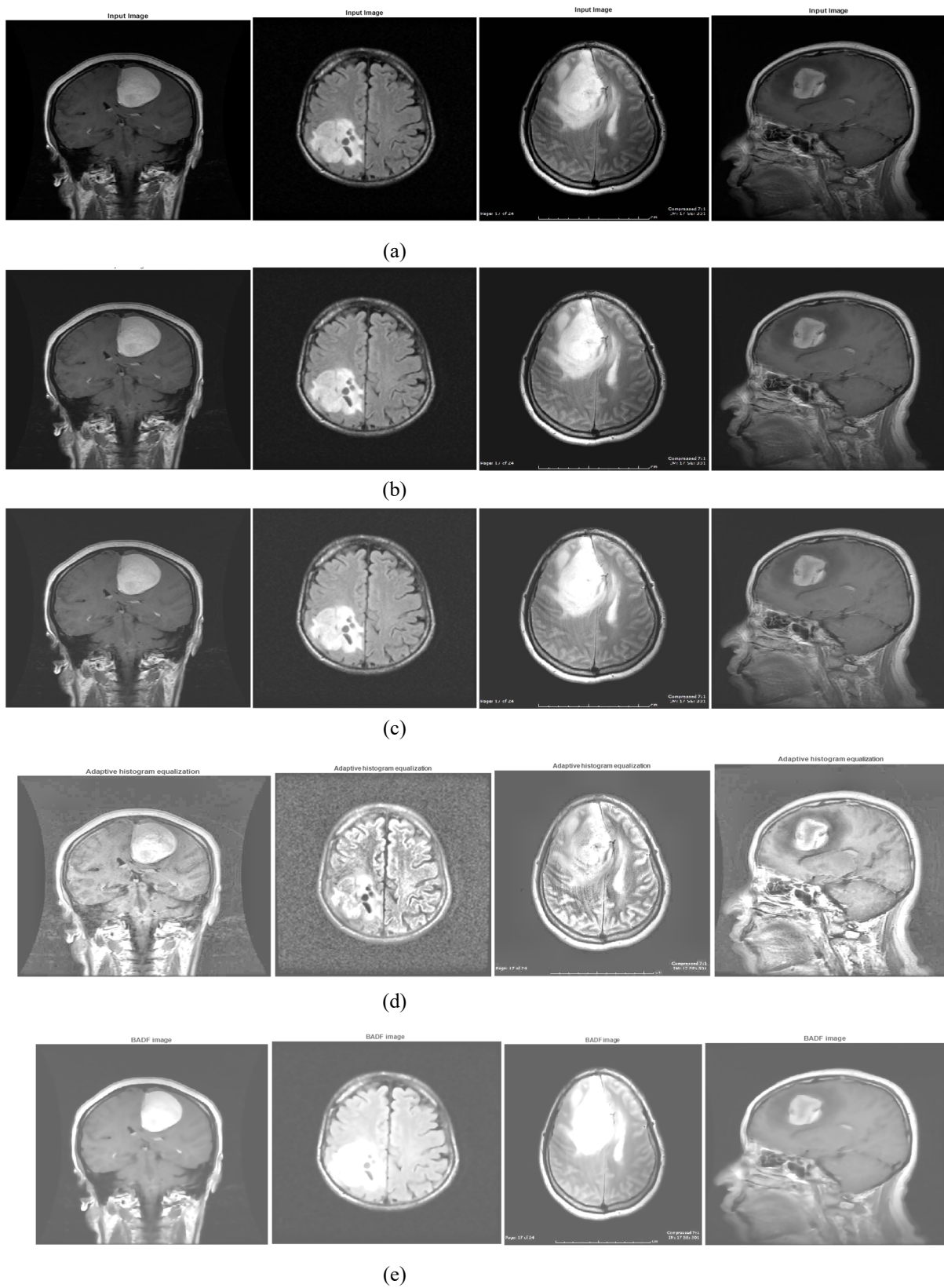


Figure 3. (a) Depiction of three different input image grades. (b) Image of bilateral filtering. The curvelet transform image is shown in (c). (d) Image with an adaptive histogram equalized histogram. (e) Adaptive boosted anisotropic diffusion filter.

A curvelet coefficient is the internal multiplication among elements $f \in K2(M2)$ and curvelet $\varphi_{a,b,c}$, as follows:

$$t(\varphi_{a,b,c}) = \langle f, \varphi_{a,b,c} \rangle \int_{M2} f(X) \overline{\varphi_{A,B,C}(X)} dX = \frac{2}{(2\pi)^2} \int \hat{F}(y) U_a(M_\theta, y) e^{i(X_c^{(A,B)}, y)} dy \quad (7)$$

The curvelets with a scale of 2^{-a} , orientation θ_1 , and position $X_c^{(A,B)} = R_{\theta_1}^{-1}(c1.2^a, c2.2^{-a/2})$ can be represented as:

$$\varphi_{a,b,c} = \varphi_a(R_{\theta_1}(X - X_c^{(A,B)})) \quad (8)$$

where $\theta_1 = 2\pi \cdot 2^{a/2} \cdot l$, with $l = 0, 1 \dots 0 < \theta_1 < 2\pi$, and R_θ being rotation by θ_1 radians.

The following is a description of the curvelet transform algorithm. The photo was taken as I.

- Using additive wavelet transform, the image 'I' is separated into three sub-bands: s_1 , s_2 , and s_3 .
- The tiling procedure is carried out in sub-bands s_1 and s_2 .
- After that, the ridgelet transform is applied to each of the tiling sub bands s_1 and s_2 .

It is mostly concerned with the sharpening of curves. Curvelet transforms are based on partitioning a picture into minute overlapping tiles and then applying a ridgelet transform. This is primarily utilized in the medical field. The curvelet has a number of important qualities, one of which is:

$$L = W^2 \quad (9)$$

where $L = \text{length} = 2^{-A}$, $W = \text{width} = 2^{-(2 \cdot A)}$, and $A = \text{scale}$.

In a tiling procedure, the altered image's sub-bands s_1 and s_2 are separated into overlapping tiles, resulting in a reduced dimension, and then the curving line is changed into a straight line, avoiding edge effects. To approximate the curving line into overlapping tiles, segmentation is used. Discrete segmented tiles are subjected to the ridgelet transform. This ridgelet transform is used to detect shapes. The ridgelet transform calculation is highly complex since it involves many stages, but nowadays it is simple because it does not employ the ridgelet transform, therefore the time complexity is substantially decreased [29].

The waveform $\varphi_a(x)$ is defined as FT $\varphi_a(y) U_a(y)$.

Let the frequency window U_a be defined in the Fourier domain by:

$$U(p, \theta) = 2 - \frac{2a}{4M} (2p) N \frac{(2\frac{a}{2}\theta)}{2\pi} \quad (10)$$

where M is the radial window, N is the angular window, and $a/2$ is the truncated integer of $a/2$.

• **Adaptive Histogram Equalization**

The main purpose of image enhancement is to improve certain characteristics of an image to improve its visual quality. This is a variation on the classic histogram equalization method. It improves visual contrast by altering the image's intensity values. The bilateral filtering image for all of those inputs is shown in Figure 3d. Every bar on the equalized histogram is of the same height, according to the physical meaning of the histogram. That is:

$$p_s(s) ds = p_r(r) dr \quad (11)$$

Assume that $s = T(r)$ is a monotonically increasing function in the interval, and that its inverse, $r = T^{-1}(s)$, is similarly monotonic. We can conclude (12) from (11):

$$p_s(s) = \left[p_r(r) \frac{1}{\frac{ds}{dr}} \right]_{r=T^{-1}(s)} = p_r(r) \frac{1}{p_r(r)} = 1 \quad (12)$$

The traditional histogram equalization algorithms mapping relationship is as follows: the relationship between I (the grey value of the pixel in the original image) and f_i (the grey value of the pixel in the enhanced image) is discrete in discrete conditions.

$$f_i = (m - 1)T(r) = (m - 1) \sum_{k=0}^i \frac{q_k}{Q} \tag{13}$$

where m is the number of gray levels in the original image, q_k denotes the number of pixels in the image with k th gray level, and Q denotes the total number of pixels in the image. If an image comprises n different gray levels, and each grey level's occurrence probability of i th gray level is p_i , the grey level's entropy can be described as follows:

$$e(i) = -p_i \log p_i \tag{14}$$

The entropy of the whole image is as follows:

$$E = \sum_{i=0}^{n-1} e(i) = - \sum_{i=0}^{n-1} p_i \log p_i \tag{15}$$

• **Adaptive Boosted Anisotropic Diffusion filter (BADF)**

The original image undergoes anisotropic diffusion, which is a non-linear and space-variant modification. This filter was first introduced in 1990 (Perona and Malik, 1990). Its main aim is to smooth an image without blurring it while keeping its important details. This filter starts by setting the required iterations number. Figure 3e shows the bilateral filtering image for all those inputs [30]. After creating the diffused image, the proposed BADF adds a partial differential equation (PDE) that enhances the existing anisotropic diffusion filter diffusion, which is missing at the borders and boundaries, and can also be used to smooth the surface. It is a powerful picture enhancing tool that uses unsupervised machine learning. It not only smooths out the image, but it also keeps certain key details like edges and textures [31]. Depending on intensive tests, good results were attained when the iterations number was set to 20. Next, the size of the image is determined so that it can be used when calculating the four nearest-neighbor differences. At this point, the iterative process starts by computing the four nearest-neighbor differences using Equations (16)–(19).

$$\nabla_N I_{i,j} = I_{i-1,j} - I_{i,j} \tag{16}$$

$$\nabla_S I_{i,j} = I_{i+1,j} - I_{i,j} \tag{17}$$

$$\nabla_E I_{i,j} = I_{i,j} - I_{i,j-1} \tag{18}$$

$$\nabla_W I_{i,j} = I_{i,j} - I_{i,j+1} \tag{19}$$

where $I_{i,j}$ is the smoothed image at each iteration, in which at the first iteration, it should fulfil $I_{i,j} = f_{i,j}$, where $f_{i,j}$ represents the input image. i, j are image coordinates; $\nabla_N, \nabla_S, \nabla_E, \nabla_W$ represent the detected differences at the east, west, north, and south, respectively. The image's high-frequency components are held by the identified differences. The high-frequency components in each direction are then attenuated using four conduction operators computed using Equations (20) and (21).

$$g_N = \frac{1}{1 + \left(\frac{\nabla_N I_{i,j}}{k}\right)^2} \tag{20}$$

$$g_S = \frac{1}{1 + \left(\frac{\nabla_S I_{i,j}}{k}\right)^2} \tag{21}$$

$$g_E = \frac{1}{1 + \left(\frac{\nabla_E I_{i,j}}{k}\right)^2} \tag{22}$$

$$g_W = \frac{1}{1 + \left(\frac{\nabla_W I_{i,j}}{k}\right)^2} \tag{23}$$

where K is a scalar that controls the smoothness degree, in that K should satisfy ($K > 1$), as increasing the value of K produces a smoother output. The value of K is set to 7 in the standard anisotropic diffusion filter. In this study, parameter K is computed automatically based on local statistics using Equation (24)

$$k = 2 * \left| \frac{\text{mean}(f_{i,j})}{(0.75 * \sigma(f_{i,j}))} \right| \tag{24}$$

where σ represents the standard deviation. Next, the entire image is smoothed using the divergence, which can be calculated using Equation (10).

$$I_{i,j} = I_{i,j} + 0.25 [(g_N * \nabla_N I_{i,j}) + (g_S * \nabla_S I_{i,j}) + (g_E * \nabla_E I_{i,j}) + (g_W * \nabla_W I_{i,j})] \tag{25}$$

where $I_{i,j}$ is a smoothed image.

