Electronic Research Archive

http://www.aimspress.com/journal/ERA

ERA, 31 (5): 2900–2924.
DOI: 10.3934/era.2023146
Received: 30 November 2022
Revised: 03 March 2023
Accepted: 06 March 2023
Published: 16 March 2023

*Research article*

# Brain tumor detection using CNN, AlexNet & GoogLeNet ensembling learning approaches

**Chetan Swarup[1], Kamred Udham Singh[2], Ankit Kumar[3,*], Saroj Kumar Pandey[3], Neeraj varshney[3] and Teekam Singh[4]**

[1] Department of Basic Science, College of Science and Theoretical Studies, Saudi Electronic University, Riyadh-Male Campus, Riyadh 11673, Saudi Arabia
[2] School of Computing, Graphic Hill Era University, Dehradun-248002, India
[3] Department of Computer Engineering & Applications, GLA University, Mathura, India
[4] Department of Mathematics, University of Petroleum & Energy Studies, Dehradun-248002, India

* **Correspondence:** Email: kumar.ankit@gla.ac.in.

**Abstract:** The detection of neurological disorders and diseases is aided by automatically identifying brain tumors from brain magnetic resonance imaging (MRI) images. A brain tumor is a potentially fatal disease that affects humans. Convolutional neural networks (CNNs) are the most common and widely used deep learning techniques for brain tumor analysis and classification. In this study, we proposed a deep CNN model for automatically detecting brain tumor cells in MRI brain images. First, we preprocess the 2D brain image MRI image to generate convolutional features. The CNN network is trained on the training dataset using the GoogleNet and AlexNet architecture, and the data model's performance is evaluated on the test data set. The model's performance is measured in terms of accuracy, sensitivity, specificity, and AUC. The algorithm performance matrices of both AlexNet and GoogLeNet are compared, the accuracy of AlexNet is 98.95, GoogLeNet is 99.45 sensitivity of AlexNet is 98.4, and GoogLeNet is 99.75, so from these values, we can infer that the GooGleNet is highly accurate and parameters that GoogLeNet consumes is significantly less; that is, the depth of AlexNet is 8, and it takes 60 million parameters, and the image input size is 227 × 227. Because of its high specificity and speed, the proposed CNN model can be a competent alternative support tool for radiologists in clinical diagnosis.

**Keywords:** image segmentation; pattern classification; MRI images; tumor; PCA; SVM

## 1. Introduction

According to the World Health Organization (WHO), cancer is the leading cause of death worldwide [1]. Although it is sometimes possible, early cancer detection does not permanently save lives. A tumor, as opposed to cancer, can be benign, pre-carcinoma, or cancerous. Friendly tumors differ from censures because they do not usually spread to other organs or tissues and can be accurately eradicated [2]. Clinical imaging encompasses a wide range of procedures that can be used as non-invasive methods of peering inside the body [3]. Clinical imaging combines a number of imaging modalities and cycles to see the edge for therapeutic and diagnostic purposes, and it then plays an important and decisive role in directing actions that improve people's welfare. Picture division may be used to achieve a higher level of image preparation, which could be a significant and critical breakthrough in image handling [4].

The main goals of image division in clinical image processing are to identify tumors or sores, use useful machine vision, and obtain reliable results. Improving the affectability and explicitness of tumors or injuries in clinical imaging has been a top priority with the use of computer aided diagnostic (CAD) frameworks. States that brain and other framework nervous cancer is the tenth leading cause of death and that the five-year survival rate for people with ruined brains is 34% for men and 36% for women. Furthermore, the WHO predicts that 400,000 people worldwide will be affected by brain tumors in the coming year, with 120,000 deaths. Furthermore, an additional 86,970 cases of nonthreatening brain and other central nervous system (CNS) malignancies are expected to be studied in the United States in 2019 [5]. A brain tumor develops when abnormal cells grow within the brain [6]. Tumors are classified as either malignant or benign. Brain tumors are dangerous because they start inside the brain, grow quickly, and wreak havoc on the surrounding tissues. It has the potential to spread to other areas of the brain and have an impact on the focal structure. There are two types of brain metastasis malignancies: those that develop within the brain and those that have spread from somewhere else. A benign neoplasm, on the other hand, could be a collection of cells that develops gradually within the brain. Early detection of brain cancer and a higher level of endurance plausibility will now be critical in improving treatment outcomes. However, because MRI images are produced in excess of daily clinical practice, manual division of tumors or sores may be laborious, demanding, and uncomfortable. Reverberation is another name for X-rays. The primary goals of imaging are to detect tumors or damage. Tumor division by MRI is one of the most essential steps in clinical image processing because it typically requires a large amount of data. Furthermore, cancers with delicate tissue restrictions were challenging to understand. As a result, determining the precise division of tumors in the human brain is a daunting task. We developed a competent approach, aided by a convolutional neural network, that aids in the division and identification of the neoplasm without the need for human intervention.

