Multimedia Tools and Applications (2024) 83:69169–69196
https://doi.org/10.1007/s11042-024-18315-8

Design a novel hybrid optimization with tuned deep convolutional neural network classifier for brain tumor segmentation and classification

A. Viswanathan[1] · M. Umamaheswari[1] · Sathya M[2] · S. J. Karthik Deep Yadav[3]

Received: 19 May 2023 / Revised: 13 October 2023 / Accepted: 19 January 2024 /
Published online: 30 January 2024

Abstract
Nowadays, segmentation and classification is most essential process to analyse the brain tumor disease. Moreover, Magnetic Resonance Imaging (MRI) scan images are used to helps the radiologist to diagnosis the tumor region using an efficient medical imaging techniques. In manual, it requires higher time to process each stage and image size is varied due to large amount of dataset. Several computer vision strategies are introduced in the literature for brain tumor segmentation and classification however, due to lower accuracy as well as ineffective decision making failed to provide the enhanced results. Therefore, this article develops the innovative hybrid Aquila coyote optimization algorithm is used to extract important elements that will be used in the classification process. Then, Deep Convolutional Neural Network (DCNN) classifier used as classification task while the weights are being modified and the suggested approach plays a significant role in improving the classification accuracy. With regard to the evaluation metrics, accuracy, sensitivity, and specificity, the suggested model's effectiveness is assessed. The performance is attained to be 97.3017%, 96.8194%, and 96.4079%, individually. This demonstrates how the suggested technique is better to those already in use for the efficient segmentation and categorization of brain tumours.

Keywords Magnetic Resonance Imaging (MRI) · Brain tumor · Segmentation · Classification · optimization

✉ A. Viswanathan
aviswanathan196@gmail.com

[1] School of CSE, Vellore Institute of Technology, Vellore, Tamil Nadu 63214, India

[2] Department of Computer science and engineering, Mount Zion College of engineering and technology, Lena Vilakku, Pudukkottai 622507, India

[3] CSE, Malla Reddy Institute of Technology & Science, Hyderabad, Telangana 500100, India

1 Introduction

Among the most frequent causes of death in both adults and children is a brain tumour. A mass of tissue that develops uncontrollably due to faulty development pathways is called a tumour. When a certain type of cell separates from its usual characteristics and develops abnormally, a brain tumour results. Malignant and non-cancerous cells that grow abnormally inside the skull or in the brain are known as tumours [4]. Cells, the basic units of tissues, are where cancer starts. These tissues constitute the body's organs [1]. Medical image analysis relies heavily on MRI devices. The data produced by different successive pulses in MRI is multidimensional. RI can reveal details about the disease and detect a number of pathologic abnormalities, allowing for a more exact diagnosis [1]. MRI options include flair, T1-weighted MRI, falir with contrast enhancement and T2-weighted MRI. The most challenging aspect in helping to construct a diagnosed system is segmenting the brain tumour MRI [3]. In the process of building an automatic diagnosis system, a novel technique is offered as follows: picture capture, pre-processing, segmentation, post-processing are the new approach used in construction of automatic diagnosis system.

Various segmentation techniques have been utilized in the past, including region expanding technique, thresholding, clustering, edge detection, and so on [2, 4]. In recent years, a radiation oncologist's manual segmentation of the tumor zone has resulted in longer operations, difficulty in segmenting the tumor territory, and maltreatment by professionals, all of which have resulted in issues. The partial volume effect, noise, and bias fields such as smoothly varying intensities inside tissues are all major challenges in segmentation. When tumors are manually segmented, they might vary widely in size, shape, location, and features. Furthermore, the tumor's intensities are comparable to those of normal brain tissue. When a tumor grows, it often deforms the healthy tissues around it, resulting in an abnormal size and shape. For efficient tumor segmentation, an automatic tumor segmentation approach is being developed to overwhelm these limitations.

The objective of this research is to develop a high-accuracy, efficient automatic image processing technique for brain MR images in a tumor segmentation system [2]. It uses a high-precision, robust diagnosis approach to quickly segment tumor masses and improve tumor detection efficiency [5]. Optimization is an important challenge in various functional domains, including production, healthcare distribution, and image processing [12]. For instance, an optimised programmer can benefit from a computer's enormous memory capacity, the velocity of a particular output or input device, or special CPU capabilities [13]. An program must always operate correctly in order to be successful. As a consequence, optimization offers business advantages in addition to being simple and practical for end users. The deployment of computer intelligence depends heavily on optimization algorithms due to the high demand for intelligence products. Potential usefulness in commerce, technology, teaching, and other fields will be significantly impacted [1].

In order to segregate brain tumours from input MRI images, the Aquila coyote method, a hybrid optimization technique, is introduced in this study. The suggested Aquila coyote algorithm is used to best tune the weights of the DCNN classifier, which classifies brain tumours. To improve the effectiveness of the classification process, the segmented image characteristics that are significant statistical features, texture features, including Gray Level Co-occurrence matrix (GLCM) features are extracted. The

suggested classification model is thoroughly explained in this study, and the conclusions allow the model's effectiveness to be confirmed. Additionally, a thorough evaluation of the effectiveness of traditional classification methodologies is given in order to assess the efficiency of the suggested strategy for classifying brain tumours.

The important contribution of the article is:

- The proposed model designs a novel optimization strategy named Aquila coyote algorithm for the brain tumor segmentation process using images of MRI.
- The important characteristics of the image sequences are essential for improving the effectiveness of the suggested brain tumour classification model.
- To verify the efficacy of the suggested strategy, the proposed scheme is compared to the traditional methods in terms of outcomes.

The remainder of the paper is structured as follows: Section 2 surveys various current approaches based on classification and segmentation process of brain tumours as well as the difficulties. Moreover, the proposed model is described in Section 3 and there outcomes are discussed in Section 4. Similarly, the paper is ends in conclusion Section 5.

2 Literature survey

This section discusses the traditional approaches of segmenting and categorising brain tumours as well as the problems with the current model that served as the impetus for the creation of the suggested model.

According on a review of the literature on existing techniques, Shubhangi Nema et al. [4] created the residual cyclic unpaired encoder-decoder network (RescueNet), which employs residual and mirrored principles to improve the ability of automated algorithms to assist radiologists. However, vanishing gradient is the major problem associated with the developed model. Aparna Natarajan and Sathiyasekar Kumarasamy [5] designed an automatic segmentation model using swarm intelligence and machine learning strategies, and are named as fuzzy logic with spiking neuron model (FL-SNM). The method possessed enhanced performance and was highly robust, but the accuracy of the model was very poor. Naresh Shiv For parameter learning across the number of features collected in each of the three methods, such as T1C, T2, and FLAIR, Shivhare and Nitin Kumar [6] utilised an MLP classifier customised utilizing SMO, GWO, or AEFA algorithms. The outcomes are united using bagging approach for enhancing the performance of brain tumor segmentation. However, more time and resources are required to enhance the accuracy of the model.

A CNN-based multi-level brain tumour classification model was created by Muhammad Sajjad et al. [8] and demonstrated improved performance when compared to evaluations conducted using more traditional techniques. The radiologist must consider other factors in addition to classification as malignant or benign before deciding on the course of treatment. S. Deepak and P.M. Ameer [9] created a 3-class categorization module by utilising the deep transfer learning approach to categorise the three main types of brain malignancies—meningioma, glioma, and pituitary tumours. The dataset used to solve the classification task is quite unbalanced, notwithstanding the model's improved classification accuracy. Gopal S. Tandel et al. [10] modeled a transfer-learning-based

AI model using CCN for brain tumor classification. The model led to improved performance, but the problem of overfitting occurs while using the developed model, which is considered as the major drawback of the model. To categorise the various types of brain tumours, ZarNawab Khan Swatiaet al. [11] employed which was before DCNN classification and a block-wise fine-tuning method based on transfer learning. To visually inspect and categorise the images in the Image Net dataset, which is hard, expensive, and time-consuming, radiologists must be available.

2.1 Problem statement

- The difficulties with the current approaches to segmenting and classifying brain tumours are discussed as follows:
- The precise structure requirement is the fundamental challenge with hierarchical FCM methods based on template matching approaches [3].
- To update cluster centers, FCM is often constructed utilizing quantization and aggregation, as well as a weight factor. FCM-based approaches, on the other hand, have the drawback of requiring a specified number of clusters before initiating clustering operations [3].
- Due to privacy concerns and significant costs associated with data gathering, obtaining high-quality labelled medical data is quite challenging. By utilising pre-trained models produced using TL techniques, this restriction can be eliminated [10].
- The automatic algorithms based on CNN and its variants were unable to significantly increase performance [9].
- The segmentation job cannot be completed successfully for single-mode MR images using self-organizing active contour (SOAC) model [5].