Outputs for the different stages of the preprocessing method, such as bilateral filtering, curvelet transform, adaptive histogram equalization, and adaptive boosted anisotropic diffusion filter are clearly depicted in Figure 3. Statistical analysis for both normal and abnormal brain images for the preprocessing method in terms of signal-to-noise ratio, mean square error, and structural similarity index measure for all the methods, such as bilateral filtering, curvelet transform, adaptive histogram equalization, and adaptive boosted anisotropic diffusion filter, is given in Tables 3 and 4.

Table 3. Parameters for measuring preprocessed images. The peak signal-to-noise ratios for various filters in five abnormal and normal images are tabulated.

Sl.no	Peak Signal-to-Noise Ratio							
	BF		CT		AHE		BADF	
	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images
1	32.9735	32.65366	32.16707	33.37299	34.37233	33.28870	40.65289	41.84186
2	32.70056	32.80955	30.69373	34.12845	34.18057	33.80906	39.17468	42.63007
3	33.36560	32.71030	30.97869	33.37448	34.31471	32.95532	39.46799	41.82132
4	33.12597	32.69257	31.57909	34.03285	34.38220	33.62941	40.06339	42.52042
5	32.44922	32.67565	28.05704	34.01673	34.05173	33.58372	36.54817	42.55479

2.3. Segmentation

The objective of the segmentation of the brain tumor image is to identify the tumor location and spread, that is the active tumorous tissue, edema, and necrotic tissue. In this study, modified K-means and HHO algorithms are used for segmentation [32,33]. The modified K-means algorithm helps to enhance the boundary region, thereby being able to remove noise in that area, and it also helps to speed up the K-means clustering method by avoiding the empty clusters in the iterative process. Moreover, a combination of modified K-means with HHO helps to deal with false detection problems at the time of segmentation by increasing the efficiency and accuracy of the system. Segmented results of MRI images using hybrid structure of modified K-means and HHO is given in Figure 4.

Table 4. (a) Gives the mean square error for various filters in five abnormal and normal images. (b) Gives the structural similarity index measure for various filters in five images of abnormal and normal. (c) The signal-to-noise ratios for various filters in five abnormal and normal images are tabulated as follows.

(a)								
Sl.no	Mean Square Error							
	BF		CT		AHE		BADF	
	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images
1	2.8210	3.5294	5.3272	2.6581	2.8417	3.5688	2.6162	1.3523
2	3.2408	2.7786	7.5974	1.9599	2.8566	2.8153	3.7525	0.9324
3	2.1503	3.4102	9.4781	2.5466	2.8734	4.3340	4.6476	1.4266
4	2.3436	3.5045	7.7107	2.2084	2.8364	3.0331	3.9181	1.0669
5	4.4340	3.3660	1189491	1.6722	2.0711	3.0867	5.9822	0.9888

(b)								
Sl.no	Structural Similarity Index Measure							
	BF		CT		AHE		BADF	
	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images
1	0.942638	0.940807	0.912608	0.893660	0.9706	0.9577	0.975781	0.987964
2	0.937472	0.943480	0.87453	0.912453	0.9597	0.9388	0.954054	0.957021
3	0.953744	0.944687	0.932535	0.904303	0.9809	0.9780	0.997524	0.949472
4	0.952283	0.940735	0.898987	0.903351	0.9415	0.9580	0.962998	0.997604
5	0.928484	0.939251	0.854447	0.863688	0.9992	0.9280	0.932256	0.929219

(c)								
Sl.no	Signal-to-Noise Ratio							
	BF		CT		AHE		BADF	
	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images	Abnormal Images	Normal Images
1	32.6522	31.7686	28.9581	31.7647	34.8183	28.6770	32.9704	35.7544
2	31.9678	32.5590	27.3717	33.4359	33.6773	31.5151	31.3449	37.4646
3	33.7956	32.0455	26.4262	31.4675	34.4663	26.9433	30.4710	35.4479
4	33.4235	31.9298	27.2839	32.8714	34.8755	30.4967	31.1575	36.8703
5	30.6925	31.9807	25.3375	33.1414	32.9180	30.2849	29.3719	37.1594

• **Modified K-means clustering with HHO**

The key advantage of employing the method of the modified K-means is that it selects all the scenarios by considering every difference in the data set, instead of arbitrarily choosing a certain group of initial centroids, as is usually done. Modified K-means will spread the instances of the data set to appropriate clusters with the highest precision. In contrast to previous K-means, it is not necessary to determine the number of K clusters, because this is done dynamically in the modified K-means methodology (see Algorithm 1).

Algorithm 1. Modified K-Mean clustering.

Step 1: Calculate distance matrix distance $d_i(x_i, x_j)$,
 where $i = 0, 1, 2, \dots, N - 1$ and $j = 0, 1, 2, \dots, N - 1$.
 Step 2: Calculate the threshold value 'Th' using (26).

$$\sum_{i=0}^{N-1} \frac{\sum_{j=0}^{N-1} \frac{dist(x_i, x_j)}{N}}{N} \quad (26)$$

where 'D' ($d_1, d_2 \dots d_n$) is the data set.

'K' is the cluster. 'n' is the number of data points.

'X' ($X_1, X_2 \dots X_n$) is an instance of the data point. Th is the threshold. 'c' is the cluster center.

Step 3: Find the minimum mean from X_i to X_j using (27).

$$\min \left(\sum_{i=0, j=0}^{N-1} dist(x_i, x_j) \right) \quad (27)$$

Step 4: Determine the minimal mean value index X_i .

Choose the data point X_i as the initial centroid.

Step 5: Until the data points change, the group repeat steps 6 and 7.

Otherwise, go to step 8.

Step 6: Determine the distance between each of the K cluster centers c_j and each data point x_i .
 if ($Th \geq d_i$)

Set nearest cluster with data point x_i .

else

$K = K + 1$;

Step 7: Recalculate the centroid of every cluster.

Step 8: End.

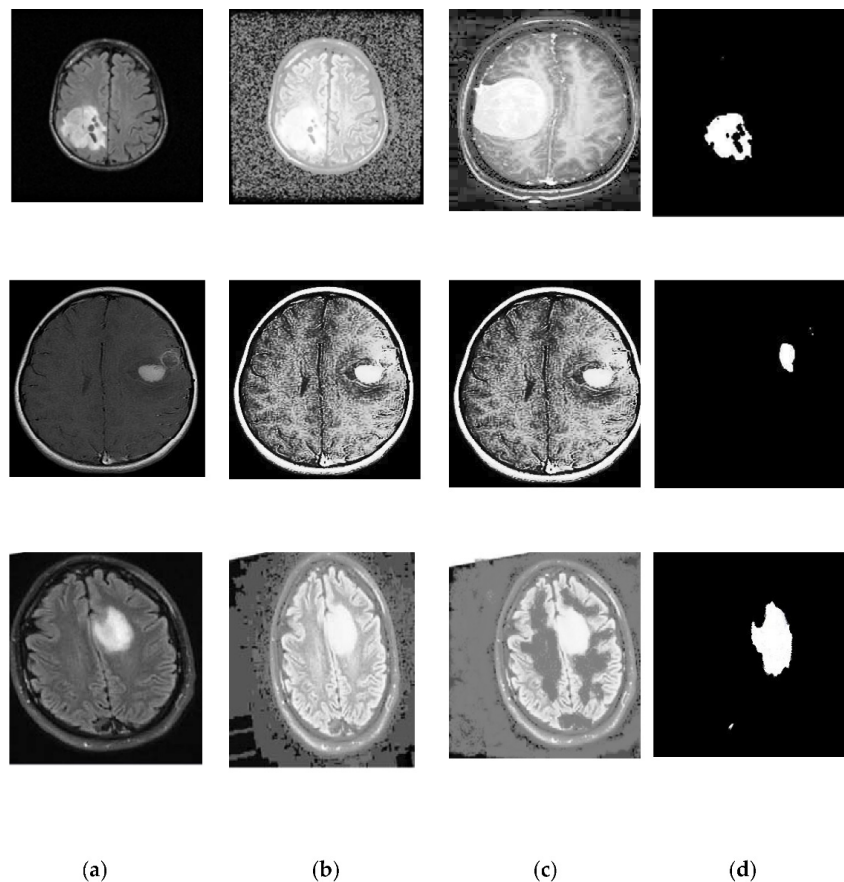


Figure 4. Segmented results of MRI images using hybrid structure of modified K-means and HHO. (a) Input image, (b) preprocessed image, (c) adaptive diffusion filter, (d) segment output after applying hybrid modified K-means and HHO.

• **Harris Hawks Optimization (HHO)**

Mathematically, Harris hawks behaviors are proposing a new metaheuristic stochastic method, the HHO algorithm. The defining characteristic of Harris hawks' activity is that via good coordination, they trace, encircle, and approach the possible prey (rabbits in most cases) and then attack them. In hunting, skillful escape behavior called "surprise pounce" is efficiently performed. The HHO method includes exploration and exploitation phases, similar to other meta-heuristic algorithms.

