Journal of Theoretical and Applied Information Technology
15th May 2025. Vol.103. No.9
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN IMPROVED XCEPTION BASED FEATURE LEARNING AND HONEYBADGER OPTIMIZED LONG SHORT-TERM MEMORY FOR ACCURATE BRAIN TUMOR CLASSIFICATION WITH DATASET BALANCING APPROACH

PARVATHY JYOTHI [1], DR.S.DHANASEKARAN[2]

Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education
Krishnankoil , Tamilnadu,India

Associate professor, Department of Information Technology, Kalasalingam Academy of Research and
Education Krishnankoil , Tamilnadu,India

parvathyjyothi@gmail.com, srividhans@gmail.com

ABSTRACT

A brain tumour (BT) is a severe and lethal disease that significantly reduces human lifespan. Magnetic resonance imaging (MRI) is a commonly employed imaging technique for the early detection of tumours. The segmentation and classification of brain tumours (BTs) through manual and traditional methods is a labour-intensive and subjective process, necessitating the involvement of expert radiologists for evaluation, which may result in prediction inaccuracies. Class imbalance presents a notable challenge in MRI datasets, impacting the efficacy of the classification system. This paper presents an optimised deep learning approach, referred to as Honey Badger optimised Long Short-Term Memory (HBLSTM), aimed at the detection of BT through effective segmentation and feature extraction methodologies. The preprocessing steps are conducted on the collected CE-MRI dataset to reduce noise through Gaussian filtering. The class imbalance issue is addressed through the application of the Adaptive Synthetic Sampling (ADASYN) model. The system employs the Spatial and Channel attention-based Three-Dimensional U-shaped Network (SC3DUNet) for tumour segmentation. The segmentation images utilise the most discriminative features through the Spatial Pyramid Pooling centred Xception Network (SPPXNet). The essential features are subsequently selected utilising the Diagonal Linear Uniform and Tangent Flight-based Butterfly Optimisation Algorithm (DTBOA). The tumour classes are classified utilising the HBLSTM algorithm. The experimental results demonstrate the efficacy of our hybrid deep learning models, achieving an average accuracy of 99.81% in tumour detection, surpassing current state-of-the-art models.

Keywords: *Brain Tumor, Segmentation, Magnetic Resonance Imaging, Figshare Dataset, Dataset Balancing, Pre-trained CNN, Deep Learning*

1. INTRODUCTION

For human beings, a BT is one of the most threatening malignancies caused by uncontrolled and abnormal cell divisions, mainly in the brain or in the nerves of the skull [1]. Almost 120 categories of BTs are affecting people, which the National BT Society reported. The primary classes are Glioma, Meningioma, and Pituitary tumors. Almost 75% of BTs were covered by these three types, which include 45% for Glioma, 15% for Meningioma, and 15% for Pituitary [2]. Nowadays, Magneto Encephalo Graphy (MEG), X-ray, Ultrasonography, Computed Tomography (CT), MRI, Electro Encephalo Graphy (EEG), and Positron Emission Tomography (PET) have not only emerged to demonstrate the complete and detailed facets of BTs but also to analyze the tumor precisely and provide exact treatment mechanisms by the doctors [3]. MRI is the most familiar imaging approach for determining BTs [4]. The initial step is considered tumor segmentation in the analysis of MRIs of infected people. A lot of time and effort was spent during the manual segmentation of tumor areas because it included many tissue regions and varying degrees of degradation. Fuzzy clustering, region-based segmentation, thresholding, watershed, k-means clustering, and Otsu's method are widely used approaches to perform tumor segmentation, which avoids manual segmentation [5]. In addition to these models, CNN and its variants are also used in image segmentation processes by learning the hierarchical features of MRI and capturing the essential features like shape, texture, and intensity variations related to the tumors.

Recently, the 3D-UNet has gained significant traction for image segmentation in the biomedical domain. It yields favourable outcomes with numerous examples. Extracting deeply hidden features for tumour segmentation remains challenging, despite the superior performance of the 3D-UNet model. The system employed a modified 3D-UNet model to extract local and global features for improved tumour segmentation accuracy. The fundamental function of medical image analysis is image classification [6, 7]. It is essential to identify the tumour type present in the patient, such as glioma, pituitary tumour, or meningioma [8]. Recent advancements in machine learning, particularly in deep learning, have facilitated the recognition and classification of brain tumours from MRI patterns. It serves effectively as an alternative to acquiring knowledge from experts or technical literature, particularly in retrieving and extracting information from data. The researchers in deep learning utilised convolutional neural networks to achieve effective classification performance and feature extraction [10]. Employing convolutional neural networks (CNNs) in medical imaging remains a challenge. First, an increase in the volume of data within medical image datasets is necessary, as the process of saving images to the database is challenging and time-intensive for radiologists. Secondly, training convolutional neural networks on small datasets presents challenges related to overfitting.

Consequently, for visible solutions to address those challenges, they used pre-trained models for transfer learning (TL) and fine-tuning [11]. The process involves using a network and training it with a wide-ranging database; this allows the filter to use the weights to create multipart activation maps associated with the dataset. The network learns the assignment with a high degree of overview only when the database is large enough [12]. The researchers trained several pre-trained models, such as residual network 50 (ResNet50), visual geometry group (VGG), GoogleNet, MobileNet v2, SqueezeNet, Inception V3, Xception, etc., to perform a specific task via transfer learning [13, 14]. The Xception network is a reliable and effective technique for best feature representation with the help of depth-wise separable convolutions that maintain higher accuracy without maximizing computational complexity [15]. However, it extracting the deep spatial features from the tumor images is difficult, so they still need improvement to categorize the tumor classes with less time. Thus, for BT classification from the MRI, the system proposes a SPPXNet-based pre-trained model along with HBLTSM to perform rich feature learning and tumor classification. The major objectives of the paper is explained as follows:

- The system employs the ADASYN method to solve the dataset's class imbalance problem by preventing the results from being biased toward a single class, i.e., the majority class.
- The system segments the tumor lesions using the SC3DUNet model, incorporating a spatial and channel attention mechanism to enhance the quality of segmentation throughout its feature hierarchy.
- The system introduces the SPPXNet model to extract features from the segmented images, incorporating SPP to learn features at a deeper level.
- The system employs the DTBOA algorithm to optimally find the best features that reduce overfitting and improve model performance.
- The system develops an HBLSTM model to classify the tumor classes; the hyperparameter is optimally selected via the GAHBO algorithm to improve the generalization and robustness of the classifier.

The paper's remaining portions are sectioned as follows: Section 2 presents the discussion of the most related works. The proposed mechanism includes all its phases in depth, as discussed in section 3. The outcomes of the proposed and existing methods regarding some performance indicators are discussed in section 4. Finally, section 5 describes the conclusion and future scope of the work.

2. LITERATURE REVIEW

This section presents the significant earlier works in BTs segmentation and classification from MRI, providing a comprehensive understanding of the field's progress.

A BT segmentation and classification system using an improved unsupervised clustering technique and an Extreme Learning Machine (ELM) was suggested by **Usharani Bhimavarapu *et al.* [16].** The system used complete mean deviation as well as the kurtosis function-based improved hybrid contrast enhancement model for enhancing images' contrast. Then, the system used the improved fuzzy model for image segmentation, and then the shapes, textures, and color features were extracted. Finally, the ELM was used to identify tumor classes. On the Fig share dataset, the method attained accuracy, precision, and recall rates of 98.47%, 98.59%, and 98.74%. **Sarmad Maqsood *et al.* [17]** presented deep neural networks and multiclass SVM (M-SVM) for multi-modal BT detection**.** At first, the edges from the source image were determined using a linear contrast stretching algorithm, and then a custom 17-layered DNN was used to segment the BTs. Thirdly, the feature extraction and selection processes were carried out using a modified MobileNetV2 architecture and entropy-based controlled method. Finally, M-SVM was utilized to classify BTs. The system was tested on BraTS 2018 and Figshare datasets, achieving 97.47% and 98.92% accuracy.

An MRI-based BT categorization using CNN and Bayesian optimization was presented by **Mohamed Ait Amou *et al.* [18].** The system used CNN on the collected images for tumor classification, in which hyperparameters were optimally selected using a Bayesian optimization algorithm. The system achieved 98.70% accuracy with the experimentation done on the T1W-CE MRI dataset. An MRI-based DL approach was presented by **Ejaz Ul Haq *et al.* [19]** for BT detection. Preprocessing was performed at the beginning to remove the noise from the MRI. The system then used the DCNN model to classify the tumor; here, the activation function and gradient are maintained based on Xavier initialization. The system used Figshare and BraTS 2018 datasets for experimentation, achieving 97.3% and 96.5% accuracy for detection. A hybrid DL approach was suggested by **Shaimaa E. Nassar *et al.* [20]** for robust BT classification. First, data augmentation was done to improve the dataset's quality and to classify the tumors; the AlexNet, ShufeNet, SqueezeNet, GoogleNet, and NASNet-Mobile methods were used. It achieved 99% accuracy with the experimentation on 3064 T1-weighted contrast-enhanced brain MR images (T1W-CE MRI).

The categorization of tumors in MRI based on DL was proffered by **Prince Priya Malla *et al.* [21].** The system used VGGNet to extract more discriminative features, and also overfitting and gradient vanishing issues were avoided using a global average pooling layer at the output. Lastly, the distinct features were classified based on a log-SoftMax layer, and the system reached 98.93% accuracy for the Figshare dataset. A novel DL-based BT classification model was proffered by **Mohd Anul Haq *et al.* [22].** To initiate, the system collected MRI from the MICCAI–RSNA and CE-MRI datasets and the system used six well-known DL algorithms such as SE-ResNet-101, EfficientNetB0, SENet154, SE-ResNet-152, EfficientNetB5, and ResNet152V2 to categorize the BTs. The method reached a maximum accuracy of 99% in the performance evaluation. **Gopal S. Tandel *et al.* [23]** suggested an ensemble deep-learning model for detecting BTs from multi-MRI sequence data. Initially, the augmentation was performed to avoid overfitting. Then, the system adopted five CNN models, say ResNet18, VGG16, AlexNet, GoogleNet, and ResNet50, for classifying the tumors. The experimentation was carried out on FLAIR, T2W, and T1W-MRI datasets, and the maximum test accuracy was 98.88%, 97.98%, and 94.75% for the respective datasets.