3 Proposed method of brain tumor segmentation and classification

With accurate and comprehensive automatic algorithms for the classification of brain tumours, there is a chance of improving diagnosis and treatment approaches. The performance depends on the pathologists' expertise because segmenting a brain tumour manually is time-consuming and error-prone. In order to execute the enhanced segmentation process effectively, it is crucial to establish an automatic segmentation approach. The primary goal of this study is to present a brand-new, finely adjusted segmentation technique for the separation of MRI images that can distinguish between benign and malignant tumours. To remove image artefacts and enhance the image quality, the input is first pre-processed. The segmentation process is then carried out using the suggested Aquila coyote optimization technique. After segmentation, the important characteristics, such as texture features, statistical features, and GLCM features, are extracted. The DCNN classifier conducts the classification process using the concatenation feature vector as its input. By using the suggested Aquila coyote optimization procedure to optimally tune the DCNN classifier's weights, the classifier's efficacy is increased. Figure 1 depicts the suggested model's schematic diagram.

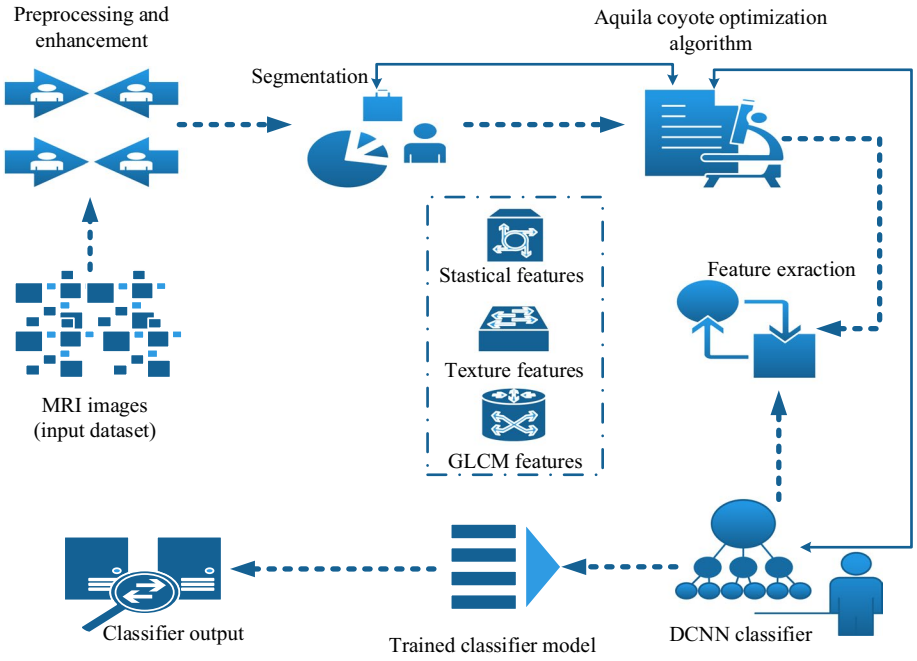

Fig. 1 Schematic diagram of proposed model of brain tumor segmentation and classification

3.1 Input image from input database

The MRI scans from the brain tumour segmentation dataset are used as the source images for analysis. Consider the mathematical formulation of the photos taken from of the input datasets:

$$I = \{D_x\}; (1 \leq x \leq w) \tag{1}$$

Where, w is the input dataset images and D_x represents x^{th} image taken into consideration. After choosing the input data, the image is filtered to determine if it is benign or malignant. The input image's dimension is set to be its size $[224 \times 224 \times 3]$.

3.2 Pre-processing of input image

The unwanted distortions and aberrations, like noises present in the brain tumor image. Such distortions and artefacts must be eliminated prior to the segmentation process in order to generate proper segmentation results. In the proposed work, noises are filtered using the Median Filter. The Median Filter replaces pixel values with the median intensity of nearby pixels, preserving the boundaries of the filtered MRI picture [14]. After pre-processing the image, the ROI is roughly segmented and optimised using the suggested hybrid Aquila coyote segmentation technique.

3.3 Proposed aquila coyote optimization algorithm in brain tumor segmentation

The first step in image analysis and algorithmic is image segmentation, one of the most difficult image processing operations with a significant impact on the final classification outcomes. The technique of separating discrete areas with equivalent pixels is referred to as image segmentation. For a variety of reasons, segmentation is frequently utilised in medical computer vision and image analysis [7]. Some of its applications include the quantification of multiple sclerosis lesions, quantification of coronary borders in angiograms, simulation of surgery, surgical planning, measurement of tumour volume as well as treatment response, automated classification of blood cells, brain development research, tissue sensing on mammography, heart image retrieval from cardiac cineangiograms, and tumour detection. Automatic MR picture segmentation is still a persistent issue. Due to overlapping brain with varied intensities as well as varying intensities of tissues at different regions, it is extremely difficult to properly classify brain tissues automatically. Because of its high functional sensitivity, accurate classification of brain tumours in MR images is essential for both diagnosing and treating them. However, it is significantly more difficult than segmenting natural images. Brain tumours can cover portions of the brain and come in a variety of sizes, shapes, and locations. Potential damage to nearby organs could result from a tumor's growth. Finding problems in medical images is quite challenging due to the irregular structures of the body, particularly if clutter or aberrations are present. The underlying reason why the offered methods are too slow too flexible to be used in practise, despite numerous studies on the issue, is the segmentation problem [3].

3.3.1 Proposed aquila coyote algorithm

The suggested Aquila coyote method is a hybrid meta-heuristics method that combines the traits of different seeker types, including Aquila and coyote seekers [15, 16]. The majority of common optimization techniques were created in response to how hunters prefer to hunt and how they help hunters locate their prey. Similar to how the suggested Aquila coyote optimization approach demonstrates the characteristics of both the coyote and the Aquila hunters in capturing their prey. The suggested Aquila coyote optimization technique, which boasts a dynamic property in tackling the difficulties related to convergence, was developed using the back-and-forth features of the Aquila seekers and the social hierarchy-based characteristic of the coyote seekers. In other words, the dynamic properties of both Aquila seekers and coyote seekers enable the suggested Aquila coyote optimization technique to address the convergence problem. As a result, the suggested Aquila coyote optimization technique is better able to arrive at the overall best solution for practical applications.

The term "optimization" refers to a search process that allows for the evaluation of the best answers to problems that arise in the actual world. Multiple agents may be used in the search process, and it may evolve through iterations based on mathematical formulae. The process of creating a successful optimization algorithm is comparable to the process of creating a self-organizing method [15]. The coyote searchers are taken into account in the suggested research because of traits like social structure and hierarchy and the predominate hunting laws related with them. The traversal velocity features and the to-and-fro hunt properties of the Aquila seekers are key factors in the algorithm's creation. These Aquila seekers are regarded as the most studied species in the entire world because of their courage in the field. In comparison to female Aquila seekers, males Aquila seekers are shown

to be more effective hunters of prey. The Aquila seekers use one of four different hunting techniques, switching between them rapidly and cunningly depending on the situation and the necessity to capture the target. Following is a description of the many hunting techniques used by Aquila seekers:

Process 1: In the first plan, the Aquila seekers ascend from the ground to a very high elevation in order to catch the birds.
Process 2: The second strategy, known as a short slide attack, has the aquila seekers only raise at a low range above the ground to capture their prey.
Process 3: The third hunting tactic is mostly employed to pursue slow-moving prey, as well as the attack carried out in this manner is known as a slow descend attack in which the Aquila seeker lowers to the ground.
Process 4: The fourth technique involves the Aquila seekers walking and dragging the prey outside the coverage region before attacking [15].

The Aquila hunters thus alternate between hunting techniques in order to capture their prey depending on where the prey is located. As a result, the Aquila searchers are regarded as the most knowledgeable and skillful seekers. The algorithm's steps, which were designed based on the traits of Aquila searchers, are as follows:

Step 1: Initialization of Population: The number of Aquila seekers is initially initialised. Each Aquila seeker has a position vector that shows its location at the moment as,

$$P_{a,b} = \left(g_1, g_2, \ldots, g_{n,b}\right); (a = 1, 2, \ldots, n) \tag{2}$$

where, n represents the search space total Aquila seeker and b is the problem dimension, and $g_{a,b}$ is the a^{th} Aquila seeker location in b^{th} dimension. The Aquila seeker each fitness is stored as,

$$F_a^v = \left(F_1, F_2, .., F_n\right) \tag{3}$$

The above equation, which assesses the survival of the fittest according on the fitness measure, stores the corresponding health measure of the each Aquila seeker.