Harris hawks will be randomly searching for prey during the exploration phase according to the following equation:

$$X(t + 1) = \left\{ \begin{array}{l} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)|q \geq 0.5 \\ (X_{\text{rabbit}}(t) - X_m(t)) - r_3(LB + r_4(UB - LB))q < 0.5 \end{array} \right\} \quad (28)$$

In iteration $(t + 1)$, the hawks are located at $X(t + 1)$, the rabbit (victim) is located at $X_{\text{rabbit}}(t)$, r_1 to r_4 and q are randomly labelled from 0 to 1, $X_{\text{rand}}(t)$ represents a selected hawk at a random position, and X_m is the average location of the current hawk population, as determined by Equation (29):

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (29)$$

where $X_i(t)$ denotes each hawk's location for iteration t , and N indicates the overall hawk number. There is a transitional phase before the exploitation stage once the discovery stage is completed. In this transitional period, the energy of the rabbit should be shaped according to Equation (30):

$$E = 2E_0 \left(1 - \frac{t}{T}\right) \quad (30)$$

where E is the rabbit's escape energy, E_0 is the initial rabbit energy state, and the maximum number of iterations is T . The E_0 value can vary from -1 to 1 dependent on the victim's physical fitness. This indicates that the victim loses energy while E_0 is heading towards -1 , and vice versa. During these last stages of the processing of the algorithm, the Harris hawks abruptly approach their prey. Four tactics of attack are available. Here, r is seen as an escape probability. When $E \geq 0.5$ and $r \geq 0.5$, Harris's hawks slowly utilize a soft besiege technique to surround the prey. The following is the mathematical model:

$$X_i^{t+1} = \Delta X_i^t - E |J X_{\text{prey}} - X_i^t|, \Delta X_i^t = X_{\text{prey}} - X_i^t \quad (31)$$

where $X_i(t + i)$ is the distance between the current individual and the prey, and J indicates the strength of the prey's jumping during the escape, and it takes a random value in the range $[0, 2]$. When $E < 0.5$, $r \geq 0.5$, because of inadequate escape energy, the prey can't escape, and the location of the Harris hawks is written as follows:

$$X_i^{t+1} = X_{\text{prey}} - E |\Delta X_i^t| \quad (32)$$

When the Harris hawks do soft besiege with increasing fast dive tactics to confuse prey when the prey have sufficient energies to effectively escape, that is $E \geq 0.5$, $r < 0.5$. It can be represented as follows:

$$X_i^{t+1} = \left\{ \begin{array}{l} Y = X_{\text{prey}} - E |J X_{\text{prey}} - X_i^t|, \text{ if } f(Y) < f(X_i^t) \\ Z = Y + S \times \text{Levy}(d), \text{ if } (Z) < f(X_i^t) \end{array} \right\} \quad (33)$$

The problem dimension is denoted by d , and S is a $1 \times d$ random vector. The Lévy Flight function is Lévy, When $E < 0.5, r < 0.5$, the prey has insufficient escape energy. The Harris hawks will attack the prey in the following ways:

$$X_i^{t+1} = \left\{ \begin{array}{l} X_{\text{prey}} - E|JX_{\text{prey}} - X_m^t|, \text{ if } f(Y) < f(X_i^t) \\ Z = Y + S \times \text{Levy}(d), \text{ if } f(Z) < f(X_i^t) \end{array} \right\} \quad (34)$$

2.4. Feature Extraction and Optimized Feature Selection

During the feature extraction step, the necessary high-quality data or characteristics of the image are obtained in the form of statistical features, form, color, and texture. The extraction of functions is a key step to reduce the classifier complexity used to categorize the characteristics of an image. In addition, the extraction function is efficiently used to enhance the accuracy of the diagnosis process by selecting major features. This paper covers two new features, the area of the tumor and the dice coefficient index, which are retrieved from the segmented image. There are a variety of existing strategies for image feature extraction; however, in this research, the adaptive swallow swarm optimization (ASSO) is utilized, and SLA12 (sea lion) is used to improve feature selection, ensuring that only important features are used by the classifier to categorize tumor type.

In the ASSO algorithm, the number of swallows is regarded to be the test case characteristic. Basically, with a test suite $SW = (T_1, \dots, T_m)$ and $C = (c_1, \dots, c_m)$. The operating procedure of the SSO algorithm is just like PSO, though it has a few more characteristics, which include the usage of three kinds of particles: (a) aimless particles (o_i), (b) explorer particles (e_i), and (c) leader particles (l_i).

Each one of them has specific duties in the group. The e_i particles are accountable for finding the best optimal hybrid criteria function space. They carry out this searching for an optimal hybrid criterion based on the impact of different parameters (V. Keerthika):

1. Position of the global leader.
2. Position of the local leader.
3. The best individual experience along the path.
4. The earlier path.

The particles utilize the equations below for carrying out a search for an optimal hybrid criterion and proceeding with the test suite process:

$$V_{HL_{i+1}} = V_{HL_i} + \alpha HL_{rand() (e_{best-e_i})} + \beta HL_{rand() (HL_{i-e_i})} \quad (35)$$

The feature selection technique chooses the best features from the features available, reducing the classifier's complexities. The accuracy, specificity, and sensitivity of the classifiers will all improve as a result of this improved feature extraction. The formula for some of the useful aspects in statistics is provided in Table 5.

In this work, Table 6 gives the computational analysis for different feature extraction methods. Table 7 show the comparison statistics of their performance measure of the existing feature extraction method with the proposed ASSO and SLA12. From the table, it is evident that SLA12 exceeds the other optimization approach with the greatest outcomes in Tables 6–8. Tables 7 and 8 show some of the most prominent aspects of the statistical analysis of first- and second-order, respectively.

Table 5. Gives the statistical parameters for calculating feature extraction.

Extracted Features	Equation
Mean	$M = \left(\frac{1}{m \times n}\right) \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y)$
Standard Deviation	$SD(\sigma) = \sqrt{\left(\frac{1}{m \times n}\right) \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (f(x, y) - M)^2}$
Entropy	$E = -\sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) \log_2 f(x, y)$
Skewness	$S_k(X) = \left(\frac{1}{mn}\right) \frac{\sum (f(x,y) - M)^3}{SD^3}$
Kurtosis	$K_{urt}(x) = \left(\frac{1}{mn}\right) \frac{\sum (f(x,y) - M)^4}{SD^4}$
Contrast	$C_{on} = \sum_{k=0}^{m-1} \sum_{y=0}^{n-1} (x - y)^2 f(x + y)$
Correlation	$Corr = \frac{\sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (x,y) f(x,y) - M_x M_y}{\sigma_x \sigma_y}$
Coarseness	$C_{ness} = \frac{1}{2^{m+n}} \sum_{k=0}^{m-1} \sum_{y=0}^{n-1} f(x, y)$

Table 6. Computational time analysis for the different feature extraction methods.

Methods	SSO	ASSO	PSO	GA	GSA	FF	Prop	SLA12
Computation time (s)	4.5100	3.7760	4.3797	4.4388	9.5249	3.9237	3.8177	2.3456

Table 7. First-order and second-order statistical analysis based on performance measure.

Statistical Analysis Based on Performance Measures for First-Order Statistics								
Performance Measure	ASSO	SSO	GA	PSO	FF	GSA	Prop	SLA12
Mean	3.2352	2.6444	2.4671	2.4346	1.5187	1.5275	2.3922	2.3462
Median	2.8012	2.2970	2.3976	2.4619	2.4503	1.4906	2.3592	2.3309
Std Deviation	1.6038	0.5980	0.9927	0.9339	0.9186	0.2522	0.9339	2.3307
Best	2.8901	2.6609	3.6169	3.7684	3.7503	1.7663	3.6555	3.6294
Worst	1.0025	1.2110	1.3448	1.1726	1.0072	1.3558	1.0222	1.0238
Statistical Analysis Based on Performance Measures for Second-Order Statistics								
Performance Measure	ASSO	SSO	PSO	GA	GSA	FF	Prop	SLA12
Entropy	0.9456	3.033	0.4657	2.099	1.122	2.786	0.8765	0.6534
Skewness	0.0055	0.0987	0.0984	0.0105	0.0051	0.0200	0.0789	0.0011
Kurtosis	2.34×10^6	1.34×10^6	1.88×10^6	2.44×10^6	3.34×10^6	1.33×10^5	2.04×10^5	2.74×10^5
Contrast	0.2650	0.6574	0.9845	0.4563	0.7834	0.3452	0.9342	0.3452
Correlation	0.9864	0.9835	0.9823	0.9674	0.8345	0.8345	0.9866	0.9875
Coarseness	8.876	11.897	9.865	10.876	11.9764	12.765	11.998	13.856

Table 8. Comparison of different optimization techniques in feature extraction.

Functions	ASSO	PSO	SSO	GA	GSA	FF	Prop	SLA12
	Avg/Std.	Avg/Std.	Avg/Std.	Avg/Std.	Avg/Std.	Avg/Std.	Avg/Std.	Avg/Std.
F1	28.3976/323.729	38.9854/372.9322	-12,389.05/ 382.570263	29.3456/ 321.821	-6981.15/848.8447	-14,219.09/ 472.66037	37.9524 /371.9542	25.23801/ 295.86792
F2	47.6299/23.4238	12.3789/42.4231	3.78×10^{15} / 1.44×10^{14}	46.0247/ 24.2408	32.30133/ 8.73574	2.66×10^{15} / 2.43×10^{14}	12.3578/ 42.3214	4.12021/ 1.90716
F3	5.119×10^5 / 3.244×10^5	4.78×10^5 / 2.04×10^5	4.32×10^{15} / 2.37×10^{15}	5.012×10^5 / 3.024×10^5	2.85×10^5 / 1.81×10^5	5.00×10^{15} / 2.94×10^{15}	4.89×10^5 / 2.02×10^5	6.42×10^{17} / 6.31×10^{15}
F4	1.9688/2.8532	1.2776/1.9821	0.00285/0.00761	1.9574/2.3541	0.00992/0.01116	0.00496/0.00876	1.2783/1.9821	0.07942/1.99659
F5	1.6734/2.5237	1.2731/1.6231	-1.00040/0.00190	1.9872/2.3457	3.81×10^{11} / 3.86×10^{11}	0.00067/0.00299	1.2701/1.6248	0.98765/0.10078
F6	2.1748/2.8945	1.3472/1.4837	0.00037/0.00046	2.1348/2.7321	0.00366/0.02005	0.00287/0.00396	1.3472/1.4215	0.98937/0.42215
F7	1.3337/0.9877	1.0367/11.9801	1.00641/0.13622	1.0227/0.9867	1.03113/0.18147	1.02163/0.14584	1.0367/12.318	1.06157/15.13597
F8	0.3199/0.8199	0.9812/1.4783	0.00052/0.00023	0.3156/0.8179	0.00056/0.00024	0.00061/0.00044	0.9043/1.3782	0.28946/0.71108

2.5. Classification

The classification method is designed to classify all pixels into one of several classes within digital images. Multi-spectral data are usually utilized to classify, and, in effect, a numerical foundation for categorization is employed as the spectral pattern in the data for every pixel. The image grading aims to detect and show the characteristics in the image about the item which these characteristics represent truly at the ground as a distinctive grey level (or color). The grading of images refers to image labelling in one of several established classifications. In this study, the selected feature is supplied to the classifier for training purposes following optimum function selection. We use CNN classifiers together with many models, such as ResNet50, VGG16, DenseNet and CNN, in our research with optimal sea lion optimization [34].