The MRI classification of BTs based on vision transformers was presented by **Sudhakar Tummala *et al.* [24].** The system's overall accuracy was 98.7% for the experimentation done on the T1W CE MRI database. **Abdullah A. Asiri *et al.* [25]** introduced a block-wise neural network for BT detection. To improve the quality of the dataset, the system first performed preprocessing. To classify the tumor classes, the system used a pre-trained VGG19 that was fine-tuned with CNN structural design in the block-wise mechanism, and it achieved 98% accuracy when the system used the CE-MRI dataset for evaluation. For classifying the BT, **Syed Muhammad Ahmed Hassan Shah *et al.* [26]** recommended a model, namely, the Voting Semi-Supervised Bayesian Ensemble Attention Mechanism (VS-BEAM). The VS-BEAM system integrated squeeze and excitation attention blocks to extract features, and a convolution autoencoder was used to segment the tumor. The system attained 98.91% accuracy while testing its efficiency by utilizing the T1W-CE MRI dataset. Based on hybrid DL models, the detection and categorization of BTs were proffered by **Baiju Babu Vimala *et al.* [27].** Preprocessing was performed first to prepare the samples. The system used pre-attained EfficientNet models, namely EfcientNetB0 and EfcientNetB4, for future extraction and classification. Then, the system utilized the CE-MRI Fig share dataset to fine-tune the model, and it offered overall accuracy and an F1-score of 99.1% and 98.7%.

Zheng Liyuan [28] presented a BT classification using the attention-guided deep-learning technique**.** A dual attention mechanism was used to choose the critical information affiliated with the objective area while avoiding irrelevant details. The experimental evaluation was made on the T1W CE-MRI database, and the approach achieved an overall accuracy of 98.61%. **K. Kavin Kumar *et al.*** [29] suggested an ensemble of transfer learning models, say ResNet 50, Inception V3 and AlexNet, with a data augmentation mechanism for BT detection. The experiment was performed on the T1W CE-MRI dataset, and the AlexNet model achieved 96.2% of higher accuracy than the other approaches. **Jiang Linqi *et al.*** [30] recommended an optimal SE-ResNeXt network for BT classification. Initially, the system dynamically adjusted the learning rate using MultiStepLR, which enhances the network's learning capability. Then, a label smoothing strategy was utilized to optimize the one-hot label to reduce the dependence of the network on the probability distribution of actual labels. Finally, the tumor classification was done using the SE-ResNeXt model, and the system reached an accuracy of 97.45% and 98.99% for the BraTS2017, and BraTS2019 datasets.

2.1 PROBLEM STATEMENT

The works above show better prediction results for BT prediction; however, the following are their significant limitations to address and improve its performance further for higher classification outcomes for BT detection.

- Noise removal is an important preprocessing step in any image processing work because the noise in the collected MRI image can degrade the quality of the image and affect its usefulness for analysis or display. However, none of the works focused on noise suppression.
- It cannot be overstated how crucial it is to balance the collected dataset to avoid biased outcomes. This imbalance occurs when one class has a significantly higher number of data samples than the others. If the dataset is not balanced, the classifier's outcomes for the minority class will be biased. This can lead to suboptimal treatment planning, such as ineffective radiation therapy.
- Few works focused on segmentation [16, 17], which is crucial and essential in the medical field and can help diagnose and predict overall growth. These traditional models of tumor segmentation require manual supervision for labelling the medical images, which is time-consuming and prone to human error. Also, the models do not scale well on higher-resolution images, and large datasets lead to computational complexity and time for processing the images effectively.
- Some works use these, such as DNN and CNN models, for BT classification [17–19]. They produce satisfactory performance; however, the random choice of hyperparameters in the network affects the model's suboptimal performance, lower accuracy, and sensitivity. It leads to the classifier being overfitted or underfitted in the training data. In addition, the poor choice of random parameters leads to higher training times. The model [18] uses Bayesian optimization for hyperparameter tuning, but the model suffers from local optimization issues and decreases the global search ability.
- While many studies utilize pre-trained CNN models for BT classification, it's important to note their limitations. These models do offer better feature learning ability than standard CNNs, requiring only a smaller amount of data for training. However, they suffer from spatial information loss due to downsampling features in deep pooling layers. This loss of spatial information can significantly impact the accuracy and effectiveness of the classification results.

Considering the limitations above, this paper offers solutions for effective BT detection using the following steps: Initially, Gaussian filtering is used to suppress the noise in the collected MRI, improving the image's quality for further analysis. After that, the ADASYN algorithm balances the dataset and prevents the classifier from having biased outcomes. Next, the system uses the SC3DUNet algorithm for segmentation that focuses on relevant parts or features of the input data. This helps attain finer details and the difference between tumor and healthy tissue more effectively than traditional segmentation approaches. Then, the system used the SPPXNet algorithm to extract the most discriminant features from the segmented tumor regions that offer improved performance over traditional pre-trained CNN models by pooling features at multiple scales and resolutions, thus reducing the loss of spatial information. After that, it uses the DTBOA algorithm to optimally select the features, leading to lower computational complexity and higher detection accuracy. Finally, the system uses the HBLSTM algorithm to classify the tumor classes based on the optimally selected features, in which the hyperparameters are optimally chosen using the GAHBO algorithm, leading to the optimal classification performance of the classifier.

3. PROPOSED METHODOLOGY

Figure 1 demonstrates the overall workflow of the proposed BT segmentation and classification method. The input images are collected from the publicly available CE-MRI dataset. Then, the system uses the following steps: preprocessing, dataset balancing, segmentation, feature extraction, feature selection, and classification to perform the detection task more accurately.

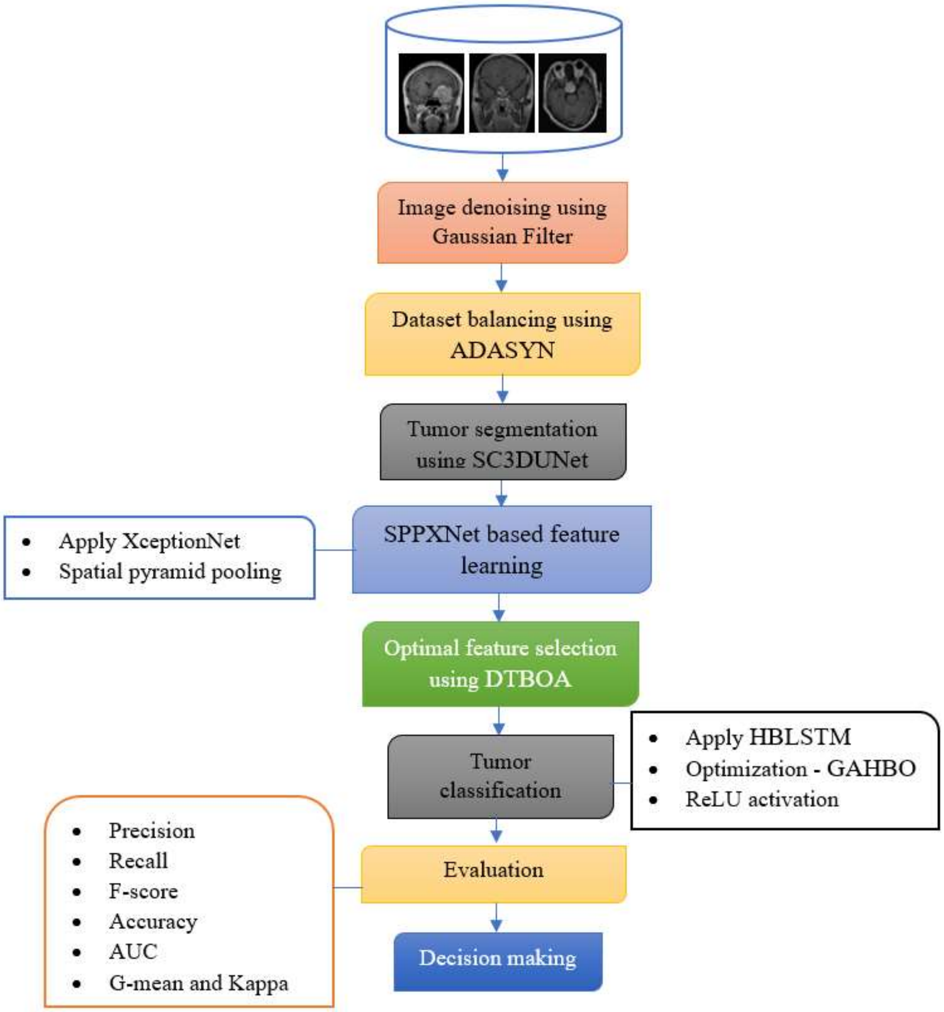


Figure 1: workflow of the proposed methodology

3.1 Preprocessing

MRI images suffer from different noise sources, including thermal noise, motion artifacts, and radiofrequency interference. These noises impose challenges in DL-based applications. Filtering and motion correction techniques are used for pre-processing to reduce noise and increase image quality. Gaussian filter is used in our study for the noise filtering process that reduces undesired signals or artifacts from MRI images and improves the image quality for subsequent analyses. Based on the Gaussian distribution, the weighted average of the surrounding pixel is computed by a non-uniform low-pass filter called the Gaussian filter. It is also used for noise reduction and is an unfocused lens practical model. This provides gentler smoothing and preserved edges, which is better than the existing filter algorithms. It is mathematically shown as follows:

$$\underline{\underline{\ddot{G}}}(m,n)=\frac{1}{2\pi\sigma^2}\exp\left\{-\frac{m^2+n^2}{2\sigma^2}\right\} \tag{1}$$

Here, $m\ and\ n$ refers to the horizontal and vertical axis' distance from the origin, and σ represents the Gaussian distribution's standard deviation.

3.2 Dataset Balancing

The data was assumed to be imbalanced when classes were not proportionally distributed. Many real-time datasets suffer from imbalanced datasets, where standard samples have more occurrences than abnormal samples. Overfitting and underfitting are the major issues faced by the classifiers running on these imbalanced datasets. Therefore, addressing the class imbalance is essential to ensure fair and accurate learning, eliminating bias towards the majority classes, and encouraging the model's ability to recognize all classes successfully, particularly in critical areas like medical diagnosis. Unfortunately, many oversampling techniques are not convenient for oversampling the multi-class imbalance problems. For example, the overlapping trouble occurs when the SMOTE is applied to the imbalanced multi-class datasets. Hence, the Adaptive Synthetic Sampling (ADASYN) method was used by the proposed system to balance the collected dataset. The weighted distribution is the main idea of ADASYN which is to use it for various minority class examples according to their learning difficulty with generating more synthetic samples for the minority classes. Therefore, the ADASYN technique used two ways to enhance the data distribution learning: firstly, class imbalance bias was reduced, and secondly, the decision boundary classification was adaptively sent towards demanding samples. The ADASYN steps are explained as follows:

Step 1: At first, degree of class imbalance is calculated $\left(\vec{\mathbb{D}}_I\right)$according to equation (2), where, Min_{CS} refers to minority and Maj_{CS} refers to majority class samples.

$$\vec{\mathbb{D}}_I=\frac{Min_{CS}}{Maj_{CS}} \tag{2}$$

In this, if $\left(\vec{\mathbb{D}}_I\right)$is lower than a particular threshold, initialize the algorithm.