### *1.1. Problem statement*

Brain Tumor division is one of the principal vital and challenging errands inside the landscape of the clinical picture; preparing as a human-helping manual characterization may bring about inaccurate forecasts and analysis. Moreover, it's an exasperating errand when there's a larger-than-usual measure of information present to be helped. Since brain tumors have a wide variety for all intents and purposes, and there is a similitude between tumors, and typical tissues, a productive and powerful framework is required for extracting tumor districts from pictures.

### *1.2. Motivations*

The primary point here is to carry out a method of programmed detection of neoplasm utilizing convolutional neural network (CNN) utilizing MRIs as tests. CNN is utilized to make the model and train it utilizing past tumor patients' records and utilizing that model to anticipate a substitution picture if it's infected or not.

The significant Contribution of the paper is as follow:

1) A pipeline based on ensemble DL is presented for efficient and cost-effective tumor identification in the brain.
2) An ensemble method is proposed by using four distinct deep learning architectures (CNN, AlexNet, ResNet50, VGG19, to establish the optimal recognition system by selecting the best-fitting Deep Learning architecture parameters.
3) In this paper, we comprehensively analyze the effectiveness of key variables affecting pertained systems' tuning.
4) We have compared the proposed models utilizing several critical characteristics for Brain Tumor detection in our research.

## 2. Related work

This section reviews the research that different researchers in this area have already done. We also identify any gaps that have yet to be filled. Brain Tumor division is everything about chief, significant and exhausting assignments inside the landscape of clinical picture handling as a human-helped manual order may bring about incorrect forecasts and analysis. It's also irritating when a large amount of data is to be sorted. Extraction of tumor areas from pictures becomes unassailable because brain tumors are varied for all intents and purposes and share similarities with normal tissues. We proposed a method using a Fuzzy C-Means bunching calculation followed by conventional classifiers and a convolutional neural architecture to detect malignancies using 2D reverberation brain images (MRI). The test research used a consistent dataset with various tumors, sizes, regions, forms, and picture capabilities.

In this research, we this research, we examine deep convolutional neural networks (ConvNets) [7] for brain tumor classification. Using MRI patches, slices, and multi-planar volumetric slices, we show three ConvNets that were trained entirely from scratch. The findings show that ConvNet improves accuracy in all cases when the model is trained on multi-planar volumetric data. With no additional effort put into feature extraction and selection, as in conventional models, it achieves a testing accuracy of 97%. We contrast our results with cutting-edge systems that rely on human feature engineering to finish the job. It demonstrates that ConvNets may improve the grading accuracy by up to 12%. By observing the results of the intermediate layer, we also look at self-learning kernels and filters at different levels.

The top-tier MRI-based brain tumor division procedures are described in detail in this study [8]. Because of the noninterfering and extremely sensitive tissue separation of MRI, a large number of frontal brains tumors division methods function on MRI images and employ a variety of attributes to gather and organize information, as well as consider spatial information in very near proximity. The motivation behind these techniques is to allow for a critical decision on tumor detection, detection, and therapy. Furthermore, it allows for reliable outcomes within a reasonable estimated time. This research presents new methodologies for MRI images of a patient's brain. Gaussian, which may be a direct route, completes the preprocessing here. GLCM highlights perform highlight extraction for the pictures at that point. Finally, the characterization was carried out using a calculation called convolutional neural organization, which can detect tumor regions. Brain tumor detection could be excellent assistance for doctors and help for clinical imaging and ventures performed on gathering CT output and MRI images. This paper [9] discusses tumor detection and extraction. The area is portioned, and therefore the evaluation of the tumor's personality using the gadget indicated here aids the specialists in determining the treatment plan production and monitoring the tumor's state. The advantages of this method are that it enhances the picture's division level and spatial confinement, as well as its efficacy, compared to the alternative framework. It consumes less of an ideal chance for calculation and makes mentoring easier with fewer boundaries than other companies. The framework's exactness will be relatively improved by utilizing an artificial neural organization given the classifier. The mechanized detection strategy created here was made out of principle with three stages: (1) Pre-preparing, (2) Classification using CNN, and (3) post-handling.

We demonstrate a new CNN model for three different forms of brain tumors [10]. The built network has been thoroughly evaluated on T1-weighted differentiation-enhanced reverberation pictures and is less complex than successfully used pre-prepared networks. The organization's presentation was assessed using four procedures: two 10-overlay cross-approval processes and two information bases. Subject-wise cross-approval, one of the 10-overlay methods, was used to assess the organization's capacity for speculation, and a more extensive picture data set was used to test the improvement. The record-wise cross-approval for the raised informational index yielded the best result for the 10-overlay cross-approval methodology, with an exactness of 96.56%. Due to its high speculation capacity and speed, the recently developed CNN design may be an elective choice aid device for radiologists in clinical diagnostics. They presented a CNN model considering local and logical data [11]. A preplanning stage standardizes the photographs, and a post-preparing phase eliminates the fake positives. Another study for neoplasm diagnosis proposes a crossover strategy combining neutrosphere and CNN. The demonstration of the concept outperforms traditional CNN, SVM, and K-closest neighbor (KNN) methods by 95.62%. This study uses the speedier R-CNN approach to analyze malignancies inside MRI brain images to find and identify them. Additionally, the different tumors are broken down into one of three tumor classes, such as meningioma, glioma, or pituitary tumors. Characterization is performed by this approach more rapidly and precisely than by regular R-CNN, which is why it was chosen.