- **Convolutional Neural Network (CNN)**

The proposed system uses CNN-based artificial segmentation methods to explore smaller 3×3 kernels. The tiny kernels help the architecture become deeper by employing fewer network weights. To achieve functionality maps, a coevolutionary layer is used to convert a sign or an image using kernels. The weight of a kernel thereby ties a unit to the previous layer in a feature map [35,36]. The kernel weights are reversed during the training phase to enhance particular input features. Convolution layers weigh less than dense FC layers because kernels are shared across all units with the same features, making CNN training easier and lowering the chance of overriding. Moreover, because the same kernel across the whole image is combined, the same feature is identified regardless of the location, resulting in invariance translation [4]. The neighborhood information, which is a helpful source of context information, is taken into consideration by employing kernels. The output of every brain unit is normally non-linear. Table 9 shows the complete CNN network. The framework on multigrade tumor segmentation and classification using the CNN network is shown in Figures 5 and 6 gives the diagrammatic representation on cascaded CNN architecture on brain tumor segmentation.

Table 9. Overall network of CNN.

Column 1	Column 2	Column 3	Column 4	Column 5
Structure	Current layer	Type	Output size	Previous layer
Conv_1	Conv_1_1	Conv2d	$H * W * 64$	Input
	Conv_1_2	Conv2d	$H * W * 64$	Conv_1_1
Conv_2	Conv_2_1	Maxpool+Conv2d	$(H/2) * (W/2) * 128$	Conv_1_2
	Conv_2_2	Conv2d	$(H/2) * (W/2) * 128$	Conv_2_1
Conv_3	Conv_3_1	Maxpool+Conv2d	$(H/4) * (W/4) * 256$	Conv_2_2
	Conv_3_2	Conv2d	$(H/4) * (W/4) * 256$	Conv_3_1
	Conv_3_3	Conv2d	$(H/4) * (W/4) * 256$	Conv_3_2
Conv_4	Conv_4_1	Maxpool+Conv2d	$(H/8) * (W/8) * 512$	Conv_3_3
	Conv_4_2	Conv2d	$(H/8) * (W/8) * 512$	Conv_4_1
	Conv_4_3	Conv2d	$(H/8) * (W/8) * 512$	Conv_4_2
Conv_5	Conv_5_1	Multi-scale fusion	$(H/8) * (W/8) * 256$	Conv_4_3
Upsmaple1	Upsmaple1	Upsample	$(H/4) * (W/4) * 256$	Conv_5_1
Conv_6	Conv6_1	Multi-scale fusion	$(H/4) * (W/4) * 128$	Upsmaple1
Upsmaple2	Upsmaple2	Upsample	$(H/2) * (W/2) * 128$	Conv_6_1
Conv_7	Conv_7_1	Multi-scale fusion	$(H/2) * (W/2) * 2$	UpsmapleZ
Avgpool	Avgpool	GlobalAveragePool	2	Conv_7_1
Output	Output	Pixel-wise product	$(H/2) * (W/2) * 1$	Conv_7_1, Avgpool

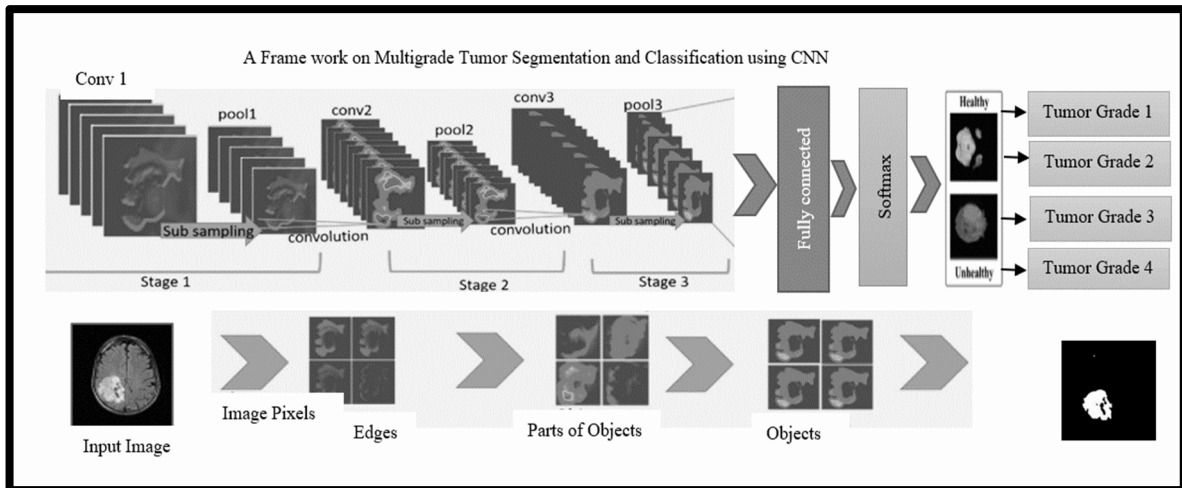


Figure 5. The framework on multigrade tumor segmentation and classification using the CNN network.

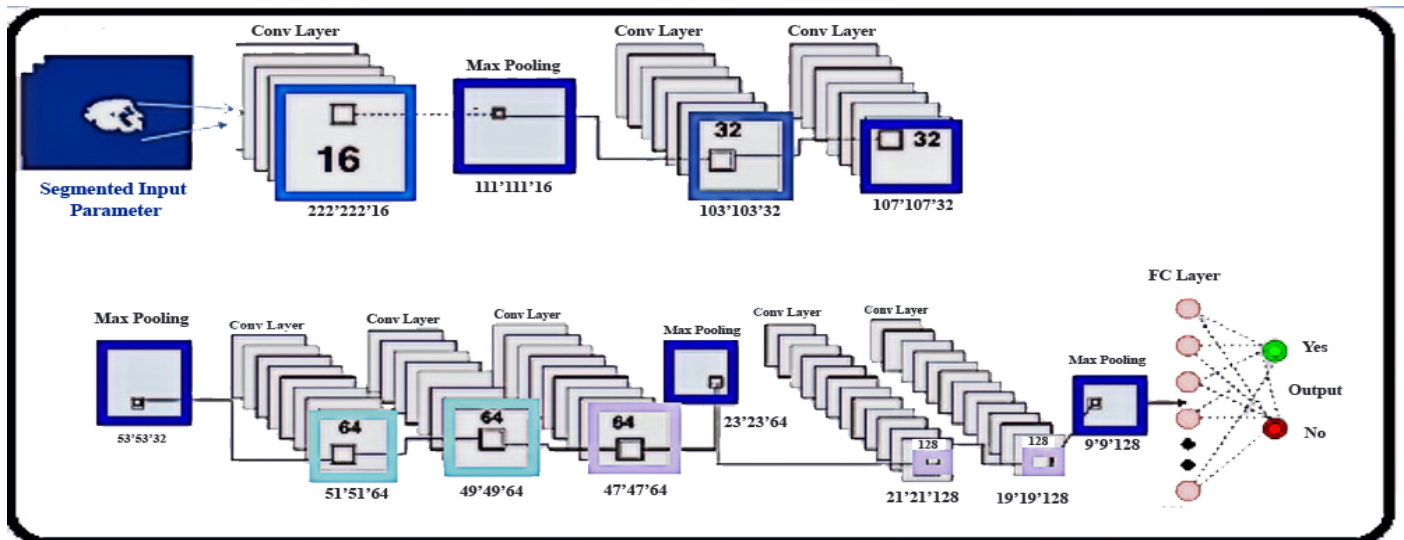


Figure 6. Diagrammatic representation on cascaded CNN architecture on brain tumor segmentation.

The retrieved characteristics become abstract with increasing depth if we stack numerous coevolutionary levels. The initial layers improve characteristics such as edges grouped as motifs, components, or objects in the succeeding layers [13,37].

In the context of CNN, the following ideas are essential:

- *Initialization:* Convergence must be achieved. The initialization of Xavier is used [38]. This maintains activation and gradient-regulated levels, otherwise gradients that are back-propagated can burst or disappear.
- *Activation Function:* Non-linear modification of the data is done through activation functions. Rectifier linear units are defined as follows:

$$f(x) = \max(0, x) \tag{36}$$

Better outcomes than traditional sigmoid and hyperbolic tangent functions were discovered to be achieved, and training accelerated. Imposing a constant 0 can nevertheless interfere with the gradient flow and consequent weight adjustment. These constraints are

addressed using a version dubbed the leaky rectified linear unit (LReLU); the negative half of the function has a modest slope. This function is defined as follows:

$$f(x) = (0, x) + \alpha \min(0, x) \quad (37)$$

where α is the parameter indicating leakiness. SoftMax is used in the final FC layer.