Step 2: The number of synthetic data samples for the generation of minority class is computed as

$$\hat{\mathbb{S}}=\left(Maj_{CS}-Min_{CS}\right)\chi \tag{3}$$

Where, $\chi\in[0,1]$ refers to the parameter, that describes the desired balanced level after the generation of synthetic data samples and $\chi=1$ indicates the fully balanced data set after the ADASYN procedure.

Step 3: Each minority instance's k-nearest neighbors were obtained and the $\mathbb{R}_V$value is computed. The $\mathbb{R}_V$ value represents the domination of the majority class in every specific neighborhood. It is illustrated as follows:

$$\mathbb{R}_V=\frac{\#Majority}{K} \tag{4}$$

Step 4: Standardize the $\mathbb{R}_V$values so as to get the sum of all $\mathbb{R}_V$values equal to 1.

$$\hat{R}_V = \frac{R_V}{\sum R_V} \tag{5}$$

Where, $\hat{R}_V$ refers to the standardized R_V value.

Step 5: The number of synthetic examples for each minority sample was calculated according to equation (6).

$$\widehat{S}_V = \widehat{S}\,\hat{R}_V \tag{6}$$

Step 6: Generate $\widehat{S}_V$ quantity of data for every neighborhood, and then the new synthetic observation $\left(N_{OB}\right)$ was generated by equation (7).

$$N_{OB} = \breve{p}_V + \left(\breve{p}_{XV} - \breve{p}_V\right)\beta \tag{7}$$

Where, β refers to the random number between 0-1, $\breve{p}_V$ represents the minority samples of a neighborhood, and $\breve{p}_{XV}$ shows that the randomly chosen minority example from the same neighborhood.

3.3 Segmentation

Segmentation is a critical part of the image classification system where the brain image is segmented into several objects like white matter, grey matter, skull, cerebral spinal fluid, Tumors, etc., from the surroundings. Our paper uses our previously suggested model called spatial and channel attention-based three-dimensional u-shaped network (SC3DUNet) [31] for segmenting the tumor lesions. 3DUNet progresses a volumetric (3D) image that is MRI brain scans by keeping the U-shaped structure with skin connections operating in three dimensions. The basic U-net structures have two paths: contracting or encodes and decoder paths. The features at different levels of abstraction are captured using each layer of the encoder. Higher layers capture more abstract and semantic features and lower layers capture low-level features such as edges and textures. This enables the network to learn complex representations of the input data increasingly. The decoder uses up convolutions and concatenations with the first features of the path. However, only some of the generated features from the encoder are useful for segmentation, and the gaps between high- and low-level features are filled using the model's skip connections, resulting in detailed information's loss. So spatial and channel attention mechanisms are included in the conventional 3DUNet model to effectively leverage both spatial and channel-wise information present in the input data, leading to enhanced performance in tasks such as medical image segmentation. Also, cross-dimensional interactions are captured by the model to avoid dimensionality reduction and improve the prediction performance with less computational cost.

3.4 Feature Extraction

The feature extraction process includes identifying and extracting significant features from the segmented images. The Spatial Pyramid pooling centered Xception Network (SPPXNet) is used by the proposed system to extract features from the segmented image. The Xception Network (XNet) architecture, also called ''Extreme Inception,'' is one of the famous and strong pre-trained models compared to other networks that are advanced under various essential concepts, such as depthwise separable convolution layer, convolutional layer, residual connections, which reduce the computational complexity while maintaining high accuracy. It contains 36 convolutional layers and 71 layers. The fully used depthwise separable convolutions as an alternative to Inception modules make this model unique. For every channel, the 1x1 convolutions are used to extract the cross-channel correlations, and for each output, the 3x3 convolutions are used to extract the spatial correlations.

The depthwise convolution is the process of the original depth-wise separable convolution layer, followed by a pointwise convolution; however, the XNet architecture has modified this by maintaining the pointwise convolution, followed by depthwise convolution. It also used GAP at the network end, which takes the average of all feature maps across all proportions, leading to the network's information loss specifically for large-size input images. Also, a layer takes the input at a fixed size so that to process the image, other operations such as cropping or resizing are required. To overcome these difficulties, the Spatial Pyramid Pooling (SPP) strategy was used instead of GAP, which pools feature maps at different scales without information loss by protecting more spatial information. The model handles the various input sizes without cropping or resizing, which makes them flexible and adaptable for the network. SPP improves the model's accuracy by pooling feature maps at various scales compared to GAP. Thus, the SPP incorporation in the existing XNet to enhance its performance is known as SPPXNet. The proposed SPPXNet diagram is shown in Figure 2.

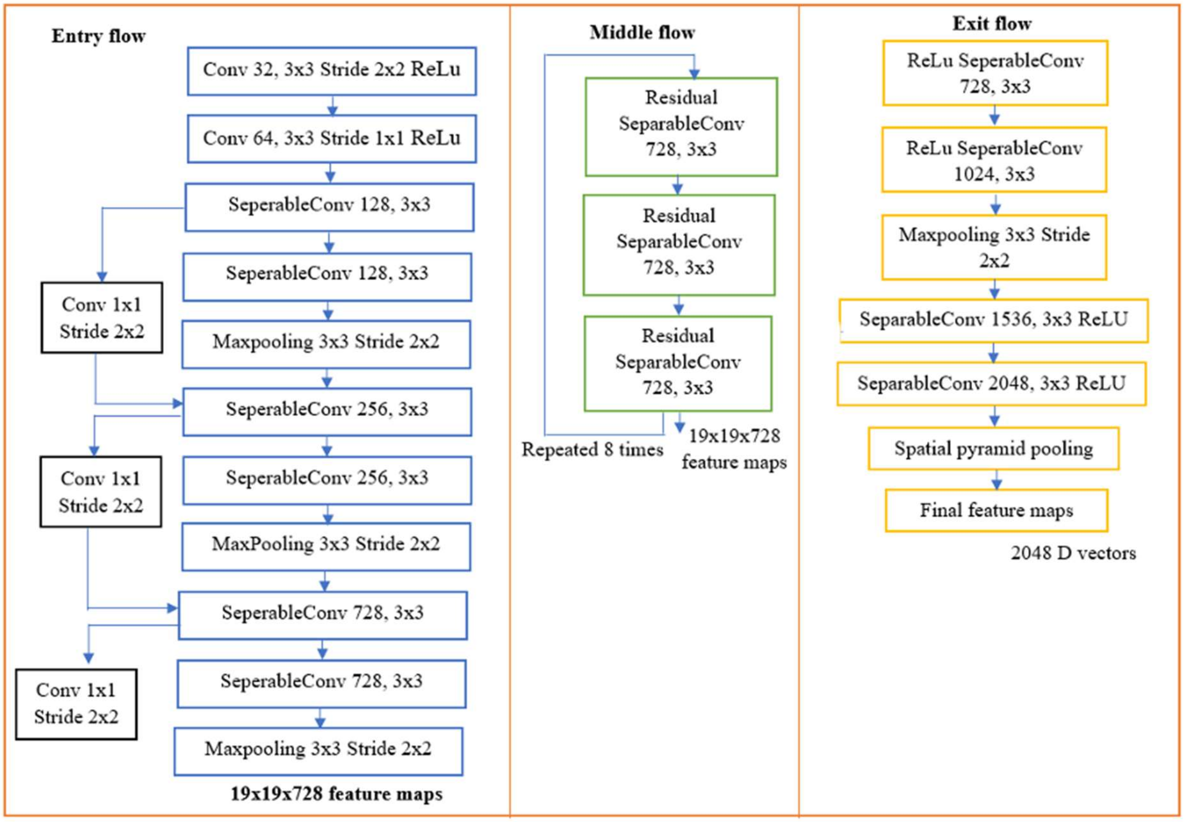


Figure 2: Architecture of the proposed Xception model

Figure 2 mainly consists of three flows. The initial flow is based on the XNet network, which is used for feature extraction. The second one is an intermediate flow that consists of separate convolution layers. The middle layer was repeated eight times. The final layer is the exit flow, which builds the dense layer. The brief explanations of these layers are as follows:

a) *Initial flow*

First, the input segmented images go through the entry flow by the size of 299×299×3. Then, it travels to the 1 x 1 convolutional layers with stride 2. The primary part of the XNet architecture is this convolution layer where feature extraction is performed. It mainly contains a mixture of linear and nonlinear processes, i.e., convolution and activation functions. A special form of a linear function called convolution is used for extracting the features, and a small array of numbers known as a kernel is functional to the input called a tensor. At each tensor's location, the system computes the element-wise product betwixt the input tensor and the kernel's element, which is then summed to get the output value in the output tensor's equivalent position. The convoluted features are passed to a separable convolution layer with a 3x3 kernel, and these layers consist of depth-wise convolutions, which is the Xception' s main part. So, the computation is decreased, and the model parameters are organized in the color channels as depth dimensions and spatial dimensions. The depth-wise convolution filters the input data channel and produces the feature map.

The depth-wise convolution of the input channel filter is attained using equation (8):

$$\breve{K}_{l,m,i} = \sum_{u,v,i} \underline{\underline{ED}}_{u,v,i} \times \breve{I}_{l+u-1,\, m+v-1,\, i} \tag{8}$$

Where, $\breve{K}$ shows the feature maps' alternative output created by $\breve{I}$ as the input feature map, $\underline{\underline{ED}}$ represents the depth wise convolution kernel, l and m indicates the filter size, and i indicates the channel, correspondingly.