Step 2: Termination criterion: The algorithm is then checked for a method involving, and if one is not found, the initial location of a Aquila seekers is modified as follows:

$$P_{a,b}^{v+1} = rand \times \left(M_b - N_b\right) + N_b \tag{4}$$

Where, $P_{a,b}^{v+1}$ represents a^{th} Aquila seeker position at $(v+1)^{th}$ iteration, *rand* represents random number, N_b represents b^{th} lower bound, and M_b represents b^{th} upper bound of the given problem.

Step 3: Mathematical representation depending on hunting method: The Aquila seekers alternate between the four hunting techniques, which allow them to change between different hunting phases including exploitation and exploration. When the criteria $v \leq \left(\frac{2}{3}\right)B$ met, the exploration procedure is carried out; if not, the exploitation phase is initiated.

(i) Position update using process 1: The first exploration procedure, also known as method 1 of hunting, is used to calculate the location update equation. It is expressed as follows:

$$P_1^{v+1} = P_{best}^{v} \times \left(1 - \frac{v}{B}\right) + \left(P_{mean}^{v} - P_{best}^{v} \times rand\right) \tag{5}$$

Where, P_1^{v+1} represents Aquila seekers position produced by the search method 1, P_{best}^{v} best solution found so far in the v^{th} repetition, and $\left(1 - \frac{v}{B}\right)$ represents the term used for controlling the first exploration process, the term P_{mean}^{v} represents the current solution of location mean value, and *rand* is the random number of range 0 to 1.

(ii) Position update using process 2: According to the features of the Lyra seekers using technique 2 of hunt, also known as the second exploration phase, the location update equation is as follows:

$$P_2^{v+1} = P_{best}^{v} \times levy(r) + P_{Rand}^{v} + (c - d) \times rand \tag{6}$$

Where, P_2^{v+1} is the position of the Aquila seekers position produced by the search method 2, levy(r) represents levy flight distribution, and P_{Rand}^{v} represents a random solution, and x and y represents search spiral shape.

(iii) Position update using process 3: The first exploitation procedure is also known as method 3 of hunting, and it is based on the features of the Aquila seekers. The location update equation is constructed as follows:

$$P_3^{v+1} = \left(P_{best}^{v} - P_{mean}^{v}\right) \times \delta - rand + ((M - N) \times rand + N)\varepsilon \tag{7}$$

Where, P_3^{v+1} represents the Aquila seekers position produced by the search method 3, δ and ε represents the adjustment parameters of exploitation.

(iv) Position update using process 4: The second exploitation procedure is also known as technique 4 of hunting, and it is based on the features of the Aquila seekers. The location update equation is constructed as follows:

$$P_4^{v+1} = FT_q \times P_{best}^{v} - \left(H_1 \times P_4^{v} \times rand\right) - H_2 \times levy(r) + rand \times H_1 \tag{8}$$

$$P_4^{v+1} = P_{AQ}^{v+1} = FT_q \times P_{best}^{v} - \left(H_1 \times P_{Aq}^{v} \times rand\right) - H_2 \times levy(r) + rand \times H_1 \tag{9}$$

Where, P_4^{v+1} represents the Aquila seekers position produced by the search method 4, H_1 signifies the Aquila seekers various movements, H_2 represents the random number that linearly decreases from 2 to 0, and FT_q represents the quality function used for balancing the search strategies. The model created using Aquila seekers uses the equation mentioned above as its standard equation. However, the features of the coyote seekers are combined with the features of the Aquila seekers in order to improve the

performance of the Aquila coyote optimization algorithm which is based on the Aquila seekers. The addition of the coyote seekers aids the optimization algorithm in improving the hunting experience had by the Aquila seekers, speeding up the system's rate of convergence [16]. The answer, emphasising the traits of a coyote seeker, is given as,

$$P_{CO}^{v+1} = P_{CO}^{v} + \tag{10}$$

The Eq. (5) represents the coyote huntees position update expression, where P_{CO}^{v} represents the coyote seeker position at v^{th} iteration, t_1 and t_2 represents the random parameters that changes from 0 to 1. The term y_1 represents the alpha influence and y_2 represents the influence of pack [16].

The above equation becomes,

$$P_{CO}^{v} = P_{CO}^{v-1} + t_1 y_1 + t_2 y_2 \tag{11}$$

And the Eq. (9) becomes,

$$P_{4}^{v+1} = P_{AQ}^{v+1} = FT_q \times P_{best}^{v} - \left(H_1 \times \left[P_{CO}^{v-1} + t_1 y_1 + t_2 y_2\right] \times rand\right) - H_2 \times levy(r) + rand \times H_1 \tag{12}$$

$$P_{AQ}^{v+1} = FT_q \times P_{best}^{v} - \left(H_1 rand\left[P_{CO}^{v-1} + t_1 y_1 + t_2 y_2\right]\right) - H_2 \times levy(r) + rand \times H_1 \tag{13}$$

Equation (13), which depends on parameters derived from the features of Aquila seekers and coyote seekers, is the updated formulation of the Aquila coyote optimization method. This equation helps to segment brain tumours and is also in charge of fine-tuning the DCNN weights to segment MRI images.

Step 4: Reassessment of fitness value: To determine whether the best answers are present, the fitness of each seeker is reevaluated. If the newly discovered solution demonstrates greater fitness, the old solution is replaced with the new one.

Step 5: Termination: After the maximum number of iterations or after the entire problem has been solved, stop the algorithm. Table 1 contains the Aquila coyote optimization algorithm's pseudocode.

3.4 Feature extraction

Extraction of the key characteristics from the segmentation MRI is a crucial step in the proposed approach. The feature vector is extracted from a regular vector using the feature extraction approach. In the suggested classification module, a feature is a standout measurement that is taken from a segment MRI. Selecting the most important features or data is required in order to carry out the categorization procedure. The statistical parameters, Texture feature, and GLCM feature are the features that must be retrieved for the proposed system.

3.4.1 Statistical features

The features of statistical are the significant features generally employed for classification as they reveal even small changes in the MRI image, which improves the accuracy

Table 1 Pseudocode of proposed Aquila coyote optimization algorithm

Sl. No	Pseudo code of proposed Aquila coyote algorithm
1	**Input:** $P_{a,b} = \left(g_1, g_2, ..., g_{n,b}\right), \quad (a = 1,2,...,n)$
2	**Output:** G_{AqCO}^{s+1}
3	Initialize population of Aquila seekers
4	Initialize maximum iteration $Iter_{\max}$, and the parameters δ, and ε.
5	**While**
6	The condition of termination is not met
7	Estimate fitness function for all solutions $F_a^v = (F_1, F_2, .., F_n)$
8	Update the Initial position based on equation (8)
9	**For** $a = 1,2,...,n$,
10	Update the value of P_{mean}^v
11	Update the values of c, d, H_1, H_2, and $levy(r)$
12	**If**
13	$v \le \left(\frac{2}{3}\right)B$
14	**Step 1: First exploration phase**
15	Update the location by means of equation (5)
16	**If**
17	$F(P^{v+1}) < F(P_1^v)$
18	$P^v = P_1^{v+1}$
19	**If**
20	$P_{best}^{v+1} < P_1^{v+1}$
21	$P_{best}^{v+1} = P_1^{v+1}$
22	**End If**
23	**End If**
24	**Else**
25	**Step 2: Second exploration phase**

Table 1 (continued)

26	Update the position by means of equation (6)
27	**If**
28	$F(P^{v+1}) < F(P_2^v)$
29	$P^v = P_2^{v+1}$
30	**If**
31	$P_{best}^{v+1} < P_2^{v+1}$
32	$P_{best}^{v+1} = P_2^{v+1}$
33	**End If**
34	**End If**
35	**Else**
36	**Step 3: First exploitation phase**
37	Update the position by means of equation (7)
38	**If**
39	$F(P^{v+1}) < F(P_3^v)$
40	$P^v = P_3^{v+1}$
41	**If**
42	$P_{best}^{v+1} < P_3^{v+1}$
43	$P_{best}^{v+1} = P_3^{v+1}$
44	**End If**
45	**End If**
46	**Else**
47	**Step 4: Second exploitation phase**
48	Update the position by means of equation (13)
49	**If**
50	$F(P^{v+1}) < F(P_4^v)$
51	$P^v = P_4^{v+1}$
52	**If**
53	$P_{best}^{v+1} < P_4^{v+1}$

Table 1 (continued)