- *Pooling and Regularization:* This mixes the feature maps with spatially close-by features. This arrangement of potentially redundant parts makes the image concise and invariant for slight changes such as minor details. Max-pooling or medium swimming is more common to link features.
- The overfitting is used to minimize. The levels of FC here utilize Dropout. It eliminates nodes with probability from the network in every training stage. This drives all FC layer nodes to learn a better representation of the data, thereby avoiding co-adaptation of nodes. Every node is used during testing. Dropout may be regarded as an ensemble of many networks and bags, since a piece of the training data is training each network.
- *Data Augmentation:* This may be used to develop training sets and eliminate overfitting. It may be utilized to increase the exercise size and to minimize overfitting. Since the center voxel is used for the patch class, the increase in data was limited to rotational operations. Some writers additionally look at translations of images, although this might lead to a misclass being attributed to the patch for segmentation [39]. We extended the data set while training by rotating the original patch and producing new patches. We have used numerous angles of 90 in our proposal to explored other alternatives.
- *Loss Function:* During the training, this is the function that will be lowered. The cross-entropy categorical approach was employed.

$$H = - \sum_{j \in \text{voxels}} \sum_{k \in \text{classes}} c_{j,k} \log(\hat{c}_{j,k}) \quad (38)$$

where the probabilistic predictions (after the SoftMax) are represented by \hat{c} , while the target is represented by c .

The architecture and training for our CNN are discussed in the following parts.

- *Architecture:* In intra-tumoral structures, brain tumors show great diversity, which is a segmentation challenge. We developed the CNN and optimized the transformation of intensity normalization for each grade of tumor to reduce such complexity.

The loss function must be reduced to train the CNN, although it is not particularly linear. Harris hawks is used as an improvement approach that takes steps in line with the negative gradient toward the local minimum level. In low-curvature locations, though, it can be lenient. We also use the accelerated momentum from Nesterov to accelerate the process in these regions. The velocity remained constant, and after each epoch, the learning rate was reduced linearly. We see an era as a whole pass of the training samples.

- **Hybrid CNN With Sea Lion Optimization algorithm**

By estimating the likelihood of each image voxel belonging to the target, the tumor or lesion part is isolated from the background. There are four input parts and convolution parts in the CNN network. The layer processes the incoming image so that the desired image patches are produced. The section convolution processes the designed image patches in which multifaceted filters are operated and feature maps are issued. The completely linked layer grouping all the maps is also available. A prediction value for classifying each voxel image is assessed, and a segmentation map is shown. The classification section helps to give each voxel a probability. The SoftMax function is used to convert maps into category probabilities in most convolutional neural networks [40]. The model is developed with Kera's. Selected 33×33 patches of MRI scans were used to grade the center-pixel of the four-layer sequential model [41]. There are four channels for each input, one of them being a channel for each image sequence, and the network may learn the relative pixel intensities

of each class. Here the classification architectures Resnet, VGG, and DenseNet have been used in this proposed system [42–44].

- **VGG**

The VGG16 and VGG19 architectures are built on and characterized by the same models. They have employed 3×3 coevolutionary filters layered in greater depth on each other and reduce the volume by maximum pooling. At last, the SoftMax classification system, which was built based on their 2014 ImageNet submission and which achieved first and second places in the localization and classification track, is complete with the three fully connected (FC) layers, two of 4096 and one of 1000 neurons (the one displays class scores).

- **DenseNet**

DenseNet's dense block consists of convolution and non-linear layers. Some optimizing methods such as drop-out and batch normalization are also applied. In addition, outputs from the preceding layers are combined, rather than the summation in the compact block given by DenseNet. Assume that the form of an image (28, 28, 3) in the input is the RGB color space in three images. First, we spread the image across the first N channels and get the image (28, 28, N). Every successive overlapping layer generates k features, with the same height and width. Different optimization strategies are applied in deep neural networks by modifying parameters like weights and learning rates to decrease loss and improve precision. The sea lion optimization (SLnO) was used in this section [8].

- **ResNet**

A bottleneck design is utilized for deeper networks such as ResNet50, ResNet152, and so on. Input images of any height or width can be taken by the network as numerous channel widths of 32 and 3. The input size is considered $224 \times 224 \times 3$ for explanatory purposes. The initial convolution and max-pooling of all ResNet architectures are achieved using 7×7 and 3×3 kernel sizes, respectively. Three layers are layered over each residual function F. The three layers are 1×1 , 3×3 , and 1×1 convolution. The 1×1 convolution layer reduces and restores dimensions afterward. The 3-to- 33×3 layer is left as a minor input/output bottleneck. The network also contains an average pooling layer and a fully linked layer of 1000 neurons (ImageNet class output).

- **Sea Lion Optimization**

Sea lions are clever animals [45]; the key characteristic of sea lions is their quick reaction to the movement of fish. They also have amazing senses that allow them to detect fish prey even in the dark depths. Their eyes point to the prey; they can focus swiftly on the prey. Specifically, they may extensively enlarge their pupils to give their eyes a clear underwater vision with enough light. However, in a cloudy atmosphere, occasionally the vision is not clear enough. Therefore, the most important feature of sea lions is their high-sensitivity whiskers. These whiskers aid them in spotting the location of prey. If the prey swims, a wake or waves are left behind. Sea lions are able to track fish using their whiskers. The following are the primary steps in the hunting process of sea lions: (i) chasing prey using their whiskers, (ii) calling for other subgroup members, (iii) encircling and trapping the prey, (iv) attacking the prey. The SLnO method assumes that either the optimum or closest answer is the target prey. This conduct is characterized mathematically by Equation (39).

$$\underline{Dist} = \left| \underline{2B.P}(t) - \underline{SL}(t) \right| \quad (39)$$

Distance and $SL(t)$ are the position vectors between the sea lion and the target lion $P(t)$. The distance between the target lion is represented by Dist. The present iteration is known as t , and \underline{B} as $[0, 1]$ is a random vector that multiplies by 2 to increase the scope for finding an optimal or almost ideal solution for search agents.

$$\underline{SL}(t + 1) = \underline{P}(t) - \underline{Dist.c} \quad (40)$$

where $(t + 1)$ refers to the succeeding iteration, and c has fallen linearly between 2 and 0 throughout all iterations, as the leader of the sea lion must travel around the target

prey. When a sea lion spots a victim, the rest of the group forms a ring around it and attacks [29,42]. Equations (41)–(43) are used to represent this behavior:

$$SP_{leader} = |(V_1(1 + V_2))/V_2| \tag{41}$$

$$V_1 = \sin\theta \tag{42}$$

$$V_2 = \sin\varnothing \tag{43}$$

where SP_{leader} denotes the sea lion leader’s sound speed, and V_1 and V_2 denote sound speed in water and air, respectively.

The $\sin\theta$ and $\sin\varnothing$ represent the speed of sound reflected in a medium that is same as of air. The speed of sound refracted into the sea for calling members in underwater is given as.

$$SL(t + 1) = |P(t) - SL(t)| \cdot \cos(2\pi m) + P(t) \tag{44}$$

where SL denotes the distance between the one searching (sea lion) and the best answer (target prey), $| \cdot |$ denotes an actual absolute value, and m denotes a random count between -1 and 1 . Every sea lion goes in a circular path around the prey (bait ball) to begin looking for food in a bait ball’s periphery. As a result, $\cos(2m)$ is used mathematically to describe this occurrence. If c is more than one, the global search agent will run the SLnO algorithm to find the best global solution. Equations (45) and (46) propose this.

$$Dist = |2B \cdot SL_{rnd}(t) - SL(t)| \tag{45}$$

$$SL(t + 1) = SL_{rnd}(t) - Dist \cdot c \tag{46}$$

where SL_{rnd} is a random sea lion taken from the present population. Random solutions are the first element of the proposed SLnO algorithm. Every search agent moves to a new site at random or based on the best solution. For both the exploration as well as exploitation stages, parameter (C) is minimized between 2 and 0 throughout iterations. If the $|C|$ value is more than one, the search agent is chosen at random. If $|C|$ is less than one, search agents are likely to be upgrading their sites. Finally, the SLnO algorithm is completed with the fulfillment of a final requirement (see Algorithm 2).

Table 10 shows the time to calculate different optimization strategies, and Table 10 gives statistical analysis of different optimized procedures with regard to different classifiers. Table 11 gives the evaluation measures used in the classification frameworks and Table 13 gives the Performance value of classifiers. From f1–f15 in Table 13, the advantages and shortcomings of the algorithm tested in these benchmarks to acquire a deeper knowledge and insight can be seen. F1–f15 has multimodal functions.

Table 10. Analysis on computational time for different classifiers.

Methods	ANN	SVM	PNN	CNN	Sea Lion Optimization with CNN		
					ResNet	VGG	DenseNet
Computational Time (min)	3.458	4.632	4.237	4.112	2.8701	2.2700	2.2345

Figure 7 gives the analytical results for EFA for sea lion, are mainly focused on the parameters such as LA, LT, OS, OP, EI, and MC, and the observed results are obtained based on path estimates, Chi-square, DE, P, CMIN/DF, GFI, SRMR, NFI, RFI, IFI, TLI, CFI, RMSEA, and RMR.

Algorithm 2. Sea Lion Optimization.

Initialize the population
 Choose $S_{L_{rnd}}$
 Output from the segmentation result is taken as input (i)
 Apply fitness function to every search agent.
 If the search agent does not belong to any S_{leader}
 If (i < max iter)
 Compute S_{leader} as per the Equation (43)
 If ($S_{P_{leader}} < 0.25$)
 If ($H < 1$)
 Using Equation (41), update the search agent's position
 Else
 Randomly choose a search agent S_{rnd} .
 Using Equation (46), update the current position of the search agent
 End if
 Else
 Using Equation (45), update the current position of the search agent
 End if
 Go to first if the condition is met
 Else
 For each search agent, compute the fitness function.
 S should be updated to reflect the optimal solution.
 Return S , the best solution
 End if
 End if
 Stop
 Best solution ($S_{P_{leader}}$) is obtained as the output for training the classifiers.

Table 11. Evaluation measures used in the classification frameworks.