Batch normalization is utilized after the depth-wise separable convolution layer. During the neural network training, all layers affect each layer's distributed input values that come before it. This variability decreases the speed of training (lower learning rate). The batch normalization is created to solve this inconsistency and speed up the learning. After that, to decrease the computational complexity, the features are passed to the pooling layer, which uses the 1 x 1 pooling with stride 2. The model's pooling layer decreases the input matrixes size by decreasing the number of parameters and the quantity of network computation, which prevents overfitting. Like, the convolutional layer, multiple input values were taken by the pooling layer inside a filter from the previous layer and until every part of the input matrix process, the filters transmit some pixels at a time. In this network, the dimension is reduced by the max-pooling layer by taking the minimum value within the window. The height and the weight of the incoming array are only decreased by the max-pooling layer, and it does not change the number of channels. It is mathematically shown as follows:

$$P''_{fm} = P''_{o}\left(C''_{FM}\right) \tag{9}$$

Where, P''_{fm} indicates the pooling map, P''_{o} indicates the pooling operation, and C''_{FM} represents the convoluted feature maps. Every convolution and pooling layer are followed by a ReLU activation function. The activation function decides the neuron activation by calculating the weighted sum and adding bias to it. It is used to initiate non-linearity in the neuron output. The entry flow produces the feature maps with a size of 19x19x728.

b) Intermediate flow

The original XNet's intermediate flow contains eight residual separable convolutional blocks within a single branch. By using different levels of convolution, the network can study various high-dimensional semantic features of the image. Three separable convolutional layers and ReLU activation functions are within the intermediate layer. The convolution kernel's size of each layer of the middle flow is 3 × 3, the number of the channels is 728, and in the intermediate stream, the data tensor is repeated eight times. By doing this, we can reduce the feature map dimension into 19×19×728 maps given to the exit flow.

c) Exit flow

In the exit flow, the network performed a computation sequence similar to the entry flow, followed by pooling. This flow contains four separable convolutional layers that flatten the features and are given to the SPP block as an alternative to the GAP layer to reduce the dimensions. The SPP block takes averages of full feature maps, and this layer pools the features and generates the output of fixed-length. Before the SPP, the extracted feature map was usually flattened and applied in a sliding window fashion, which gives an output of varied size. These SPPs maintain the local spatial bins' spatial information, in which the bins' quantities as well as size are permanent. The system pools each filter's spatial bin responses, and three levels of pooling functions are used in SPP blocks.

1. In the first pooling layer, the output contains a single bin and covers a complete feature. This is similar to the global pooling operation, and the output is 728.
2. In the second pooling layer, the feature map is pooled into 4 bins, resulting in an output of size 4*728.
3. In the third pooling, 16 bins of feature maps are gathered, resulting in an output size of 16*728.

Every pooling layer output was flattened and concatenated to provide a fixed dimension irrespective of input size. Finally, this vector was taken as an input by the fully connected network. A bias value is added to the connection weights, and the data from the preceding layer is multiplied with them to obtain the final feature vectors. The output produces a feature size of 19 × 19 × 728, and it carries 2048 features for a standard-size of input image. The layer information of the proposed SPPXNet is presented in Table 1.

Table 1: Layers information

Flows	Layers	Output size	Settings
Entry flow	Conv	299×299×3	Conv 1×1 and stride 2×2
	SeparableConv	256×256×3	Conv 3×3 and stride 2×2
	Max-pool	19×19×728	Conv 1×1 and stride 2×2
Middle flow	SeparableConv	19×19×728	Conv 3×3 and stride 2×2 (Repeated 8 times)
Exit flow	SeparableConv	19×19×728	Conv 3×3 and stride 2×2
	SPP	19×19×728	Conv 1×1 and stride 2×2
	Fully connected	2048-dimensional vector	1×1 and stride 2×2

3.5 Feature Selection

Feature selection is a process that isolates the most consistent, non-redundant, and relevant features from the extracted features for model construction. The proposed system uses the earlier suggested diagonal linear uniform with a tangent flight-based butterfly optimization algorithm (DTBOA) [31] for feature selection. The BOA indicates butterflies' food-seeking as well as mating behaviour. The algorithm is relatively simple to implement than some other optimization algorithms that can necessitate complex mathematical formulations. The model achieves robust performance even for noisy data or incomplete problem formulations. Even though BOA produces efficient outcomes, they can occasionally be prone to local optima and fail to explore the search areas globally, which leads to premature convergence. So, two mechanisms, diagonal linear uniform (DLU) and tangent flight (TAF), are included in BOA to overcome its drawbacks. Initially, the DLU approach is used as a population initialization model that avoids local optimal solutions by spreading the initial solutions uniformly across the search space. It guarantees that the initial population covers a widespread range of possible solutions by diagonally distributing individuals, increasing the possibilities of exploring different regions of the search space. Secondly, a TAF is included in the position updating phase to improve the global search capability of the model. The TAF introduces the diversity in BOA to explore different regions and directions of the search space, which aids in maintaining population diversity and avoids clustering around local optima. This diversity also avoids premature convergence and improves the model's capability to find globally optimal solutions.

3.6 Classification

Lastly, the HBLSTM is used to classify the classes of BTs based on the optimally selected features. LSTM is a DL network designed as the development of a conventional Recurrent Neural Network (RNN). The LSTM model is superior to the other DL models because the LSTM model can accumulate long-term information without being affected by the existing input or output and has a separate memory cell. This allows them to study long-term dependencies and avoids the problems like vanishing or exploding gradients. It consists of three parts: a forget gate, an input gate, and an output gate; each part has a specific function. It uses random hyperparameters such as bias, weights, learning rate, batch size, etc., to train the learning model, which may require a huge number of evaluations to discover a good solution, particularly in high-dimensional spaces, and it does not take advantage of any prior knowledge or structure in the search space. Therefore, to select the optimal hyperparameters, our system uses the Group Aggregation-centered Honey Badger Optimization algorithm (GAHBO).

In addition, the sigmoid and tanh activation methods are utilized by the existing LSTM network for frequent connections and output. The gradient goes to zero when the input is extremely large or extremely small in the sigmoid and tanh activation. The gradient descent tends to enclose very slow convergence (vanishing gradient problem) when the gradient goes to zero. The proposed system uses the ReLU activation function to resolve these problems instead of the conventional sigmoid as well as tanh activations. The ReLU training speed is six times faster than that of the sigmoid as well as tanh. Hence, ReLU is utilized to attain faster training times and avoid the gradient effect. These improvisations included in the existing LSTM are known as HBLSTM. The structural design of the proposed LSTM is shown in Figure 3.

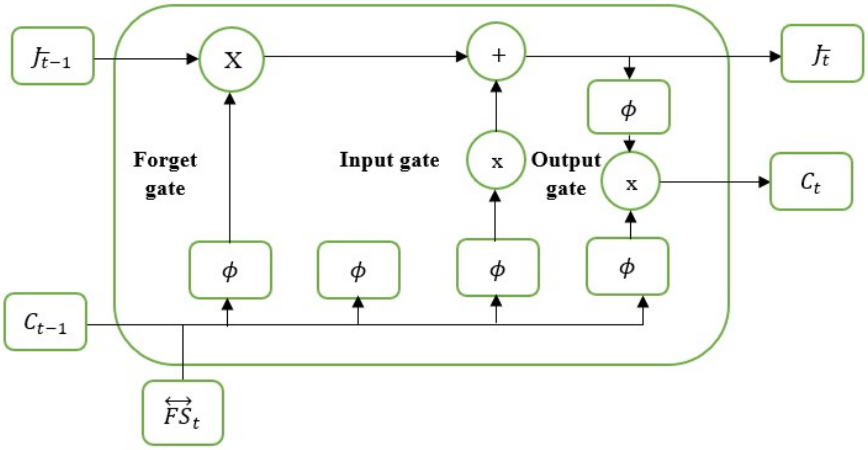


Figure 3: Architecture of the LSTM model

The above structural design mainly comprises three gates. Input gate $\left(\overleftrightarrow{MI}_t\right)$, forget gate $\left(\overleftrightarrow{MF}_t\right)$, and output gate $\left(\overleftrightarrow{MO}\right)$. All gates are activated when input data $\left(\overleftrightarrow{FS}_t\right)$ at the current state as well as output $\left(\dot{\hbar}_{t-1}\right)$ from the hidden state of the preceding layer is entered. The forget gate controls amount of information used for the current state from the previous time steps. The forget gate's output is between 0 and 1. It signals that the information will be eliminated, every time the value is near to zero. There is a tendency to keep more information nearer to oneself. The formula of the forget gate is computed using equation (10).

$$\overleftrightarrow{MF}_t = \varphi\left(\overleftrightarrow{\not\varpi}_{\overleftrightarrow{MF}}\left[\dot{\hbar}_{t-1}, \overleftrightarrow{FS}_t\right] + \breve{\not B}_{\overleftrightarrow{MF}}\right) \tag{10}$$

Where, $\overleftrightarrow{\not\varpi}$ refers to the weight, $\breve{\not B}$ refers to the bias, t refers to the time step, and φ indicates the ReLU activation function as an alternative to sigmoid activation. The ReLU activation function is nonlinear and valuable for eliminating the gradient difficulty which occurs during backpropagation training. The LSTM model uses ReLU to improve the regularization, and the drop out possibility is fixed to a high value to avoid possible overfitting problems. The input will be directly returned by ReLU if the value is greater than 0. If less than 0, then 0 is returned. It is computed as follows:

$$\varphi = \max\left(0, \overleftrightarrow{FS}\right) \tag{11}$$

Then, the input gate uses the transform function to change the value from 0 to 1 that decides which part of input information is updated in the cell state. Where, 1 refers to significance and 0 refers to insignificance. It is computed as follows:

$$\overleftrightarrow{MI}_t = \varphi\left(\overleftrightarrow{\not\varpi}_{\overleftrightarrow{MI}}\left[\dot{\hbar}_{t-1}, \overleftrightarrow{FS}_t\right] + \breve{\not B}_{\overleftrightarrow{MI}}\right) \tag{12}$$

Then, candidate of the ell state $\breve{\not J}_t$ is evaluated, and to reflect the change, the cell state includes the new value.

$$\breve{J}_t = \varphi\left(\vec{\varpi}_{\breve{J}}\left[\dot{h}_{t-1}, \overleftrightarrow{FS}_t\right] + \breve{B}_{\breve{J}}\right) \tag{13}$$

$$J_t = \overleftrightarrow{MF}_t * J_{t-1} + \overleftrightarrow{MI}_t * \breve{J}_t \tag{14}$$

Where, $\breve{J}_t$ refers to the cell state' candidate at time step t. At last, the output gate decides the selection of the next hidden state. Then, the model transforms the new hidden state $\dot{h}_t$ and new memory cell J_t to subsequent time step. It is represented as follows:

$$\overleftrightarrow{MO}_t = \varphi\left(\vec{\varpi}_{\overline{MO}}\left[\dot{h}_{t-1}, \overleftrightarrow{FS}_t\right] + \breve{B}_{\overline{MO}}\right) \tag{15}$$

$$\dot{h}_t = \overleftrightarrow{MO}_t * \varphi\left(J_t\right) \tag{16}$$

Finally, the SoftMax activation function was used to produce a probability distribution over the class, which outputs the BT classes.