54	$P_{best}^{v+1} = P_4^{v+1}$
55	**End If**
56	**End If**
57	**End If**
58	**End For**
59	**End While**
60	**Return** P_{AQ}^{v+1}

in classification [17]. The statistical features assist the fact that the outcomes obtained are real, and are deliberated as follows,

a) *Mean:* One of the key statistical features, the mean, calculates the average of all MRI data instances relative to all instances. The equation is provided by,

$$Q_{M_N} = \frac{1}{i}\sum_{k=1}^{i} R_k \tag{14}$$

where, i represents the total occurrences and the R_k represents the k^{th} data.

b) *Variance:* Variance is defined as the average of the squared differences between the mean Q_{M_N} and each individual piece of data and is formulated as,

$$Q_{V_R} = \frac{1}{i-1}k\sum_{k=1}^{i}\left(R_k - Q_{M_N}\right)^2 \tag{15}$$

The variance value concentrates even on small variations at various time instances, which contributes much to the enhancement of the prediction process.

c) *Standard deviation:* SD, or standard deviation, is the measure of widely distributed data used to analyse MRI results in relation to the mean.

$$Q_{S_D} = \sqrt{\frac{1}{i-1}\sum_{k=1}^{i}\left(R_k - Q_{M_N}\right)^2} \tag{16}$$

d) *Entropy-based features:* Entropy is a crucial measurement of data because it quantifies the range of uncertainty in stochastic process and assesses the data rate more thoroughly. The relationship of dependence, severance, and independence among the distinct temporal occurrences of the MRI image data is divided by the entropy-based features. The entropy measurement is assessed as,

$$Q_{E_N} = \sum_{k=1}^{C(u)} V_k log V_k \tag{17}$$

where, u indicates the attribute vector, $C(u)$ represents the number of distinctive occurrences in MR image, and V_k indicates the probability of k^{th} occurrence.

e) *Skewness:* The third central moment, which is measured by the factor skewness, is understood to represent the rate of evenness or absence of symmetry. The following is the skewness formula:

$$Q_{S_K} = \sum \frac{\left(R_k - Q_{M_N}\right)^3}{i\alpha^3} \tag{18}$$

f) *Kurtosis:* Kurtosis, which is zero for a Gaussian distribution, is defined as the evaluation of the combined influence of the distribution's tails that correspond to the rest of it. The definition of kurtosis is as follows:

$$Q_{K_R} = \sum \frac{\left(R_k - Q_{M_N}\right)^4}{i\alpha^4} \tag{19}$$

where, R_k indicates the k^{th} value of R, and α represents the sample measure of standard deviation.

3.4.2 Texture feature

An essential aspect of MRI that is typically utilised to rate the image's quality is its texture feature. For the classifying of the various regions of the input image, the measurement of texture variable Q_{T_X} is used [18]. The normal and abnormal tissues in the MRI can be represented using the texture feature. The variations that take place in the texture parameter can quantitatively exhibit the variations in the physical composition of the tissue.

3.4.3 GLCM feature

A GLCM feature is developed by combining the co-occurring grey level intensity measures of an MRI image providing a linear spatial association among two pixels. The spatial association is deliberated using a pair of parameter (ψ, K), where ψ represents the orientation and c indicates the space among two pixels. It is general to describe a set of parameter pairs (ψ, K), and to then join the GLCM matrices, generally to afford rotational invariance with the orientation parameters, characteristically in eight orientations, separated $\pi/4$ radians apart. The count of grey level values S_e indicates the number of unique intensity measures that exist in an image. Basically, an image is scaled from [0, 255] to $[0, S_e]$ before evaluating a GLCM, where S_e is a defined count of grey-levels [19]. The features including energy, contrast, correlation, homogeneity, and dissimilarity are evaluated for the extraction of GLCM features. The term energy is expressed as,

$$Q_{E_N} = \sum_{G=0,T=0}^{S_e-1} L_{G,T}^2 \tag{20}$$

Similarly, the terms contrast is described as,

$$Q_{C_{ON}} = \frac{\sum_{G=0,T=0}^{S_e-1} L_{G,T}(G-T)^2}{\left(S_e - 1\right)^2} \tag{21}$$

The expression for homogeneity is given as,

$$Q_{H_O} = \sum_{G=0,T=0}^{S_e-1} \frac{L_{G,T}}{1+(G-T)^2} \tag{22}$$

The equation for correlation is expressed as,

$$Q_{C_{\mathrm{OR}}} = \frac{\left[\sum_{G=0,T=0}^{S_e-1} L_{G,T}\left[\frac{(G-\mu_x)(T-\mu_y)}{\sqrt{(\omega_G^2)(\omega_T^2)}}\right]\right]+1}{2} \tag{23}$$

The term dissimilarity is formulated as,

$$Q_{D_I} = \frac{\sum_{G=0,T=0}^{S_e-1} L_{G,T}|G-T|}{(S_e-1)} \tag{24}$$

where, the variable $L(G,T)$ indicates the value at $(G,T)^{th}$ position in the grey level co-occurrence matrix.

3.5 Feature concatenation

The significant patient data that has to be examined are included in the feature vector that was produced from the MRI input. The features, that are concatenated to create the feature vector, include statistical features, texture, and GLCM features. Thus, the feature vector is written as,

$$Q_{FINAL} = \{Q_{STATICAL}, Q_{TEXTURE}, Q_{GLCM}\} \tag{25}$$

The feature vectors dimension is, $[1 \times 3000]$ for the prediction of MR image containing brain tumor. The majority of the MRI image's features are represented by their

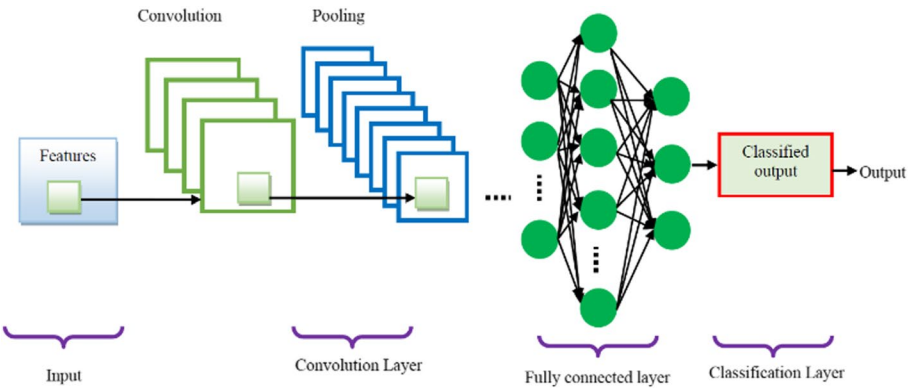

Fig. 2 Architecture of DCNN

characteristics in the feature vector. The selection of the significant features among the whole features is important in the improvement of accuracy in the proposed classifier. Hence, the significant features are selected and concatenated in the proposed module to perform effective classification of brain tumor. The high dimensionality of feature vector deteriorates the training rate of the DCNN classifier, and hence the Principle Component Analysis (PCA) in the proposed research in such a way to reduce the dimension of the feature vector [20]. The feature vector dimension is reduced to $[1 \times 1000]$ through PCA in the proposed research.

3.6 Classification of brain tumor using proposed Aquila coyote tuned DCNN

This component does the categorization of brain tumours utilising photos from the input dataset, involving the DCNN classifier in the classification stage. The benefit of employing DCNN [8] would be that the classifier completes the classification procedure without requiring any human involvement.

3.6.1 Structure of Deep-CNN

The DCNN classifier's configuration consists of a number of layers, each of which carries out a specific function. The convolutional layers create feature maps, while the classification layers produce the final output. The DCNN classifier's structure is shown in Fig. 2.

Input layer The DCNN classifier input layer is initially fed the feature vector produced by the patients' MRIs.

Convolutional layers For comparing the feature maps, convolutional filters are used. Layer-to-layer connections between neurons are made possible by varying loads. The output of these layers is stated mathematically as,

$$\left(W^{p+1}\right) = (W^{p}) + \sum_{f=0}^{j} \sum_{h=0}^{j} (Weight)_f * (Bias)_f \tag{26}$$

where, $*$ represents the operator of convolutional, $\left(W^{p+1}\right)$ represents the convolutional layer fixed feature map. Consider, the convolutional layers weights be, $(Weight)_f$ and $(Bias)_f$ represents the bias, which is tuned with the proposed Aquila Coyote algorithm optimally.

Batch normalization layer This layer is set up between the convolutional layer and the ReLU layer to improve the training properties of the suggested classification model. In order to achieve effective training, this layer controls the gradients and authorizations in the network [8].

ReLU layer The non-parametric ReLU layer, which has no weight or bias, is used to transmit the image features from the convolutional layers.

Max pooling layer The feature maps produced from of the convolution layer in the max—pooling are down sampled to decrease the data as well as the spatial size.