Performance Measure	Formulae
Sensitivity	$\frac{TP}{TP+FN} \times 100\%$
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN} \times 100\%$
Precision	$\frac{TP}{TP+FP} \times 100\%$
F-measure	$\frac{2 * Precision * Recall}{Precision + Recall}$
Jaccard Index	$J(P, Q) = \frac{S(P \cap Q)}{S(P \cup Q)}$
Dice Overlap Index (DOI)	$D(P, Q) = 2X \frac{P \cap Q}{P+Q}$
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN} \times 100\%$
MCC	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(FN+TN)(TN+FP)(TN+FN)}}$

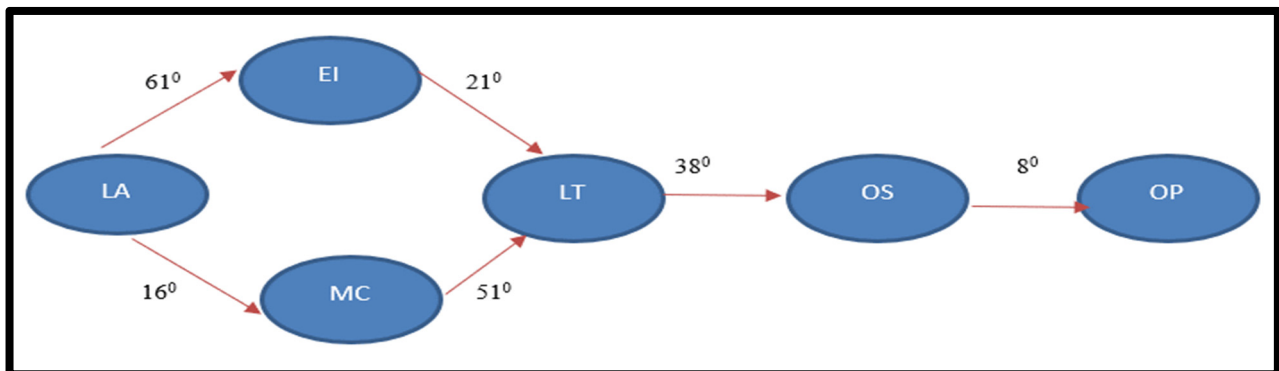


Figure 7. Exploratory factor analysis for sea lion optimization algorithm.

On multiple MRI images, the suggested technique was checked and simulated in Python using Kera's and TensorFlow. For this experiment, the BRATS dataset was employed. Preprocessing methods are used to eliminate noise and improve image quality in the sample images (Figure 3). Following this process, the tumor area is segmented from images and the segmented findings obtained are compared to the tumor area reported and segmentation limitations in the data set, and are summarized in Table 10.

The performance of existing and proposed techniques is evaluated in this section based on the segmentation and classification of the tumors in MRI images. Table 11 shows the suggested technique compared with current approaches in terms of tumor accuracy, precision, sensitivity, segmentation F measurement, Jaccard index, Dice overlapping index, accuracy, and MCC (Matthews' correlation coefficient). The comparison is based on a set of 50 MRI scans, which are divided into two groups: normal and abnormal. Of these 50 MRIs, 25 random MRIs were selected for use in the training process, with 15 normal and 10 anomalous images; the remaining 15 MRI images included 9 ordinary ones and 16 anomalous MRIs. In addition, using SLA12 as an optimization technique potentially boosted the efficiency of every network that we used for analysis. Tables 12 and 13 provide the resulting performance values.

Finally, a classification procedure is applied to obtain the severity stages and better accuracy. By analyzing Table 4, the performance measures such as PSNR, MSE, and SSIM are analyzed for preprocessed brain cancer MRI images, and the values are obtained. As can be seen from Table 12, the experimental result proves that by using the optimized technique, the overall accuracy is high. From our experiments on different images, it is observed that the proposed method works well in both cases when the objects in the image are indistinct and distinct from the background. Each image MCC is high for CNN with the HHO model. This denotes that the proposed study has achieved a better result. Figure 8 gives the experimental output image for the input image for different stages such as preprocessing, segmentation and for sealion optimization.

Table 12. Performance evaluation of proposed segmentation models.

	METRICS	K-Means	Modified K-Means	HHO	CNN with HHO
Image 1	Dice Coefficient	0.9818	0.9863	0.9658	0.9868
	Jaccard Coefficient	0.9642	0.9729	0.9256	0.9744
	MCC	0.9814	0.9859	0.9103	0.9863
	Accuracy	0.9991	0.9992	0.9980	0.9995
	Sensitivity	0.9456	0.9247	0.8934	0.9127
	Specificity	0.9346	0.8923	0.9012	0.9983
Image 2	Dice Coefficient	0.7846	0.9133	0.9254	0.9442
	Jaccard Coefficient	0.6456	0.8405	0.9675	0.8945
	MCC	0.7647	0.9163	0.9324	0.9447
	Accuracy	0.9989	0.9991	0.8912	0.9995
	Sensitivity	0.9823	0.9214	0.9823	0.9847
	Specificity	0.9234	0.7865	0.9167	0.9882

Table 12. *Cont.*

	METRICS	K-Means	Modified K-Means	HHO	CNN with HHO
Image 3	Dice Coefficient	0.7964	0.8714	0.9217	0.9327
	Jaccard Coefficient	0.6617	0.7721	0.9357	0.8732
	MCC	0.8117	0.8775	0.9812	0.9346
	Accuracy(%)	0.9957	0.9972	0.9843	0.9973
	Sensitivity	0.8967	0.9876	0.9673	0.9657
	Specificity	0.8912	0.8679	0.9124	0.9878
Image 4	Dice Coefficient	0.9088	0.9616	0.9452	0.9882
	Jaccard Coefficient	0.8328	0.9260	0.95632	0.9774
	MCC	0.9114	0.9617	0.94352	0.9884
	accuracy	0.9974	0.9986	0.9214	0.9987
	Sensitivity	0.8967	0.9231	0.9563	0.9972
	Specificity	0.8999	0.9213	0.8941	0.9657
Image 5	Dice Coefficient	0.9818	0.9863	0.8912	0.9867
	Jaccard Coefficient	0.9642	0.9729	0.9812	0.9745
	MCC	0.9814	0.9859	0.9123	0.9862
	accuracy	0.9991	0.9992	0.8892	0.9997
	Sensitivity	0.9124	0.9823	0.8824	0.9878
	Specificity	0.9342	0.9912	0.8723	0.9342
Image 6	Dice Coefficient	0.7846	0.9133	0.9126	0.9443
	Jaccard Coefficient	0.6456	0.8405	0.8909	0.8941
	MCC	0.7647	0.9163	0.9012	0.9443
	accuracy	0.9989	0.9991	0.9143	0.9995
	Sensitivity	0.9112	0.9982	0.8872	0.9126
	Specificity	0.8923	0.9365	0.8891	0.9457

Table 13. Performance value of classifiers.

Metrics	CNN	Sea Lion Optimization with CNN		
		ResNet	VGG	DenseNet
Accuracy	0.9678	0.9651	0.9756	0.9867
Error	0.0445	0.0453	0.0367	0.0289
Sensitivity	0.9667	0.9654	0.9773	0.9830
Specificity	0.9833	0.96535	0.9733	0.9834
Precision	0.9602	0.96026	0.9744	0.97612
F1-Score	0.9698	0.96002	0.9722	0.978

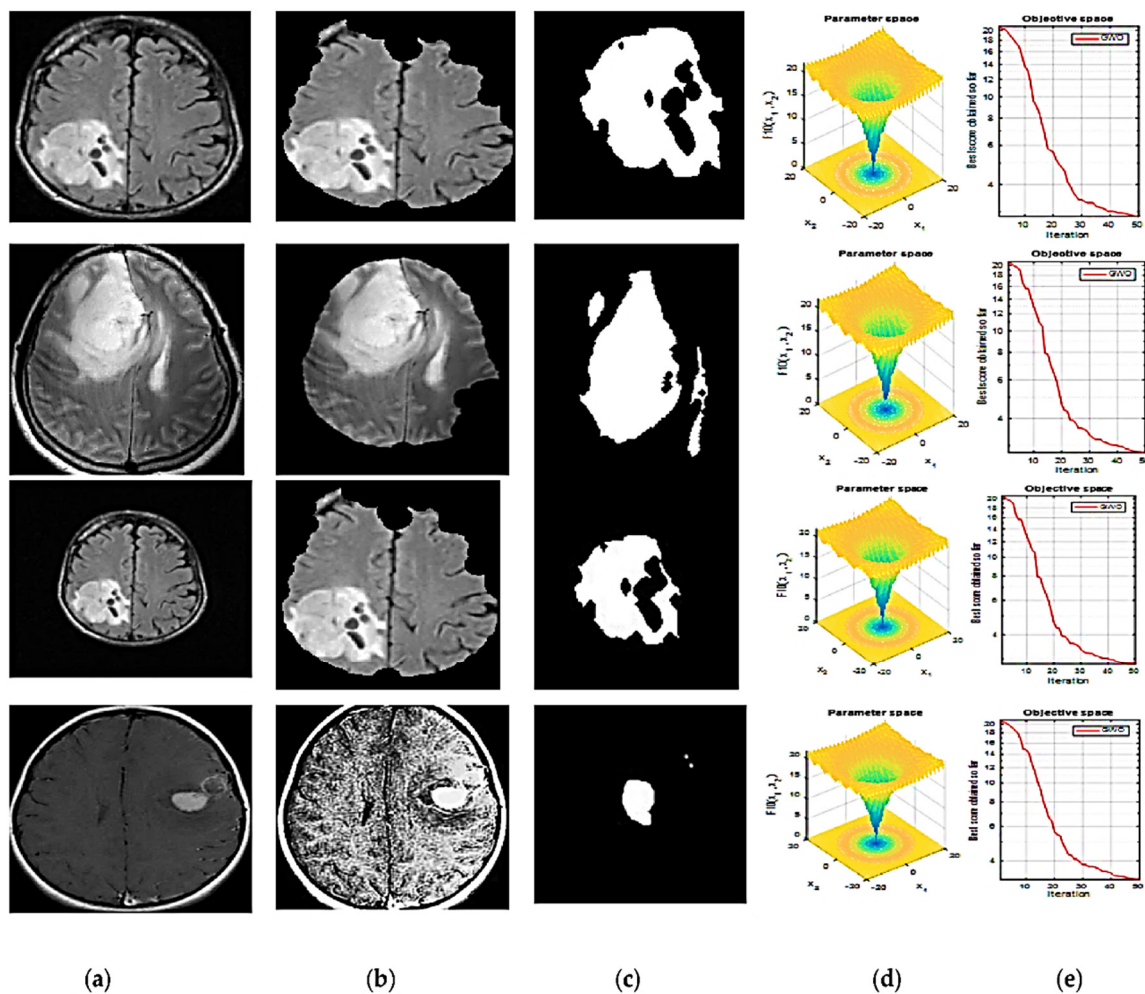


Figure 8. Experimental output for the given input brain image (a) gives the input meningioma brain image (b)preprocessed brain image after applying CLAHE and anisotropic diffusion filter (c) gives the segmented output of the image (d) parameter space output after applying the sea lion optimization and figure (e) gives the object space output.