1.6.1 Hyperparameter tuning

In the LSTM network, the experts adjusted some hyperparameters, such as weights, bias, learning rate, activation function, batch size, etc., to enhance the learning process. The network hyper-parameter's random adjustment can affect the network presentation and take a larger number of iterations to select the ideal one. Therefore, the GAHBO is used by the proposed system to optimally select the hyperparameters. The honey badger's intelligent foraging motivates the HBO algorithm, which is a modern metaheuristic search technique to find prey. The HB tracks or scents and digs the honeyguide bird to finds the food sources. The initial stage is digging mode; then, a further stage is the honey mode. In the earlier phase, sniffing skills are used to evaluate its prey's position; Honey badgers directly locate beehives in the final model using a honeyguide bird. Even though the HBO algorithm professionally handles large data, it also suffers from local optima problems due to being deficient in local exploitation, global exploration and diversity issues in the search space. This paper uses HBO's group aggregation strategy to overcome these deficiencies, which can avoid local optimal issues by improving their capability to discover the searches are effectively. It also discovers finer solutions to optimize the problems. These improvisations in HBO are described as GAHBO algorithm, and the steps implicated are explained as follows:

Step 1: Initialize the candidate's population (honey badger) solutions randomly, where each solution specifies a set of hyperparameters for the LSTM. Set other algorithm parameters such as population size and maximum number of iteration counts.

Step 2: calculate the fitness of each individual according to Equation (17). In our study, we take the classification error as the fitness, aiming to choose the solution offering lower classification error for the classification BT classes.

$$\begin{aligned} fitness &= Min(classifier\ error) \\ &= Min\left(\frac{No\ of\ misclassified\ samples}{Total\ no\ of\ samples} * 100\right) \end{aligned} \tag{17}$$

Where, $\breve{V}_S$ refers to the total number of samples, $\breve{V}_{TP}$ refers to true positive and $\breve{V}_{TN}$ indicates the true negative, respectively.

Step 3: The intensity $(\breve{I})$ is linked to the target's concentration strength along with the spacing betwixt the honey badger and its target. If the smell's intensity is high, the honey badger's movement is fast towards the target, and vice versa. Smell's intensity is calculated by equation (18).

$$\breve{I}_b = \alpha \times \frac{\breve{\kappa}_L}{4\pi \ddot{d}_b^2} \tag{18}$$

Where, $\breve{\kappa}_L$ and $\ddot{d}$ refers to the smell source's strength as well as the spacing betwixt the honey badger and its target, and α refers to the random number.

Step 4: The HBO location's update process is divided into two parts. The first process is the "Mining stage" in which honey badgers follow similar to Cardioid shape to look for preys as the expression:

$$\breve{Z}_{New} = \breve{Z}_{prey} + \rho_V'' * \psi * \breve{I} * \breve{Z}_{prey} + \rho_V'' * \alpha * \breve{\epsilon}_S * \breve{d}_b * \left|\cos(2\pi\alpha) * (1 - \cos(2\pi\alpha))\right| \tag{19}$$

Where, $\breve{Z}_{New}$ refers to the new position of honey badge, $\breve{Z}_{prey}$ refers to the target location, ψ indicates a constant ability to find food, ρ_V'' denotes the flag that changes the search direction, and $\breve{\epsilon}_S$ represents the group aggregation strategy to improve the local exploration precision, the ability of algorithm exploitation, and accelerates the algorithm convergence. The honey badger will look directly for the optimal food sources surrounded by the population when gathering and foraging in groups. The degree of aggregation slowly increases when they move closer, and the honey badger's range of activity decreases. Based on the honey badger's performance, the specific formula is as follows:

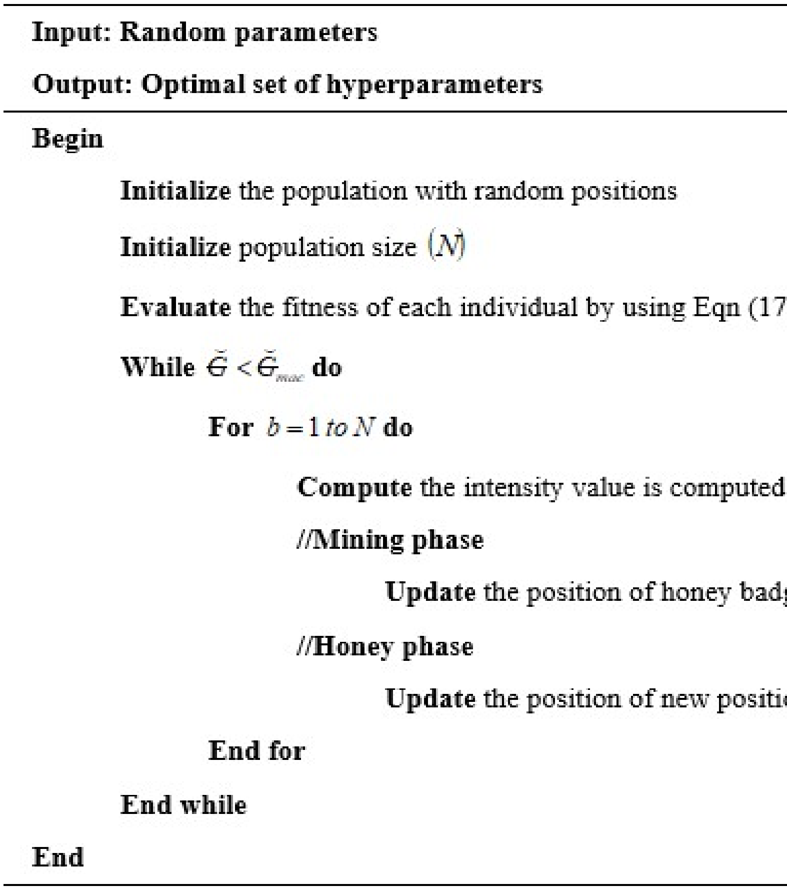
$$\breve{\epsilon}_S = \frac{\arctan\left(2\pi\left(\breve{G}/\breve{G}_{mac}\right)^{0.5}\right)}{\arctan(2\pi)} \tag{20}$$

Where, $\breve{G}$ refers to the current number of iteration and $\breve{G}_{mac}$ denotes the highest number of iterations. When the degree of aggregation was high, the honey badger's range of movement was smaller. Due to the honey badger population's low dispersion at the beginning, the aggregation's degree quickly increased and then transitioned to a minimal increase when it achieved a particular level. Then, the "Honey stage" is the second location update process that locates the honey beehive the badger takes aid from the honey-guide bird and follows its path. It is computed as follows:

$$\breve{Z}_{New} = \breve{Z}_{prey} + \rho_V'' * \alpha * \breve{\epsilon}_S * \ddot{d}_b \tag{21}$$

Where, $\breve{Z}_{New}$ refers to the new position of honey badger's and $\breve{Z}_{prey}$ refers to the target's location.

Step 5: In the end, the highest number of evaluations are reached by the best possible solutions (hyperparameters) of the LSTM. The pseudocode of the proposed HBO algorithm is shown in Figure 4.



```
Input: Random parameters
Output: Optimal set of hyperparameters
Begin
    Initialize the population with random positions
    Initialize population size $(N)$
    Evaluate the fitness of each individual by using Eqn (17
    While $\breve{G} < \breve{G}_{mac}$ do
        For $b = 1 \; to \; N$ do
            Compute the intensity value is computed
            //Mining phase
                Update the position of honey bad
            //Honey phase
                Update the position of new positi
        End for
    End while
End
```

Figure 4: Pseudocode of the proposed HBO

4. RESULTS AND DISCUSSION

Here, an experimental investigation of the proposed as well as existing systems is carried out regarding some performance indicators. The proposed technique is executed in Python with 64-bit Windows 10 OS, Intel (R) Xeon (R) Silver 4210 CPU @ 2.20 GHz (2 processors), 128GB RAM, and NVIDIA Titan RTX. The hyperparameters used in the work are given in Table 2.

Table 2: Hyperparameter settings

SI. No.	Hyperparameters	Values
1	Learning Rate	0.001
2	Optimizer	GAHBO
3	Dropout	0.5
4	Activation function	ReLU
5	Batch size	32

4.1 Dataset descriptions

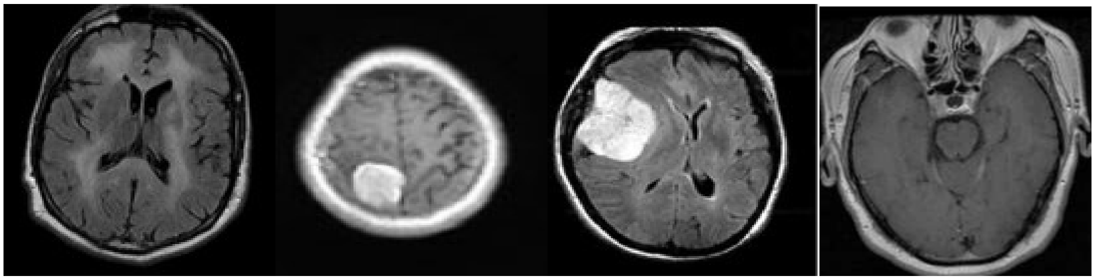
We use the freely available CE-MRI Figshare dataset, accessed through https://figshare.com/articles/dataset/brain_tumor_dataset/1512427, to train and verify the system's efficiency. The tumor classes of the collected dataset include four distinct classes: glioma, pituitary, normal, and meningioma, which are shown in Fig. 4. The dataset consists of a healthy brain (500 images), meningioma (937 images), glioma (926 images), and pituitary gland tumor (901 images).

Normal: These images do not contain any masses or abnormalities.

***Glioma*:** The irregularly formed masses with heterogenous intensity come under this category, and these tumors are more challenging to recognize.

Meningioma: These tumors are usually a rounded mass, well defined, and have a more uniform texture compared to gliomas.

***Pituitary*:** These tumors are located at the base of the brain and are small pea-sized glands. The pituitary gland produces hormones by regulating several bodily functions, which can control growth, reproduction, metabolism, and other vital procedures.