Table 2 Description of layers in DCNN

S.No	CNN layers	Layer type	Layer activations	Total learnables
1	32×32×3 input layer	Input	32×32×3	0
2	32×32×3 Convolution 2D with Stride [22] + Padding [0 0 0 0]	Convolutional	32×32×16 (ReLU)	3984
3	2×2 max pooling with stride [11] + Padding [0 0 0 0]	Max pooling	16×16×16	0
4	25% dropout	dropout	16×16×16	0
5	16×16×16 Convolution 2D with Stride [22]	Convolutional	16×16×16 (ReLU)	28752
6	2×2 max pooling with stride [11]	Max pooling	8×8×16	0
7	25% dropout	dropout	8×8×16	0
8	8×8×16 Convolution 2D with Stride [22]	Convolutional	8×8×36 (ReLU)	46672
9	2×2 max pooling with stride [22]	Max pooling	4×4×36	0
10	15% dropout	dropout	4×4×36	0
11	Flatten and 10% dropout	dropout	[1×1×512]	0
12	Fully connected layer	Fully connected	[1×1×2]	367285
13	Softmax	Softmax	[1×1×2]	0
14	Output with “no brain tumor” and “brain tumor”	Classification	-	0

Fully connected layers The final output generation in the FC layer based on the feature map of convolutional layer is expressed as,

$$A = \chi\left(W^{p+1}\right) \text{with} \left(W^{p+1}\right) = (W^{p}) + \sum_{f=0}^{j} \sum_{h=0}^{j} (Weight)_f * (Bias)_f \tag{27}$$

Using the suggested Aquila Coyote strategy, the DCNN classifier is trained to determine the weights of the network in the best possible way.

Softmax layer To effectively process the classification layer, the FC layer's output is regularised using the softmax activation function.

Classification layer The classification layer in the last layer of the DCNN uses the possibilities produced by the softmax activation to assign a class to each input image. Thus, the input image is categorised in this manner as glioma, malignant tumors, pituitary, and no tumour. Table 2 provides a thorough description of the levels in the DCNN classifier.

4 Result and discussion

This section goes into great depth about the outcomes produced using both the planned Aquila Coyote performance tuning DCNN and the traditional approaches.

4.1 Experimental setup

The analysis is completed using the PYTHON tool, which is installed on a system running Windows 10 64-bit and with 16 GB of RAM.

4.2 Database description

The dataset employed for the implementation is the Brain tumor segmentation (MRI) dataset [21], which includes a total of 4950 files.

4.3 Evaluation metrics

The efficiency of proposed Aquila Coyote optimization-based DCNN is verified using the following measures.

4.3.1 Accuracy

The degree of similarity between the obtained quantity and the real quantity is how accuracy is defined. In mathematics, it is written as,

$$Accuracy = \frac{True\ positive + True\ negative}{real\ positive + real\ negative} \tag{28}$$

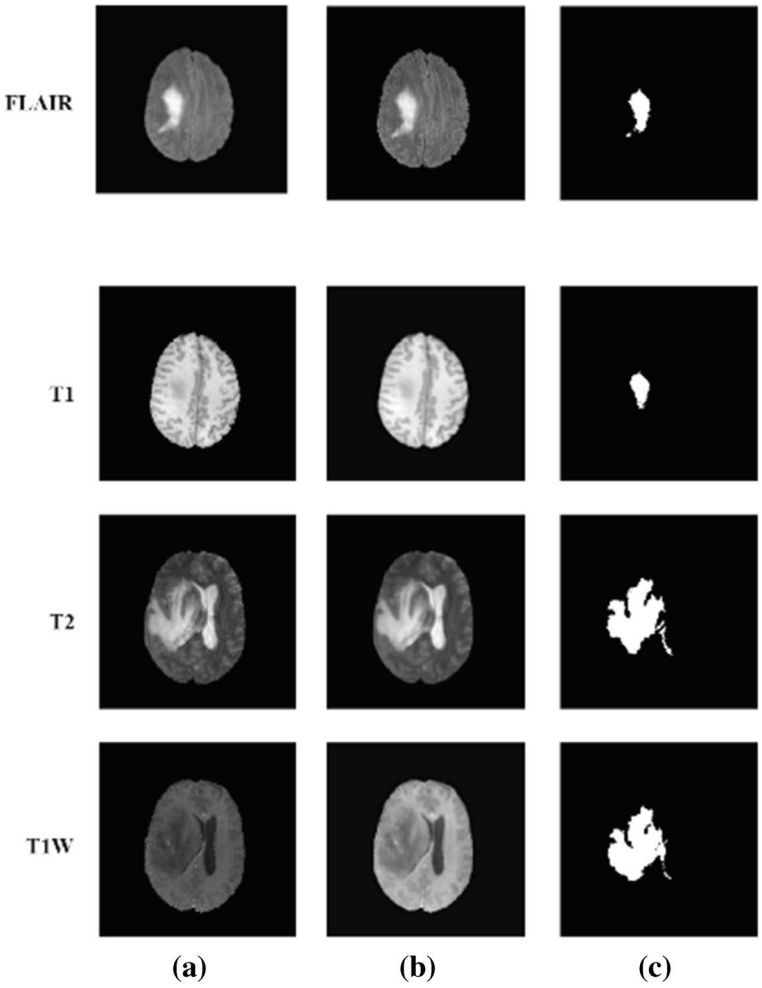

Fig. 3 Experimental results, (**a**) input MRI image, (**b**) Pre-processed image, and (**c**) Segmented image

4.3.2 Sensitivity

The ratio of genuine positives to the total range of actual positive cases is what is referred to as sensitivity.

$$Sensitivity = \left(\frac{True\ positive}{no\ of\ real\ positive\ cases} \right) \tag{29}$$

4.3.3 Specificity

It is defined as the ratio of the number of true negative to the total number of real negative instances in the data.

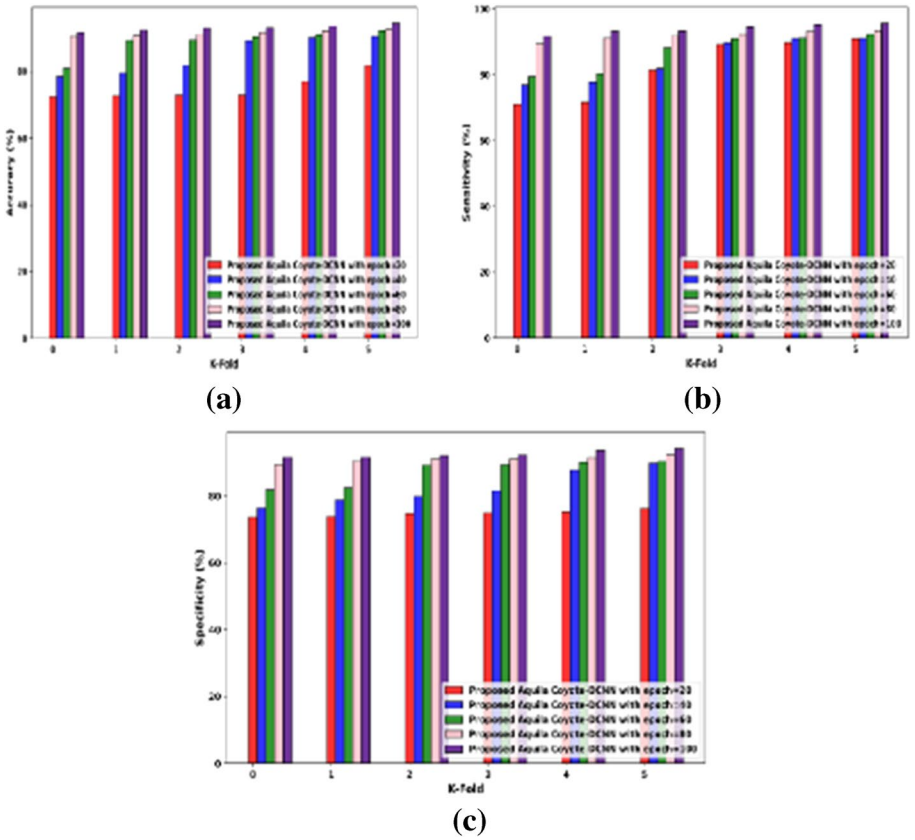

Fig. 4 Performance analysis in terms of training percentage (**a**) accuracy, (**b**) sensitivity, and (**c**) specificity

$$Sensitivity = \left(\frac{True\ negative}{no\ of\ real\ negative\ cases} \right) \tag{30}$$

4.4 Experimental results

This section discusses the experimental results of the suggested brain tumour segmentation approach. The experimental results of the suggested segmentation model are shown in Fig. 3. The FLAIR, T1, T2, and T1W MRI imaging modalities are the four different types. The input MRI picture is shown in Fig. 3(a), and the image is obtained after preprocessing is shown in Fig. 3(b). The segmented picture created by the suggested Aquila coyote optimization technique is shown in Fig. 3(c).