The experimental findings highlight the efficiency of our suggested technique to address more segmentation challenges by boosting segmentation quality and precision within a minimum period. Table 13 shows a greater precision, sensitivity, and specificity of the sea-lion-based CNN classifiers; the error value is less than the suggested classifier. The results show that the suggested method performs quite well. The implementation of the proposed tumor detection approach has been shown to enhance the effectiveness and precision of clinical practice. The time analyses of several deep models are shown in Table 10. Table 13 represents the overall model analysis with regard to computational time. The precision of different models is illustrated in Figure 9a,b which exhibits the error of different models in which DenseNet has less error and more accuracy. Figure 10a illustrates the sensitivity and specificity of different models, and Figure 10b displays the F1 score with regard to different models in which DenseNet offers greater performance than the alternatives with sea lion optimization. Figure 11a shows several models without sea lion, and Figure 11b shows different models with sea lion, in which this calculation is reduced on a particular level when optimized.



Figure 9. (a) Confusion matrix without optimization. (b) Confusion matrix with optimization.

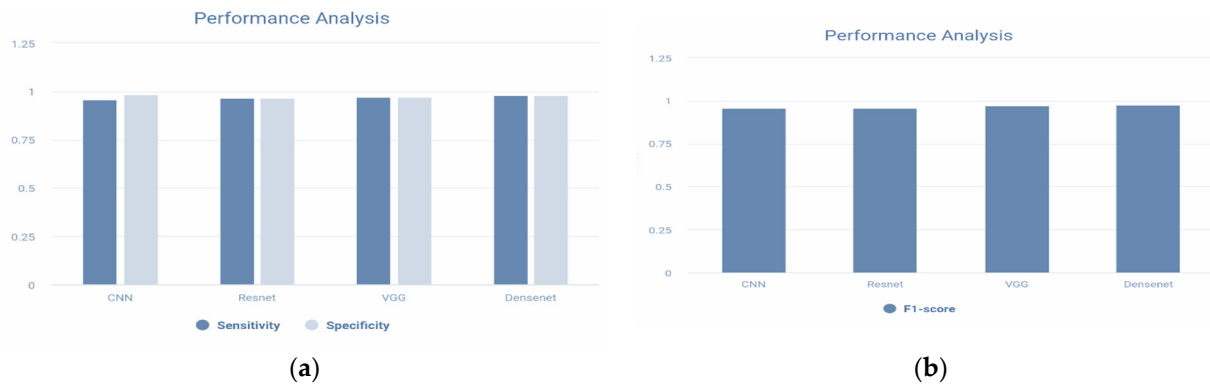


Figure 10. (a) Sensitivity, specificity vs. models. (b) F1-score vs. models.

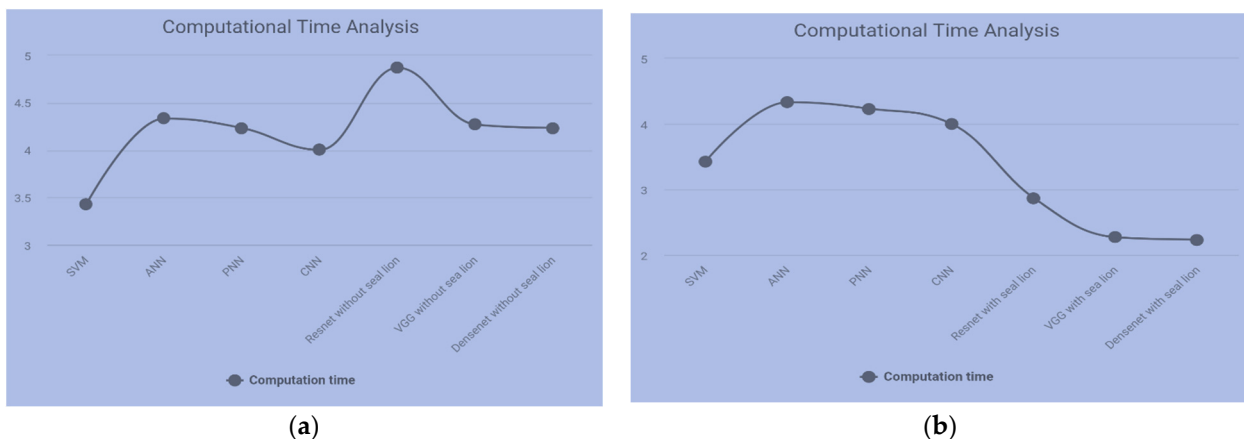


Figure 11. (a) Computation time without sea lion. (b) Computation time with sea lion.

2.6. Comparative Analysis with the State-of-the-Art Methods

We compare our findings to those of other researchers, including the existing system to demonstrate the validity of our method [45–47]. To classify and segment, the CNN algorithm with deep learning method is employed in our proposed system. Table 14 shows the comparative study outcome. The Table 14 provides the comparison findings with the current outcomes of the proposal. The analysis achieves the highest accuracy in our suggested technique.

Table 14. Comparison with state-of-the-art methods for our proposed system.

Methods	Accuracy (%)	Sensitivity (%)	Precision (%)	Specificity (%)
Wu et al. (2020) [42]	96.1	95.4	96.91	96.7
Lodh et al. (2020) [37]	96.2	96.39	96.391	96.6
Badza et al. (2020) [46]	95	95.05	94.24	94.2
Veeramuthu et al. (2015) [47]	96.5	96	96.78	95
Seetha et al. (2018) [48]	96.08	97.01	97	96.23
Krishnakumar et al. (2021) [49]	85	85.034	85.89	84.3
Jun et al. (2020) [50]	89.59	90.77	85.97	88.4
Proposed Method				
<i>SL+resnet 50</i>	96.5	96.54	96.02	96.54
<i>SL+VGG</i>	97	97.73	96.026	97.33
<i>SL+DenseNet</i>	98	98.30	97.44	98.33

3. Conclusions

For the diagnosis of brain tumors, an accurate diagnosis with no nonconformities is essential. Any misinterpretation has irreversible consequences. Every year, brain tumor cases are increasing, and this has also brought about a certain increase in strain on medical staff in this area. The segmentation of brain tumor images requires a precise and practical strategy to swiftly resolve growing demand. This document provides optimal models based on this background to increase segmentation and cluster accuracy and get automatic identification without operator intervention. It consists of CNN and a serially connected integrated sea lion optimization technique. In comparison to typical classifiers, this classification provides excellent precision. Different architectures and detailed hyper-parameter tweaking may be designed to further increase the model's efficiency. In brain tumor patients, the system presented can play an important predictive role in tumor identification. Furthermore, in our proposed method, preprocessing is operated on a small region rather than the entire image, which in turn increases the computational cost and increases the complexity of the circuit. The complexity in the circuit increases the time consumption, so further concentration has been given to rectify this drawback. Moreover, in other clinical fields relating to health imaging, in particular with regards to lung cancer and breast cancer, whose worldwide mortality is quite high, our presented method can play a beneficial role in the early diagnosis of high-risk diseases.

Author Contributions: All authors contributed for the preparation of manuscript. A.S. developed the concept and Manuscript, A.A. supervised and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Acknowledgments: This research has been supported by The Analytical Center for the Government of the Russian Federation (Agreement No. 70-2021-00143 dd. 01.11.2021, IGK 000000D730321P5Q0002).

Conflicts of Interest: The authors declare no conflict of interest.

Futuristic Way to Multidisciplinary Research: As this paper mentions the story of brain tumors over the past decade, this will be potentially very helpful for the healthcare community (neurosurgeons) and research community for better integration and development of models for even more accurate results in terms of theoretical and practical aspects.

Bibliometric Inferences: This paper gives much a brief account of various publications put forwarded by many active researchers. In addition, it provides a brief review of various researchers who bring great innovations to the diagnosis of brain tumors. Therefore, for this, an essential tool is required for getting a clear view of this area.

Available Datasets: This paper brings over two different datasets that are available for training and testing, such as the DL model, and will provide support for the research community for better integrated models.

Limitations and Future Enhancements: The most advanced technology (i.e., DL) has certain limitation for the healthcare community that need to be addressed and taken care of. The limitations involve the enormous volume of data that needs to be fed in, the quality of data for better processing, dealing with natural data, and the perpetuality of the domain, and these can be addressed by using other features, encrypting every piece of data, ensuring the privacy of the system that is about to develop, and keeping DL to the right path by incorporating the knowledge of experts. Creating DL systems that are time sensitive will lead to a better DL model that could effectively support for the healthcare system.