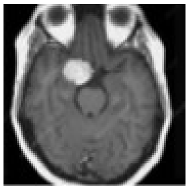
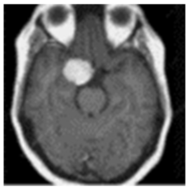
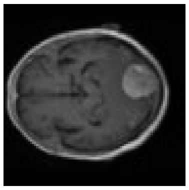
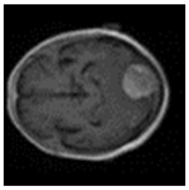


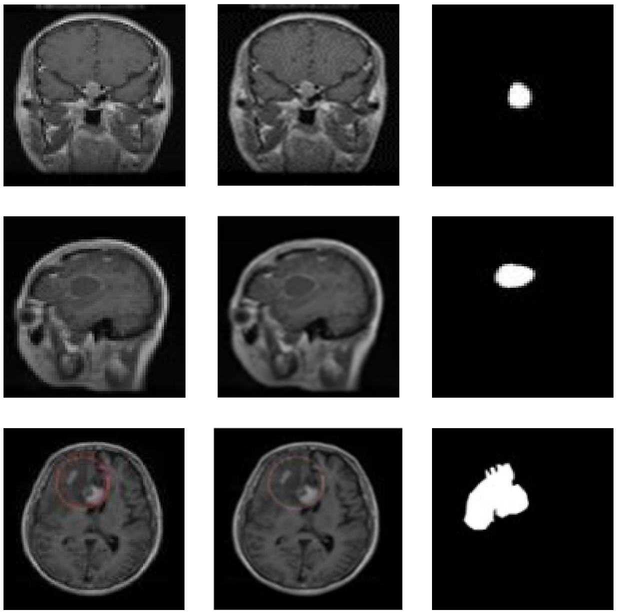
Normal Glioma Pituitary Meningioma

Figure 4: Sample brain MR images from Figshare

Figure 4 shows the sample images form the dataset and table 3 shows the imaging results of the proposed preprocessing and segmentation system for accurate BT detection form various types of brain MRI.

Table 3: Resultant Images

Input image	Preprocessing	Segmentation
		
		



4.2 Evaluation Metrics

Our study uses the following performance indicators to assess the performance of the proposed and existing systems.

i) Accuracy

It is defined as the ratio of correctly identified tumor classes over total number of tumor classes or images in the dataset. It is computed as follows:

$$Accuracy = \frac{\breve{V}_{TP} + \breve{V}_{TN}}{\breve{V}_{TP} + \breve{V}_{TN} + \breve{V}_{FP} + \breve{V}_{FN}} \tag{22}$$

Where, $\breve{V}_{TP}$, $\breve{V}_{TN}$, $\breve{V}_{FP}$, and $\breve{V}_{FN}$ indicates the true positive, true negative, false positive, and false negative values respectively.

ii) Precision

The ratio of accurately detected positive samples over the samples is predicted as positive by the classifier. It is represented as follows:

$$precision = \frac{\breve{V}_{TP}}{\breve{V}_{TP} + \breve{V}_{FP}} \tag{23}$$

iii) Recall

It is the ratio of accurately detected positive instances over the total amount of positive instances in the dataset. It is depicted as follows:

$$recall = \frac{\breve{V}_{TP}}{\breve{V}_{TP} + \breve{V}_{FN}} \tag{24}$$

iv) F1-score

It is a precision and recall's harmonic mean and it is a single metric, which summarizes both metrics. It is mathematically expressed as

$$f1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{25}$$

v) Area under Curve (AUC)

It is a graphical plot of the receiving operating characteristic curve that indicates the trade-off between the true positive rate and the false positive rate under several discrimination thresholds. It ranges from 0 to 1; a value near 1 indicates the classifier's higher accuracy for prediction, and a value near 0 and its variants indicates the classifier's lower prediction performance. It is mathematically formulated as follows.

$$AUC = \int_0^1 TPR \ d\left(FPR\right) \tag{26}$$

vi) ***Geometric mean (G-mean)***

This metric evaluates the performance of the classifiers, particularly when handling imbalanced datasets. It provides a balanced assessment of the classifier's performance across negative and positive classes. It can be computed as follows:

$$G - mean = \sqrt{precision \times recall} \tag{27}$$

vii) Kappa

This statistic is generally utilized to evaluate classification models and inter-rater reliability assessment. It measures the agreement between two sets of ratings for categorical items. A higher kappa value means that the model strongly agrees with the actual labels of the classifier. It is computed as follows:

$$Kappa = \frac{\overleftrightarrow{Q}_V - \overleftrightarrow{\gamma}_V}{1 - \overleftrightarrow{\gamma}_V} \tag{28}$$

Where, $\overleftrightarrow{Q}_V$ and $\overleftrightarrow{\gamma}_V$ refers to the actual values and predicted values, respectively.

viii) Error rate

It is estimated as the ratio of number of all incorrect predictions to the total number of the samples $\left(\overleftrightarrow{S}_P\right)$ in the dataset. It is formulated as follows:

$$Error \ rate = \frac{\breve{V}_{TN} + \breve{V}_{FN}}{\breve{V}_S} \tag{29}$$

Where, $\breve{V}_S$ indicates the total number of instances.

ix) Classification time

It is the time taken by the classifier to classify the tumor classes. It is computed by taking the difference between the prediction completed time $\left(\overleftrightarrow{\zeta}_{CT}\right)$ and the starting time $\left(\overleftrightarrow{\zeta}_{IT}\right)$ and it is represented as follows:

$$CLT = \overleftrightarrow{\zeta}_{IT} - \overleftrightarrow{\zeta}_{CT} \tag{30}$$

4.3 Performance Analysis

In this section, the outcomes of the recommended model with the existing methods such as visual geometry group 16 (VGG16), residual network-12 (ResNet-12), AlexNet, and deep convolutional neural network (DCNN) for the four classes such as Normal, Glioma, meningioma, and pituitary are analyzed. Figure 6 displays the training as well as testing accuracy of the proposed system for 20 epochs. An epoch is the one complete pass via the entire training dataset during the training of a classifier. The classifier uses the entire dataset during training to update its parameters (weights and biases) based on learned features and corresponding labels. The epoch will vary based on the complexity of the dataset to attain the optimal solutions in classification. Our study uses 20 epochs to classify BTs accurately. The model with higher classification accuracy and lower classification loss indicates its effectiveness in accurately detecting classifiers. The training and validation accuracy of the system are 0.521 and 0.725 at 0 epochs. When the epochs increase from 0 to 20, the proposed one achieves 0.998 accuracy.

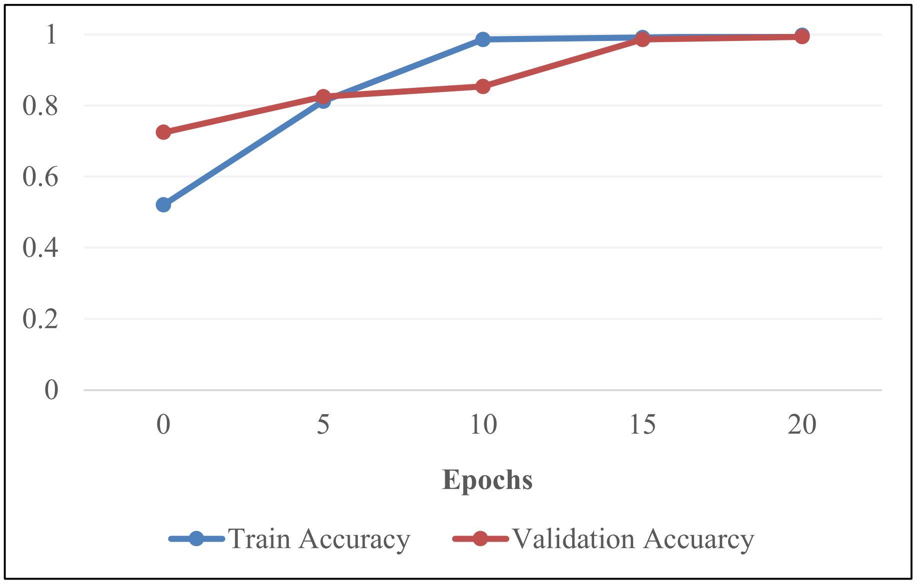


Figure 6: Analysis of training and testing accuracy

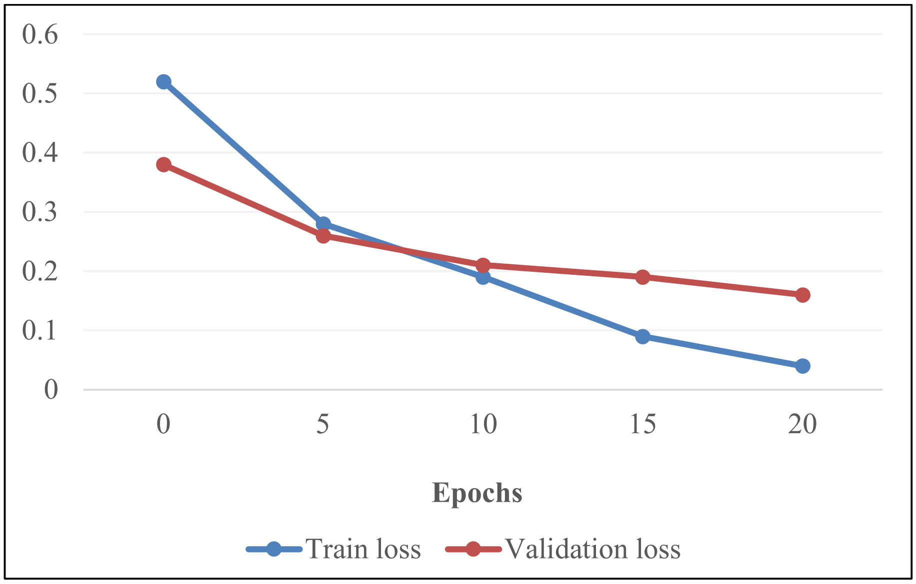


Figure 7: Analysis of training and testing loss

Figure 7 illustrates the training as well as testing loss of the proposed system for 20 epochs. The training and validation loss of the system is 0.52 and 0.38 for the 0 epoch. When the epochs are increased, the loss of the system is decreased. However, the proposed system maintains a lower loss (below 1), showing the model's effectiveness in detection. Thus, the outcomes illustrate that there is no overfitting of the detection model because the accuracy and loss are nearly identical between the training and validation datasets.