4.5 Performance analysis

This section discusses how the Aquila coyote-DCNN performs in terms of training % and k-fold value.

4.5.1 Performance analysis in terms of training percentage

Figure 4 shows how well the Aquila coyote-DCNN approach performed in terms of training %. The precision of the performance is shown in Fig. 4(a). The accuracy of the suggested approach for the epochs of 20, 40, 60, 80, and 100 is 87.698, 88.8298, 88.9235, 90.413, and 93.1454, respectively, for a training percentage of 50%. The accuracy of the suggested approach for the epochs of 20, 40, 60, 80, and 100 is, respectively, 83.6856, 85.222, 89.945, 90.5895, and 90.5454 for a training percentage of 60%. The accuracy of the Aquila coyote-DCNN approach for the epoch of 20, 40, 60, 80, and 100 is 83.3, 83.4, 88.4432, 90.534, and 92.334, respectively, for a training percentage of 70%. The accuracy of the Aquila coyote-DCNN approach for the epochs of 20, 40, 60, 80, and 100 is 91.3324, 85.724, 90.235, 89.8567, and 92.1576, respectively, for a training percentage of 80%.

The performance created on sensitivity is shown in Fig. 4(b). The Aquila coyote-DCNN method's sensitivity for the epochs of 20, 40, 60, 80, and 100 is 82.1694, 83.3719, 83.655, 84.4517, and 88.8336, respectively, for a training percentage of 50%. The Aquila coyote-DCNN method's sensitivity for the epochs of 20, 40, 60, 80, and 100 is 78.8747, 81.1673, 81.2648, 84.4517, and 85.6468, respectively, for a training percentage of 60%. The Aquila coyote-DCNN method's sensitivity for the epochs of 20, 40, 60, 80, and 100 is 74.8911, 78.078, 85.576, 85.646, and 88.833, respectively, for a training percentage of 70%. The Aquila coyote-DCNN method's sensitivity for the epochs of 20, 40, 60, 80, and 100 is 9176.882, 84.173, 85.049, 87.638, and 91.223, respectively, for a training percentage of 80%.

The performance based on specificity is shown in Fig. 4(c). The specificity of the Aquila coyote-DCNN approach for the epoch of 20, 40, 60, 80, and 100 is 87.8298, 87.9767,

Table 3 Performance analysis in terms of training percentage accuracy, sensitivity, and specificity

Epochs	Accuracy of proposed Aquila coyote-DCNN approach			
	50% of training	60% of training	70% of training	80% of training
20	87.698	83.6856	83.3	91.3324
40	88.8298	85.222	83.4	85.724
60	88.9235	89.945	88.4432	90.235
80	90.413	90.5895	90.534	89.8567
100	93.1454	90.5454	92.334	92.1576
	Sensitivity of proposed Aquila coyote-DCNN approach			
	50% of training	60% of training	70% of training	80% of training
20	82.1694	78.8747	74.8911	9176.882
40	83.3719	81.1673	78.078	84.173
60	83.655	81.2648	85.576	85.049
80	84.4517	84.4517	85.646	87.638
100	88.8336	85.6468	88.833	91.223
	Specificity of proposed Aquila coyote-DCNN approach			
	50% of training	60% of training	70% of training	80% of training
20	87.8298	83.4651	82.6448	83.6702
40	87.9767	84.1618	83.6702	9.2072
60	88.8487	90.0275	87.1564	89.6173
80	91.8732	91.2579	90.0714	91.0529
100	91.8732	93.3087	93.5137	92.9244

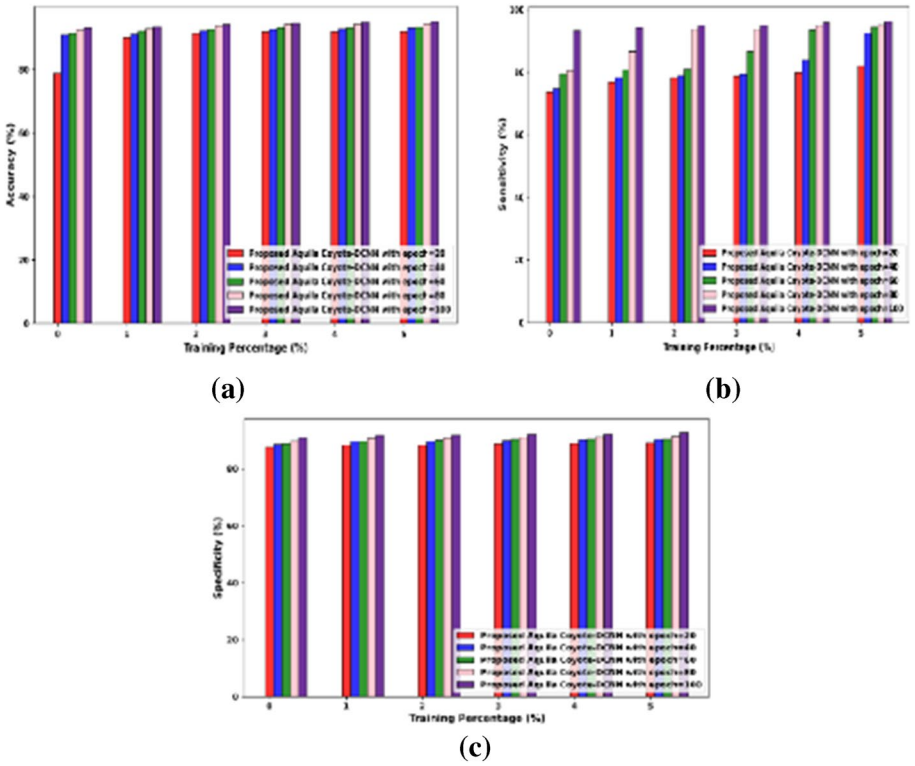

Fig. 5 Performance evaluation in terms of k-fold value (**a**) accuracy, (**b**) sensitivity, and (**c**) specificity

88.8487, 91.8732, and 91.8732, respectively, for a training percentage of 50%. The specificity of the Aquila coyote-DCNN approach for the epoch of 20, 40, 60, 80, and 100 is 83.4651, 84.1618, 90.0275, 91.2579, and 93.3087, respectively, for a training percentage of 60%. The specificity of the Aquila coyote-DCNN method for the epoch of 20, 40, 60, 80, and 100 is 82.6448, 83.6702, 87.1564, 90.0714, and 93.5137, respectively, for a training percentage of 70%. The specificity of the Aquila coyote-DCNN method for the epoch of 20, 40, 60, 80, and 100 is 83.6702, 9.2072, 89.6173, 91.0529, and 92.9244, respectively, for a training percentage of 80%. Moreover, Performance analysis in terms accuracy, sensitivity, and specificity measures ate enclosed in Table 3.

4.5.2 Performance analysis in terms of k-fold value

Figure 5 shows how the Aquila coyote-DCNN approach performs in terms of the k-fold value. The precision of the performance is displayed in Fig. 5(a). The accuracy of the Aquila coyote-DCNN technique for the k-fold value of 2 is 71.1990, 91.7092, 92.0918, 92.9847, and 93.3673, respectively, for the epoch of 20, 40, 60, 80, and 100. The accuracy of the Aquila coyote-DCNN approach for the k-fold value of 4 is 76.1582, 89.7959, 91.4541, 91.7092, and 94.1327, respectively, for the epoch of 20, 40, 60, 80, and 100. The accuracy of the Aquila coyote-DCNN method for the k-fold value of 8 is 90.3061, 91.3265, 92.2194,

93.1122, and 94.3878, respectively, for the epoch of 20, 40, 60, 80, and 100. The accuracy of the Aquila coyote-DCNN method for the k-fold value of 10 is 91.9643, 90.9439, 89.0306, 91.1990, and 93.75, respectively, for the epochs of 20, 40, 60, 80, and 100.

The performance based on sensitivity is shown in Fig. 5(b). The accuracy of the Aquila coyote-DCNN approach for the k-fold value of 2 is 85.0341, 85.2887, 85.7979, 85.9213, and 87.3255, respectively, for the epoch of 20, 40, 60, 80, and 100. The Aquila coyote-DCNN method's sensitivity for the k-fold value of 4 is 80.4514, 84.7795, 85.1827, 86.5617, and 89.6168, respectively, for the epochs of 20, 40, 60, 80, and 100. The Aquila coyote-DCNN method's sensitivity for the k-fold value of 8 is 85.2887, 85.7979, 87.3255, 84.2703, and 87.3985, respectively, for the epochs of 20, 40, 60, 80, and 100. The Aquila coyote-DCNN method's sensitivity for the k-fold value of 10 is 82.4882, 84.0157, 79.1785, 83.9518, and 87.8909, respectively, for the epochs of 20, 40, 60, 80, and 100.