References

1. El-Dahshan, E.S.A.; Mohsen, H.M.; Revett, K.; Salem, A.B.M. Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm. *Exp. Syst. Appl.* **2014**, *41*, 5526–5545. [[CrossRef](#)]
2. Saba, T.; Mohamed, A.S.; El-Affendi, M.; Amin, J.; Sharif, M. Brain tumor detection using fusion of hand crafted and deep learning features. *Cogn. Syst. Res.* **2020**, *59*, 221–230. [[CrossRef](#)]
3. Logeswari, T.; Karnan, M. An improved implementation of brain tumor detection using segmentation based on the hierarchicself-organizing map. *Int. J. Comput. Theory Eng.* **2010**, *2*, 591. [[CrossRef](#)]
4. Kabir-Anaraki, A.; Ayati, M.; Kazemi, F. Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. *Biocybern. Biomed. Eng.* **2019**, *39*, 63–74. [[CrossRef](#)]
5. Jana, G.C.; Swetapadma, A.; Pattnaik, P.K. Enhancing the performance of motor imagery classification to design a robust brain computer interface using feed forward back-propagation neural network. *Ain Shams Eng. J.* **2018**, *9*, 2871–2878. [[CrossRef](#)]
6. Sidhu, H.S.; Benigno, S.; Ganeshan, B.; Dikaios, N.; Johnston, E.W.; Allen, C.; Kirkham, A.; Groves, A.M.; Ahmed, H.U.; Emberton, M.; et al. Textural analysis of multiparametric MRI detects transition zone prostate cancer. *Eur. Radiol.* **2017**, *27*, 2348–2358. [[CrossRef](#)] [[PubMed](#)]
7. Afshar, P.; Mohammadi, A.; Plataniotis, K.N. Brain tumor type classification via capsule networks. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; IEEE: Athens, Greece, 2018; pp. 3129–3133.
8. El-Omari, N.K.T. Sea Lion Optimization Algorithm for Solving the Maximum Flow Problem. *IJCSNS* **2020**, *20*, 30.
9. Kabir, M.A. Automatic brain tumor detection and feature extraction from MRI image. *Sci. World J.* **2020**, *8*, 695–711.
10. Deepak, S.; Ameer, P.M. Brain tumor classification using deep CNN features via transfer learning. *Comput. Biol. Med.* **2019**, *111*, 103345. [[CrossRef](#)] [[PubMed](#)]
11. Banerjee, S.; Mitra, S.; Shankar, B.U. Automated 3D segmentation of brain tumor using visual saliency. *Inf. Sci.* **2018**, *424*, 337–353. [[CrossRef](#)]
12. Arasi, P.R.E.; Suganthi, M. A Clinical Support System for Brain Tumor Classification Using Soft Computing Techniques. *J. Med. Syst.* **2019**, *43*, 1–11. [[CrossRef](#)] [[PubMed](#)]
13. Sathi, K.A.; Islam, M.S. Hybrid Feature Extraction Based Brain Tumor Classification using an Artificial Neural Network. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 30–31 October 2020; pp. 155–160. [[CrossRef](#)]
14. Jiachi, Z.; Shen, X.; Zhuo, T.; Zhou, H. Brain tumor segmentation based on refined fully convolutional neural networks with a hierarchical dice loss. *arXiv* **2017**, arXiv:1712.09093.
15. Gumaei, A.; Hassan, M.M.; Hassan, M.R.; Alelaiwi, A.; Fortino, G. A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. *IEEE Access* **2019**, *7*, 36266–36273. [[CrossRef](#)]
16. Nyoman, A.; Hanif, M.; Hesaputra, S.T.; Handayani, A.; Mengko, T.R. Brain tumor classification using convolutional neural network. In Proceedings of the World Congress on Medical Physics and Biomedical Engineering 2018, Prague, Czech Republic, 3–8 June 2018; Springer: Singapore, 2019; pp. 183–189.
17. Mohan, N. Tumor Detection From Brain MRI Using Modified Sea Lion Optimization Based Kernel Extreme Learning Algorithm. *Int. J. Eng. Trends Technol.* **2020**, *68*, 84–100. [[CrossRef](#)]

18. Roy, S.; Bandyopadhyay, S.K. Detection and Quantification of Brain Tumor from MRI of Brain and its Symmetric Analysis. *J. Inf. Commun. Technol.* **2012**, *2*, 477–483.
19. Mohsen, H.; El-Dahshan, E.-S.A.; El-Horbaty, E.-S.M.; Salem, A.-B.M. Classification using deep learning neural networks for brain tumors. *Future Comput. Inform. J.* **2018**, *3*, 68–71. [[CrossRef](#)]
20. Özyurt, F.; Sert, E.; Avci, E.; Dogantekin, E. Brain tumor detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy. *Measurement* **2019**, *147*, 106830. [[CrossRef](#)]
21. Praveen, G.B.; Agrawal, A. Hybrid approach for brain tumor detection and classification in magnetic resonance images. *Commun. Control Intell. Syst. CCIS* **2015**, *2015*, 162–166. [[CrossRef](#)]
22. Meenakshi, R.; Anandhakumar, P. Brain Tumor Identification in MRI with BPN Classifier and Orthonormal Operators. *Eur. J. Sci. Res.* **2012**, *85*, 559–569.
23. Aswathy, S.U.; Devadhas, G.G.; Kumar, S.S. Brain tumor detection and segmentation using a wrapper based genetic algorithm for optimized feature set. *Clust. Comput.* **2018**, *22*, 13369–13380. [[CrossRef](#)]
24. Sharma, M.; Purohit, G.N.; Mukherjee, S. Information Retrieves from Brain MRI Images for Tumor Detection Using Hybrid Technique K-means and Artificial Neural Network (KMANN). In *Networking Communication and Data Knowledge Engineering*; Springer: Singapore, 2018. [[CrossRef](#)]
25. Asokan, A.; Anitha, J. Adaptive Cuckoo Search based optimal bilateral filtering for denoising of satellite images. *ISA Trans.* **2020**, *100*, 308–321. [[CrossRef](#)] [[PubMed](#)]
26. Rachmad, A.; Chamidah, N.; Rulaningtyas, R. Image Enhancement Sputum Containing Mycobacterium Tuberculosis Using A Spatial Domain Filter. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *546*, 052061. [[CrossRef](#)]
27. Karthik, R.; Menaka, R.; Chellamuthu, C. A comprehensive framework for classification of brain tumour images using SVM and curvelet transform. *Int. J. Biomed. Eng. Technol.* **2015**, *17*, 168. [[CrossRef](#)]
28. Nayak, D.R.; Dash, R.; Majhi, B.; Prasad, V. Automated pathological brain detection system: A fast discrete curvelet transform and probabilistic neural network based approach. *Expert Syst. Appl.* **2017**, *88*, 152–164. [[CrossRef](#)]
29. Bhadauria, H.; Dewal, M. Medical image denoising using adaptive fusion of curvelet transform and total variation. *Comput. Electr. Eng.* **2013**, *39*, 1451–1460. [[CrossRef](#)]
30. Kumar, R.R.; Kumar, A.; Srivastava, S. Anisotropic Diffusion Based Unsharp Masking and Crispening for Denoising and Enhancement of MRI Images. In Proceedings of the 2020 International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), Patna, India, 10–11 July 2020; IEEE: Patna, India, 2020; pp. 1–6.
31. Anoop, B.N.; Joseph, J.; Williams, J.; Jayaraman, S.; Sebastian, A.M.; Sihota, P. A prospective case study of high boost, high frequency emphasis and two-way diffusion filters on MR images of glioblastoma multiforme. *Australas. Phys. Eng. Sci. Med.* **2018**, *41*, 415–427. [[CrossRef](#)] [[PubMed](#)]
32. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
33. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics* **2020**, *9*, 1295. [[CrossRef](#)]
34. Zhou, X.; Tang, Z.; Xu, W.; Meng, F.; Chu, X.; Xin, K.; Fu, G. Deep learning identifies accurate burst locations in water distribution networks. *Water Res.* **2019**, *166*, 115058. [[CrossRef](#)]
35. Yanda, M.; Wei, M.; Gao, D.; Zhao, Y.; Yang, X.; Huang, X.; Zheng, Y. CNN-GCN aggregation enabled boundary regression for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Lima, Peru, 4–8 October 2020; Springer: Cham, Switzerland, 2020; pp. 352–362.
36. Toğaçar, M.; Ergen, B.; Cömert, Z. BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model. *Med. Hypotheses* **2020**, *134*, 109531. [[CrossRef](#)]
37. Lodh, C.C.; Mahanty, C.; Kumar, R.; Mishra, B.K. Brain Tumor Detection and Classification Using Convolutional Neural Network and Deep Neural Network. In Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 13–14 March 2020; IEEE: Gunupur, India, 2020; pp. 1–4.
38. Xavier, G.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
39. Sharif, M.; Amin, J.; Nisar, M.W.; Anjum, M.A.; Muhammad, N.; Shad, S. A unified patch based method for brain tumor detection using features fusion. *Cogn. Syst. Res.* **2020**, *59*, 273–286. [[CrossRef](#)]
40. Badža, M.M.; Barjaktarović, M. Segmentation of Brain Tumors from MRI Images Using Convolutional Autoencoder. *Appl. Sci.* **2021**, *11*, 4317. [[CrossRef](#)]
41. Xing, Y.; Yu, J.; Zhang, F.; Gong, Y. Image Denoising Algorithm Based on Local Adaptive Nonlinear Response Diffusion. *Mater. Sci. Eng.* **2020**, *790*, 012103. [[CrossRef](#)]
42. Wu, W.; Li, D.; Du, J.; Gao, X.; Gu, W.; Zhao, F.; Feng, X.; Yan, H. An Intelligent Diagnosis Method of Brain MRI Tumor Segmentation Using Deep Convolutional Neural Network and SVM Algorithm. *Comput. Math. Methods Med.* **2020**, *2020*, 4916497. [[CrossRef](#)]
43. Rao, B.S. Accurate leukocoria predictor based on deep VGG-net CNN technique. *IET Image Process.* **2020**, *14*, 2241–2248.

44. Naoya, T.; Mitsufuji, Y. D3Net: Densely connected multidilatedDenseNet for music source separation. *arXiv* **2020**, arXiv:2010.01733.
45. Rao, I.V.; Rao, V.M. Massive MIMO perspective: Improved sea lion for optimal antenna selection. *Evol. Intell.* **2020**, *14*, 1831–1845. [[CrossRef](#)]
46. Badža, M.M.; Barjaktarović, M. Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network. *Appl. Sci.* **2020**, *10*, 1999. [[CrossRef](#)]
47. Veeramuthu, A.; Meenakshi, S.; Darsini, V.P. Brain Image Classification using Learning Machine Approach and Brain Structure Analysis. *Procedia Comput. Sci.* **2015**, *50*, 388–394. [[CrossRef](#)]
48. Seetha, J.; Raja, S.S. Brain Tumor Classification Using Convolutional Neural Networks. *Biomed. Pharmacol. J.* **2018**, *11*, 1457–1461. [[CrossRef](#)]
49. Krishnakumar, S.; Manivannan, K. Effective segmentation and classification of brain tumor using rough K means algorithm and multi kernel SVM in MR images. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 6751–6760. [[CrossRef](#)]
50. Jun, W.; Zhang, Q.; Liu, M.; Xiao, Z.; Zhang, F.; Geng, L.; Liu, Y.; Wang, W. Diabetic macular edema grading based on improved Faster R-CNN and MD-ResNet. *Signal Image Video Process.* **2020**, *15*, 743–751.