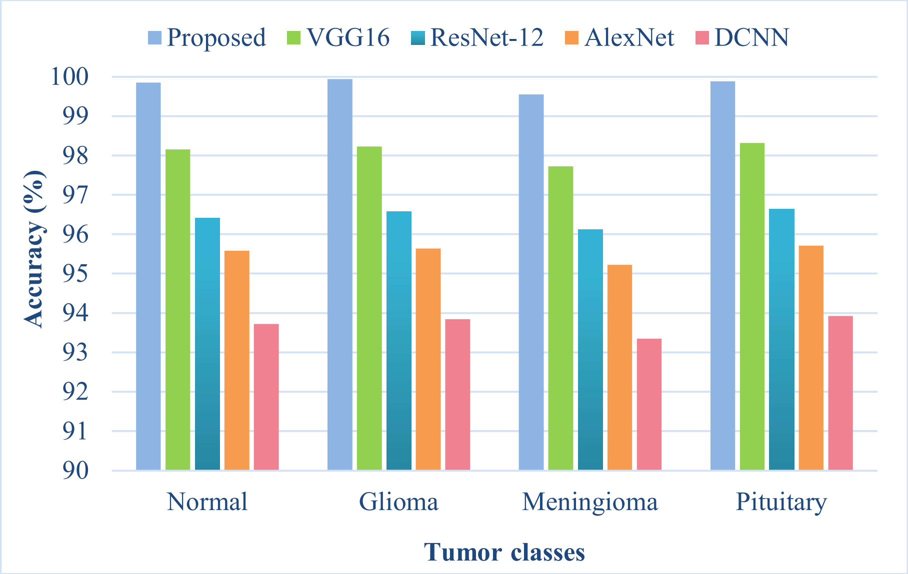


Figure 8: Accuracy analysis

Figure 8 demonstrates the efficiency of the classifiers in terms of accuracy metric. Accuracy is an important metric that shows how the proposed model works on the collected CE-MRI dataset. In this, for normal class, the existing methods such as VGG-16, ResNet-12, AlexNet, and DCNN achieve 98.15%, 96.42%, 95.58%, and 93.72% accuracy, but the proposed one attains the accuracy rate of 99.85%, which is higher than others. Also, for the remaining classes, such as Glioma, meningioma, and pituitary classes, the proposed one offers better classification accuracy compared to the existing methods.

Table 4: Precision, recall, and f1-score analysis

Metrics	Tumor classes	Proposed	VGG16	ResNet-12	AlexNet	DCNN
Precision (%)	Normal	99.92	98.23	96.54	95.69	93.84
	Glioma	99.99	98.35	96.69	95.76	93.96
	Meningioma	99.64	97.82	96.25	95.33	93.48
	Pituitary	99.97	98.45	96.79	95.85	93.99
Recall (%)	Normal	99.76	98.07	96.35	95.49	93.66
	Glioma	99.86	98.15	96.49	95.56	93.79
	Meningioma	99.48	97.67	96.04	95.13	93.28
	Pituitary	99.76	98.24	96.54	95.65	93.84
F1-score (%)	Normal	99.88	98.19	96.45	95.64	93.75
	Glioma	99.97	98.26	96.62	95.69	93.88
	Meningioma	99.58	97.75	96.16	95.26	93.38
	Pituitary	99.91	98.35	96.68	95.74	93.96

Table 4 provides the outcomes of the classifiers regarding precision, recall, and f1-score metrics. The proposed one attains 99.92%, 99.99%, 99.64%, and 99.97% precision for the class labels such as normal, glioma, meningioma, and pituitary. For the same four class labels, the recall and f-score achieved by the proposed system are 99.76%, 99.86%, 99.48%, and 99.76% and 99.88%, 99.97%, 99.58%, and 99.91%. The existing models also show better performance for BT classification; however, compared to our proposed system, the models show lower performance for diagnosing the BTs.

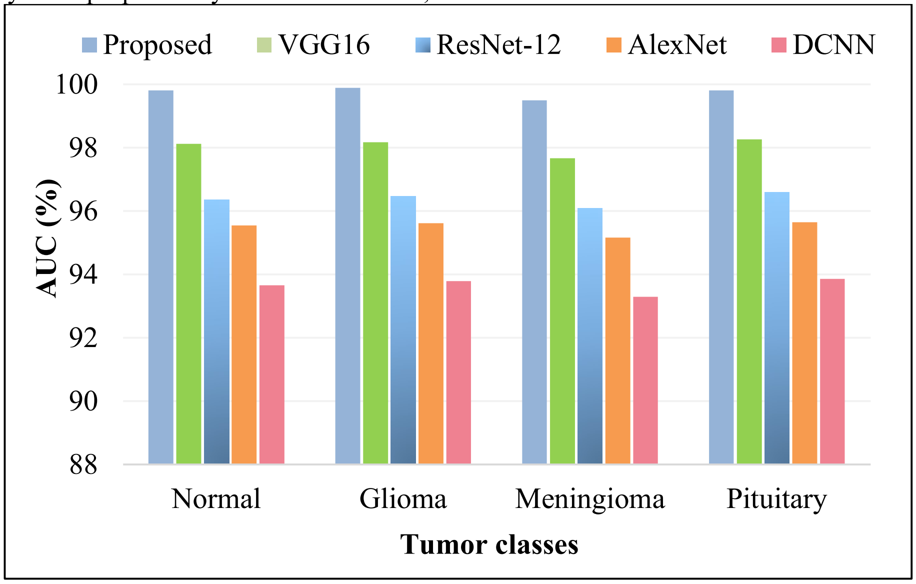


Figure 9: AUC analysis

AUC based performance investigation of the proposed and the prevailing approaches is shown in figure 9. For the normal class, the proposed one achieves 99.81% AUC, which is 1.69%, 3.45%, 4.27%, and 6.16% higher than the existing VGG16, ResNet-12, AlexNet, and DCNN. Also, for the remaining classes such as glioma, meningioma, and pituitary, the proposed one attains an AUC of 99.89%, 99.49%, and 99.81%, which is better than the existing methods.

Table 5: Results analysis of the proposed model

Metrics	Proposed	VGG16	ResNet-12	AlexNet	DCNN
G-mean (%)	99.72	98.02	96.33	95.39	93.58
Kappa (%)	99.65	97.85	96.14	95.12	93.41
Error rate (%)	0.32	0.55	1.76	2.85	3.41
Classification time (s)	74.23	107.42	196.13	241.22	308.18

Table 5 demonstrates the average outcomes of the proposed and the existing methods for BT detection regarding g-mean, kappa, error rate and classification time. In this, the existing models are also kind of pre-trained models, which proffers better accuracy, but comparatively lower than the proposed method because our model comes with several advantages over existing methods like effective feature learning, selection and classification capabilities, which improves the system performance with reduced classification time. For example, the proposed one achieves higher prediction results of 99.72% G-mean. 99.65% kappa and 0. 32% error rate, along with it taking less classification time of 74.23s. Thus, the overall experimental analysis shows that our technique attains better outcomes than the existing methods.

4.4 Comparative Analysis

Here, the comparative assessment of the proposed and existing related works is done regarding detection accuracy, which is shown in table 6.

Table 6: Comparative analysis

Author Name & Ref. No.	Method used	Dataset used	Accuracy (%)
Ours	HBLSTM	CE-MRI	99.81
Usharani Bhimavarapu *et al.* [16]	ELM	Figshare	98.47
Sarmad Maqsood *et al.* [17]	DNN and M-SVM	BraTS 2018 and Figshare	97.47 and 98.92
Mohamed Ait Amou *et al.* [18]	CNN	T1W-CE MRI	98.70
Ejaz Ul Haq *et al.* [19]	DCNN	Figshare and BraTS 2018	97.3 and 96.5
Shaimaa E. Nassar *et al.* [20]	GoogleNet, AlexNet, ShufeNet, SqueezeNet, and NASNet-Mobile	T1W-CE MRI	99
Prince Priya Malla *et al.* [21]	VGGNet	Figshare	98.93
Mohd Anul Haq *et al.* [22]	SE-ResNet-101, SE-ResNet-152, SENet154, ResNet152V2, EfficientNetB0, and EfficientNetB5	CE-MRI	99
Gopal S. Tandel *et al.* [23]	AlexNet, VGG16, ResNet18, GoogleNet, and ResNet50	FLAIR, T2W, and T1W-MRI	98.88, 97.98, and 94.75
Sudhakar Tummala *et al.* [24]	Vision transformers ensembling	T1W-MRI	98.7
Abdullah A. Asiri *et al.* [25]	VGG-19	CE-MRI	98
Syed Muhammad Ahmed Hassan Shah *et al.* [26]	VS-BEAM	T1W-MRI	98.91
Baiju Babu Vimala *et al.* [27]	EfcientNetB0 and EfcientNetB4	CE-MRI Figshare	99.1
Wen Jun and Zheng Liyuan [28]	Attention guided DL	T1W-MRI	98.61
K. Kavin Kumar *et al.* [29]	AlexNet, ResNet 50, and Inception V3	T1W-MRI	96.2
Jiang Linqi *et al.* [30]	SE-ResNeXt	BraTs2017 and BraTs2019	97.45 and 98.99

The above analysis shows that most of the existing works attain highly satisfied performance to classify the BTs, but improvement is still possible by focusing on the major problems like the absence of dataset balancing that leads to biased outcomes for the majority class label of data. Most of the works used several pre-trained CNN models for BT detection and analyzed their performance across diverse datasets. However, the utilization of more pre-trained CNN increases the complexity of the system, and it is time-consuming. The CNN [18 and 29] and its variants [21 and 25] alone are not capable of focusing on relevant parts of the tumor images by avoiding irrelevant ones for classification. Also, the CNN requires a large amount of training data to achieve target accuracy. Most of the works offer up to 98 to 99% accuracy for BT detection, which is good. However, our model achieves higher results than the existing systems.

The reason is that the proposed model initially performs noise removal from the dataset using a Gaussian filtering algorithm, which suppresses the noise and improves the quality of the dataset. Then, it solves the class imbalance issue using the ADASYN method, which prevents the network from having biased outcomes for the majority classes. After that, the segmentation is performed using SC3DUNet, which helps to measure the tumor size, shape, and volume precisely, which is vital for staging and assessing the progression of the tumor. Then, the system performs feature extraction via the SPPXNet model that effectively captures hierarchical and abstract representations of segmented tumor lesions. Then, the system uses the DTBOA algorithm to optimally select the features from the extracted features that only allow relevant features for classification by reducing the computation overhead. Finally, the classification was done based on the HBLSTM algorithm, in which the hyperparameter was optimally chosen via the GAHBO algorithm for optimal performance. In this way, our model achieves remarkable performance over others.

5. CONCLUSION

This paper proposes an HBLSTM-based BT classification system with SC3DUNet model-based tumor segmentation and ADASYN model-based. The system used the CE-MRI fig share dataset to verify its efficacy over existing methods for BT detection. The proposed technique is weighted against the existing methods, namely, VGG16, ResNet-12, AlexNet, and DCNN models for normal, Glioma, Meningioma, and pituitary classes. The evaluations are done with respect to the accuracy, precision, recall, f1-score, AUC, G-mean, kappa, error rate, and classification time metrics. In this, the proposed one achieves an average outcome of 99.81% accuracy, 99.88% precision, 99.72% recall, 99.84% f1-score, 99.75% AUC, 99.72% G-mean, 99.65%kappa, and 0.32% error rate, along with it takes less classification time of 74.23s. Thus, the overall analysis shows that the proposed method achieves outstanding outcomes compared to the existing methods. Our model could be applicable to the early detection of BTs because of its high performance. In future, our work will be validated on diverse datasets with various imaging modalities, tumor types, and patient demographics to evaluate their generalization ability and robustness in prediction.