The accomplish competitive on specificity is shown in Fig. 5(c). The accuracy of the Aquila coyote-DCNN approach for the k-fold value of 2 is 8591.0308, 92.6675, 92.6675, 92.9082, and 95.5558, respectively, for the epoch of 20, 40, 60, 80, and 100. The specificity of the Aquila coyote-DCNN method for the k-fold value of 4 is 84.7246, 92.1861, 92.2744, 92.9082, and 95.7965, respectively, for the epoch of 20, 40, 60, 80, and 100. The specificity of the Aquila coyote-DCNN method for the k-fold value of 8 is 89.7792, 2.6675, 92.9082, 93.2692, and 95.5558, respectively, for the epoch of 20, 40, 60, 80, and 100. The specificity of the Aquila coyote-DCNN method for the k-fold value of 10 is 92.1861, 93.0205, 93.1489, 94.0154, and 95.5558, respectively, for the epoch of 20, 40, 60, 80, and 100.

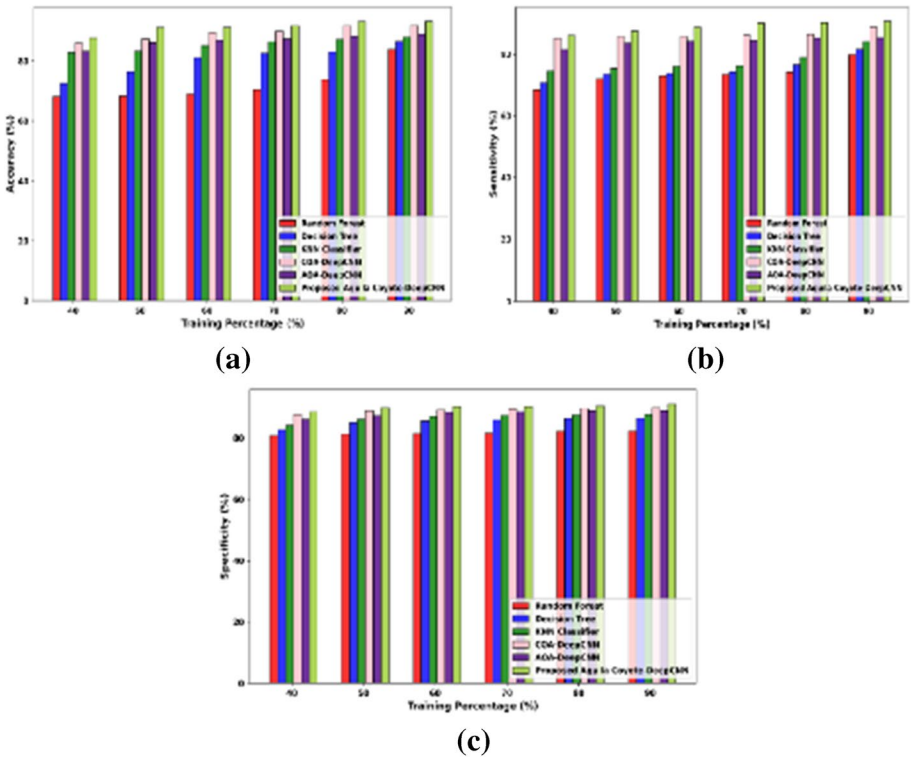

Fig. 6 Performance evaluation in terms of training percentage (**a**) accuracy, (**b**) sensitivity, and (**c**) specificity

4.6 Comparative evaluation

The comparative evaluation of approaches in terms of k-fold and training % is discussed in this section. The strategies considered for comparison with the proposed Aquila coyote-DCNN are the Random forest classifier [22], Decision tree classifier [23], K-nearest neighbor classifier [24], AOA-DCNN [8, 15], and the COA-DCNN [8, 16].

4.7 Comparative analysis in terms of training percentage

Figure 6 illustrates the comparison of approaches based on training %. Figure 6 shows the comparative analysis based on accuracy (a). For a training percentage of 50%, the accuracy of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is 68.3787, 76.4293, 83.3573, 87.3349, 86.1171, and 91.2007, respectively. For a training percentage of 60%, the accuracy of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is 68.9256, 81.1149, 85.1253, 89.5199, 86.9367, and 91.2386, respectively. For a training percentage of 70%, the accuracy of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is 70.4974, 82.6098, 86.4165, 89.9221, 87.4083, and 91.8602, respectively. For a training percentage of 80%, the accuracy of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is 73.8361, 82.9969, 87.2366, 91.8895, 88.3572, and 93.2449, respectively.

Figure 6 displays the comparative assessment based on sensitivity (b). For a training proportion of 50%, the sensitivity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 71.9607, 73.439, 75.3039, 85.6428, 83.612, and 87.4083. For a training proportion of 60%, the sensitivity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 72.9002, 73.7337, 76.0505, 85.6719, 84.1976, and 88.6995. For a training percentage of 70%, the sensitivity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 73.2913, 74.2321, 76.2303, 86.2085, 84.4342, and 89.9907. For an 80% training rate, the sensitivity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the proposed Aquila coyote-DCNN, is calculated as follows: 74.1279, 76.5735, 78.9036, 86.4165, 85.1527, and 90.1633.

Figure 6 displays the comparative assessment based on specificity (c). For a training percentage of 50%, the specificity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN is 81.2576, 85.1253, 86.1171, 88.9989, 87.372, and 89.9104, respectively. For a training percentage of 60%, the specificity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is, respectively, 81.5179, 85.5442, 86.938, 89.1734, 88.3452, and 90.2491. For a training percentage of 70%, the specificity of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 81.6444, 85.89, 87.4083, 89.4763, 88.6415, and 90.2510. The specificity of various methods, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the proposed Aquila coyote-DCNN, is 82.217, 86.4165, 87.5654, 89.6388, 88.9318, and 90.4244, respectively, for a training percentage of 80%.

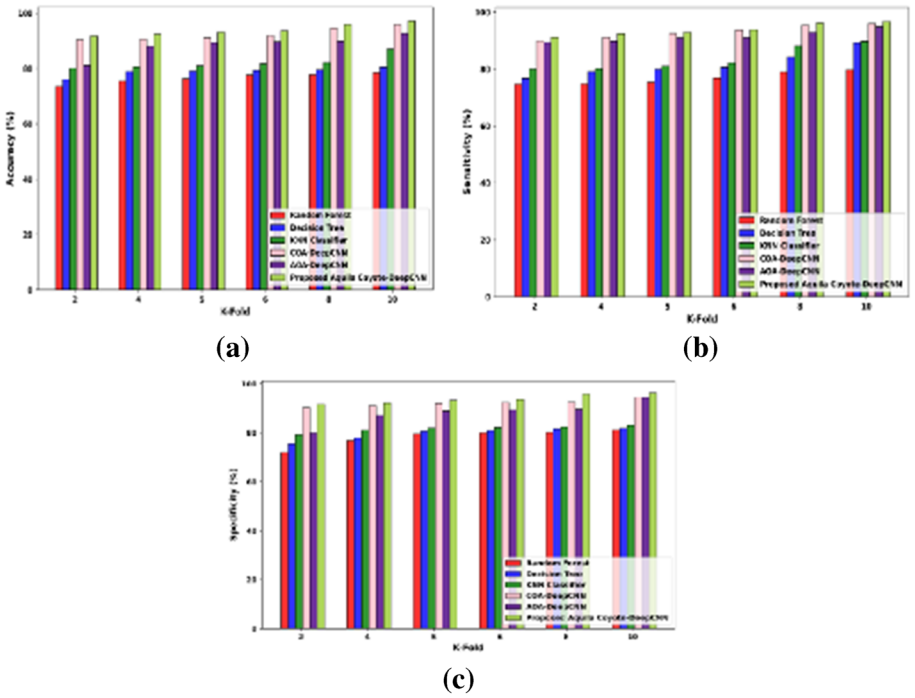

Fig. 7 Performance analysis in terms of k-fold value (**a**) accuracy, (**b**) sensitivity, and (**c**) specificity

4.8 Comparative analysis in terms of k-fold value

Figure 7 presents the comparison evaluation of the techniques in terms of k-fold value. Figure 7 shows the comparative analysis based on accuracy (a). For the k-fold value of 2, the accuracies of the various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, are, in order, 75.5252, 79.0112, 80.657, 90.6617, 88.1077, and 92.4624. For the k-fold number of 4, the accuracy of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 76.4421, 79.1742, 80.9753, 91.1669, 89.4314, and 93.1536. For the k-fold value of 6, the accuracy of several approaches including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN is, respectively, 77.7786, 79.4248, 81.8386, 91.8211, 89.7346, and 93.6999. For the k-fold number of 8, the accuracy of various approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 77.875, 79.5841, 82.1397, 94.6434, 89.9294, and 95.9758.

Figure 7 displays the comparative assessment based on sensitivity (b). For the k-fold value of 2, the sensitivity of several approaches, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN, is calculated as follows: 74.7711, 79.1742, 80.1896, 91.1477, 89.8091, and 92.4034. For the k-fold value of 4, the sensitivity of several techniques including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN is 75.4177, 79.9647, 81.0903, 92.513, 91.0142, and 92.8561, respectively. For the k-fold value of 6, the sensitivity of several techniques including RF,

Table 4 Comparative discussion

S.No	Evaluation means	Metrics	Methods					
			RF	DT	K-NN	COA-DCNN	AOA-DCNN	Proposed Aquila coyote-DCNN
1	Training percentage	Accuracy (%)	83.8393	86.5975	88.0304	91.9395	88.9453	93.3098
		Sensitivity (%)	79.8544	81.5170	84.0683	88.7744	85.2232	90.6915
		Specificity (%)	82.2262	86.5174	87.7077	89.9000	88.9597	91.1678
2	K-fold value	Accuracy (%)	78.4583	80.6062	87.1183	95.9419	92.6790	97.3017
		Sensitivity (%)	79.7418	89.2863	89.7412	96.0144	94.8917	96.8194
		Specificity (%)	81.2730	81.9004	83.1890	94.4364	94.3550	96.4079

DT, K-NN, COA-DCNN, AOA-DCNN, and the suggested Aquila coyote-DCNN is 76.8652, 80.657, 82.1397, 93.7612, 91.0576, and 93.8587, respectively. The methods' respective sensitivities are 79.0311, 84.3043, 88.1917, 95.4515, 92.8692, and 96.2199 for RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the proposed Aquila coyote-DCNN with a k-fold of 8.

Figure 7 illustrates the comparative evaluation based on specificity (c). The specificity of various algorithms, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the proposed Aquila coyote-DCNN, is calculated as follows: 76.961, 77.8768, 81.0903, 91.0118, 87.2237, and 92.2614 for the k-fold value of 2. For the k-fold value of 4, the specificities of various methods, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the proposed Aquila coyote-DCNN, are, in order, 7079.5821, 80.657, 81.982, 92.1453, 88.9994, and 93.4233. For the k-fold value of 6, the specificities of various methods, including RF, DT, K-NN, COA-DCNN, AOA-DCNN, and the proposed Aquila coyote-DCNN, are, in order, 80.0228, 80.9064, 82.1397, 92.404, 89.3432, and 93.7262. The methods' respective degrees of specificity are 80.2236, 81.7063, 82.3239, 92.4984, 89.5996, and 95.7724 for the proposed Aquila coyote-DCNN with k-fold measure of 8.

In Table 4, a comparison of the techniques used to segment and categorise brain tumours is presented. The analysis shows that, when compared to other techniques, the Aquila coyote-DCNN method achieves greater accuracy and efficiency in the segmentation and classification of brain tumours.

5 Conclusion

In this study, a hybrid optimization approach for brain tumour segmentation from MRI data is proposed. For the classification of brain tumours, an AI-based classification technique is also introduced. The Aquila Coyote Optimization Algorithm specifically involved in the segmentation of brain tumour, and then the DCNN classifier does the classification procedure using the features collected from the segmented picture. By varying the weights of the DCNN classifier, the proposed Aquila coyote method plays a crucial role in improving the effectiveness of the proposed classification model. The evaluation indices accuracy, sensitivity, and specificity, which are obtained to be 97.3017%, 96.8194%, and 96.4079%, respectively, are used to examine the effectiveness of the Aquila coyote-DCNN technique. In the future, ensemble model will be used to increase the precision of the suggested model for classifying and segmenting brain tumours.

Funding Not applicable.

Data availability Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Code availability Not applicable.

Declarations

Informed consent Informed consent does not apply as this was a retrospective review with no identifying patient information.

Consent to participate Not applicable.

Consent for publication Not applicable.

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human or animal subjects performed by any of the authors.

References

1. George EB, Rosline GJ, Rajesh DG (2015) Brain tumor segmentation using Cuckoo search optimization for magnetic resonance images. In 2015 IEEE 8th GCC Conference & Exhibition (pp 1–6). IEEE
2. Singh A, Sharma DK (2020) Image collection summarization: Past, present and future. Spotting Data Points with Artificial Intelligence, Data Visualization and Knowledge Engineering, pp 49–78
3. Soleimani V, Vincheh FH (2013) Improving ant colony optimization for brain MRI image segmentation and brain tumor diagnosis. In 2013 first Iranian conference on pattern recognition and image analysis (PRIA) (pp 1–6). IEEE
4. Nema S, Dudhane A, Murala S, Naidu S (2020) RescueNet: An unpaired GAN for brain tumor segmentation. Biomed Signal Process Control 55:101641
5. Natarajan A, Kumarasamy S (2019) Efficient segmentation of brain tumor using FL-SNM with a metaheuristic approach to optimization. J Med Syst 43(2):25
6. Shivhare SN, Kumar N (2021) Tumor bagging: a novel framework for brain tumor segmentation using metaheuristic optimization algorithms. Multimed Tools Appl 80(17):26969–26995
7. Chen S, Ding C, Liu M (2019) Dual-force convolutional neural networks for accurate brain tumor segmentation. Pattern Recogn 88:90–100
8. Sajjad M, Khan S, Muhammad K, Wu W, Ullah A, Baik SW (2019) Multi-grade brain tumor classification using deep CNN with extensive data augmentation. J Comput Sci 30:174–182
9. Deepak S, Ameer PM (2019) Brain tumor classification using deep CNN features via transfer learning. Comput Biol Med 111:103345
10. Tandel GS, Balestrieri A, Jujaray T, Khanna NN, Saba L, Suri JS (2020) Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm. Comput Biol Med 122:103804
11. Swati ZNK, Zhao Q, Kabir M, Ali F, Ali Z, Ahmed S, Lu J (2019) Brain tumor classification for MR images using transfer learning and fine-tuning. Comput Med Imaging Graph 75:34–46
12. Yang XS, Deb S (2014) Cuckoo search: recent advances and applications. Neural Comput Appl 24:169–174
13. Layeb A (2011) A novel quantum inspired cuckoo search for knapsack problems. Int J Bio-Inspired Comput 3(5):297–305
14. Zikic D, Ioannou Y, Brown M, Criminisi A (2014) Segmentation of brain tumor tissues with convolutional neural networks. Proc MICCAI-BRATS 36(2014):36–39
15. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng 157:107250
16. Pierezan J, Coelho LDS (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. In 2018 IEEE congress on evolutionary computation (CEC) (pp 1–8). IEEE
17. Kumari N, Saxena S (2018) Review of brain tumor segmentation and classification. In 2018 International conference on current trends towards converging technologies (ICCTCT) (pp 1–6). IEEE
18. Deng W, Shi Q, Luo K, Yang Y, Ning N (2019) Brain tumor segmentation based on improved convolutional neural network in combination with non-quantifiable local texture feature. J Med Syst 43:1–9
19. Mathew AR, Anto PB, Thara NK (2017) Brain tumor segmentation and classification using DWT, Gabour wavelet and GLCM. In 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (pp 1744–1750). IEEE
20. Sawakare S, Chaudhari D (2014) Classification of brain tumor using discrete wavelet transform, principal component analysis and probabilistic neural network. Int J Res Emerg Sci Technol 1(6):2349–2761
21. Brain Tumor Segmentation dataset. https://www.kaggle.com/andrewmvd/brain-tumor-segmentation-in-mri-brats-2015. Accessed on Nov 2021
22. Lefkovits L, Lefkovits S, Szilágyi L (2016) Brain tumor segmentation with optimized random forest. In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Second International Workshop, BrainLes 2016, with the Challenges on BRATS, ISLES and mTOP 2016, Held in

Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Revised Selected Papers 2 (pp 88–99). Springer International Publishing
23. Naik J, Patel S (2014) Tumor detection and classification using decision tree in brain MRI. Int J Comput Sci Netw Secur (IJCSNS) 14(6):87
24. Sundararaj GK, Balamurugan V (2014) Robust classification of primary brain tumor in Computer Tomography images using K-NN and linear SVM. In 2014 International Conference on Contemporary Computing and Informatics (IC3I) (pp 1315–1319). IEEE