REFERENCES:

[1]. Siar, M., & Teshnehlab, M. (2022). A combination of feature extraction methods and deep learning for brain tumor classification. IET Image Processing, 16(2), 416-441.

[2]. Saha, P., Das, R., & Das, S. K. (2023). BCM-VEMT: Classification of brain cancer from MRI images using deep learning and ensemble of machine learning techniques. Multimedia Tools and Applications, 82(28), 44479-44506.

[3]. Ahuja, S., Panigrahi, B. K., & Gandhi, T. K. (2022). Enhanced performance of Dark-Nets for brain tumor classification and segmentation using colormap-based superpixel techniques. Machine Learning with Applications, 7, 100212.

[4]. Rammurthy, D., & Mahesh, P. K. (2022). Whale Harris hawks optimization based deep learning classifier for brain tumor detection using MRI images. Journal of King Saud University-Computer and Information Sciences, 34(6), 3259-3272.

[5]. Kumar, A. (2023). Study and analysis of different segmentation methods for brain tumor MRI application. Multimedia Tools and Applications, 82(5), 7117-7139.

[6]. Hossain, A., Islam, M. T., Rahman, T., Chowdhury, M. E., Tahir, A., Kiranyaz, S., ... & Soliman, M. S. (2023). Brain tumor segmentation and classification from sensor-based portable microwave brain imaging system using lightweight deep learning models. Biosensors, 13(3), 302.

[7]. Aggarwal, M., Tiwari, A. K., Sarathi, M. P., & Bijalwan, A. (2023). An early detection and segmentation of Brain Tumor using Deep Neural Network. BMC Medical Informatics and Decision Making, 23(1), 78.

[8]. Samee, N. A., Mahmoud, N. F., Atteia, G., Abdallah, H. A., Alabdulhafith, M., Al-Gaashani, M. S., ... & Muthanna, M. S. A. (2022). Classification framework for medical diagnosis of brain tumor with an effective hybrid transfer learning model. Diagnostics, 12(10), 2541.

[9]. Saeedi, S., Rezayi, S., Keshavarz, H., & R. Niakan Kalhori, S. (2023). MRI-based brain tumor detection using convolutional deep learning methods and chosen machine learning techniques. BMC Medical Informatics and Decision Making, 23(1), 16.

[10]. Zheng, P., Zhu, X., & Guo, W. (2022). Brain tumor segmentation based on an improved U-Net. BMC Medical Imaging, 22(1), 199.
[11]. Ullah, N., Khan, J. A., Khan, M. S., Khan, W., Hassan, I., Obayya, M., ... & Salama, A. S. (2022). An effective approach to detect and identify brain tumors using transfer learning. Applied Sciences, 12(11), 5645.
[12]. Anaya-Isaza, A., & Mera-Jiménez, L. (2022). Data augmentation and transfer learning for brain tumor detection in magnetic resonance imaging. IEEE Access, 10, 23217-23233.
[13]. Alanazi, M. F., Ali, M. U., Hussain, S. J., Zafar, A., Mohatram, M., Irfan, M., ... & Albarrak, A. M. (2022). Brain tumor/mass classification framework using magnetic-resonance-imaging-based isolated and developed transfer deep-learning model. Sensors, 22(1), 372.
[14]. Zoetmulder, R., Gavves, E., Caan, M., & Marquering, H. (2022). Domain-and task-specific transfer learning for medical segmentation tasks. Computer Methods and Programs in Biomedicine, 214, 106539.
[15]. Salim, F., Saeed, F., Basurra, S., Qasem, S. N., & Al-Hadhrami, T. (2023). DenseNet-201 and Xception pre-trained deep learning models for fruit recognition. Electronics, 12(14), 3132.
[16]. Bhimavarapu, U., Chintalapudi, N., & Battineni, G. (2024). Brain Tumor Detection and Categorization with Segmentation of Improved Unsupervised Clustering Approach and Machine Learning Classifier. Bioengineering, 11(3), 266.
[17]. Maqsood, S., Damaševičius, R., & Maskeliūnas, R. (2022). Multi-modal brain tumor detection using deep neural network and multiclass SVM. Medicina, 58(8), 1090.
[18]. Ait Amou, M., Xia, K., Kamhi, S., & Mouhafid, M. (2022, March). A novel MRI diagnosis method for brain tumor classification based on CNN and Bayesian Optimization. In Healthcare (Vol. 10, No. 3, p. 494). MDPI.
[19]. Haq, E. U., Jianjun, H., Li, K., Haq, H. U., & Zhang, T. (2023). An MRI-based deep learning approach for efficient classification of brain tumors. Journal of Ambient Intelligence and Humanized Computing, 1-22.
[20]. Nassar, S. E., Yasser, I., Amer, H. M., & Mohamed, M. A. (2024). A robust MRI-based brain tumor classification via a hybrid deep learning technique. The Journal of Supercomputing, 80(2), 2403-2427.
[21]. Malla, P. P., Sahu, S., & Alutaibi, A. I. (2023). Classification of tumor in brain MR images using deep convolutional neural network and global average pooling. Processes, 11(3), 679.
[22]. Haq, M. A., Khan, I., Ahmed, A., Eldin, S. M., Alshehri, A., & Ghamry, N. A. (2023). DCNNBT: A novel deep convolution neural network-based brain tumor classification model. Fractals, 31(06), 2340102.
[23]. Tandel, G. S., Tiwari, A., Kakde, O. G., Gupta, N., Saba, L., & Suri, J. S. (2023). Role of ensemble deep learning for brain tumor classification in multiple magnetic resonance imaging sequence data. Diagnostics, 13(3), 481.
[24]. Tummala, S., Kadry, S., Bukhari, S. A. C., & Rauf, H. T. (2022). Classification of brain tumor from magnetic resonance imaging using vision transformers ensembling. Current Oncology, 29(10), 7498-7511.
[25]. Asiri, A. A., Aamir, M., Shaf, A., Ali, T., Zeeshan, M., Irfan, M., ... & Alshehri, A. H. (2022). Block-Wise Neural Network for Brain Tumor Identification in Magnetic Resonance Images. Computers, Materials & Continua, 73(3).
[26]. Shah, S. M. A. H., Ullah, A., Iqbal, J., Bourouis, S., Ullah, S. S., Hussain, S., ... & Mustafa, G. (2023). Classifying and localizing abnormalities in brain MRI using channel attention based semi-Bayesian ensemble voting mechanism and convolutional auto-encoder. IEEE Access.
[27]. Babu Vimala, B., Srinivasan, S., Mathivanan, S. K., Mahalakshmi, Jayagopal, P., & Dalu, G. T. (2023). Detection and classification of brain tumor using hybrid deep learning models. Scientific Reports, 13(1), 23029.
[28]. Jun, W., & Liyuan, Z. (2022). Brain tumor classification based on attention guided deep learning model. International Journal of Computational Intelligence Systems, 15(1), 35.
[29]. Nimma, D., Aarif, M., Pokhriyal, S., Murugan, R., Rao, V. S., & Bala, B. K. (2024, December). Artificial Intelligence Strategies for Optimizing Native Advertising with Deep Learning. In 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA) (pp. 1-6). IEEE.

[30]. Dash, C., Ansari, M. S. A., Kaur, C., El-Ebiary, Y. A. B., Algani, Y. M. A., & Bala, B. K. (2025, March). Cloud computing visualization for resources allocation in distribution systems. In AIP Conference Proceedings (Vol. 3137, No. 1). AIP Publishing.

[31]. Kumar, A. P., Fatma, G., Sarwar, S., & Punithaasree, K. S. (2025, January). Adaptive Learning Systems for English Language Education based on AI-Driven System. In 2025 International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-5). IEEE.

[32]. Elkady, G., Sayed, A., Priya, S., Nagarjuna, B., Haralayya, B., & Aarif, M. (2024). An Empirical Investigation into the Role of Industry 4.0 Tools in Realizing Sustainable Development Goals with Reference to Fast Moving Consumer Foods Industry. In Advanced Technologies for Realizing Sustainable Development Goals: 5G, AI, Big Data, Blockchain, and Industry 4.0 Application (pp. 193-203). Bentham Science Publishers.

[33]. Kaur, C., Al Ansari, M. S., Rana, N., Haralayya, B., Rajkumari, Y., & Gayathri, K. C. (2024). A Study Analyzing the Major Determinants of Implementing Internet of Things (IoT) Tools in Delivering Better Healthcare Services Using Regression Analysis. In Advanced Technologies for Realizing Sustainable Development Goals: 5G, AI, Big Data, Blockchain, and Industry 4.0 Application (pp. 270-282). Bentham Science Publishers.

[34]. Alijoyo, F. A., Prabha, B., Aarif, M., Fatma, G., & Rao, V. S. (2024, July). Blockchain-Based Secure Data Sharing Algorithms for Cognitive Decision Management. In 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET (pp. 1-6). IEEE.

[35]. Elkady, G., Sayed, A., Mukherjee, R., Lavanya, D., Banerjee, D., & Aarif, M. (2024). A Critical Investigation into the Impact of Big Data in the Food Supply Chain for Realizing Sustainable Development Goals in Emerging Economies. In Advanced Technologies for Realizing Sustainable Development Goals: 5G, AI, Big Data, Blockchain, and Industry 4.0 Application (pp. 204-214). Bentham Science Publishers.

[36]. Praveena, K., Misba, M., Kaur, C., Al Ansari, M. S., Vuyyuru, V. A., & Muthuperumal, S. (2024, July). Hybrid MLP-GRU Federated Learning Framework for Industrial Predictive Maintenance. In 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT) (pp. 1-8). IEEE.

[37]. Kumar, K. K., Dinesh, P. M., Rayavel, P., Vijayaraja, L., Dhanasekar, R., Kesavan, R., ... & Alhussen, A. (2023). Brain Tumor Identification Using Data Augmentation and Transfer Learning Approach. Computer Systems Science & Engineering, 46(2).

[38]. Linqi, J., Chunyu, N., & Jingyang, L. (2022). Glioma classification framework based on SE-ResNeXt network and its optimization. IET Image Processing, 16(2), 596-605.

[39]. Jyothi P, Dhanasekaran S. An attention 3DUNET and visual geometry group-19 based deep neural network for brain tumor segmentation and classification from MRI. J Biomol Struct Dyn. 2023 Nov 18:1-12. doi: 10.1080/07391102.2023.2283164.