Brain tumor detection is done utilizing MRI and dissecting it [12]. AI might be an incredible system for detecting malignancy tumors on MRI. They achieved a preparation accuracy of 99% and an approval accuracy of 98.6% in this location, with approval misfortune ranging from 0.704 to 0.000 for more than 35 years. This model was made on the CPU-based tensor stream, and the GPU rendition of the tensor stream is far quicker to mentor, which can prompt a lot quicker model creation. Here, convolutional neural network (CNN) calculation is utilized. The following steps are information procurement, expansion, model creation, etc. The system that appeared during this paper could be an essential picture order strategy for Lenet architecture. The more unique methodologies are accessible.

In this paper [13], we investigate deep convolutional neural networks (Con-vNets) for the structural organization of brain tumors using multisession MR images. For MRI patches, cuts, and multi-planar volumetric cuts, we present three ConvNets that are constructed without any preparation. In scenarios when the model is created on a multi-planar volumetric dataset, the results show that ConvNet achieves greater precision all around. It achieves a testing precision of 97 percent with no additional effort put into highlight extraction and determination, depending on the circumstances in ordinary models. We also compare our results to cutting-edge approaches requiring manual component design. It indicates a dramatic improvement of 12 percent in ConvNets evaluating execution. Using a representation of the yields from the halfway layer, we further examine the characteristics of self-learned parts and channels at different levels.

In this study [14], we develop a system for classifying fresh brain MRI scans into those with and those without tumors. It should be done with no human intercession. To utilize a few assortments of classifiers, we preferred to preprocess a few parts of the photos like the shading, space of interest, picture record expansion, and different level. We utilized two mainstream devices to accomplish this, viz. ImageJ and MATLAB. We removed the preeminent significance and separated the highlights of the preprocessed pictures a short time later. During this stage, we remove ten distinct highlights. At last, we utilize an apparatus called WEKA 3.6 to utilize four distinctive grouping calculations on these highlights and ascertain the exactness/review, the F-measure, the portion of accurately arranged pictures, and the time taken to make each model. One of the basic techniques wanted to identify tumors [15] inside the brain is reverberation imaging (MRI). It gives essential data utilized in the technique for examining the internal design of the actual construction indeed. Because of the variety and complexity of brain tumors, clustering MR images is anything but straightforward. The suggested approach for detecting a tumor in MR images includes sigma filtering, flexible edge, and detection site. The suggested technique employs two classifiers that rely on controlled strategies: the first classifier was the C4.5 decision tree calculation, followed by the multi-layer perceptron (MLP) calculation. The classifiers categorize the braincase as typical or uncommon; one benign tumor type and five different tumor types may be found in the bizarre brain. Considering 174 samples of brain MR images and using the MLP computation, the most extreme precision of nearly 95% is achieved.

The author proposed a method [16] for automated brain tumor diagnosis that uses a computer-assisted system to improve accuracy and efficiency compared to manual diagnosis. The process involves preprocessing the brain MRI image, obtaining tumor proposals using an agglomerative clustering-based method, and transferring the proposals to a backbone architecture for feature extraction. Refining the proposals through a refinement network, aligning the refined proposals to the same size, and finally, using a head network for classification. The method was tested on a publicly available brain tumor dataset and showed an overall classification accuracy of 98.04%, outperforming existing approaches. The model achieved high accuracy and sensitivity in the classification task. In this paper, the author[17] proposed a method for detecting brain tumors using an automated magnetic resonance imaging (MRI) technique. The process includes preprocessing the MRI images, using two different deep learning models to extract features, combining the features into a hybrid feature vector using the partial least squares (PLS) method, and revealing the top tumor locations via agglomerative clustering. Aligning the proposals to a predetermined size and finally using a head network for classification. The method showed a high classification accuracy of 98.95%, outperforming existing approaches. The proposed technique can potentially improve the accuracy and efficiency of brain tumor diagnosis.

## 3. Material & method

The convolutional neural network is often used to preprocess medical image data. Many scientists spent a long time developing a model that might better detect tumors. We wanted to develop a reliable approach for appropriately categorizing tumors from 2D brain MRI scans. Although a wholly linked neural structure may identify tumors, we chose CNN for our model due to border sharing and association sparsity [18] shown in Figure 1.

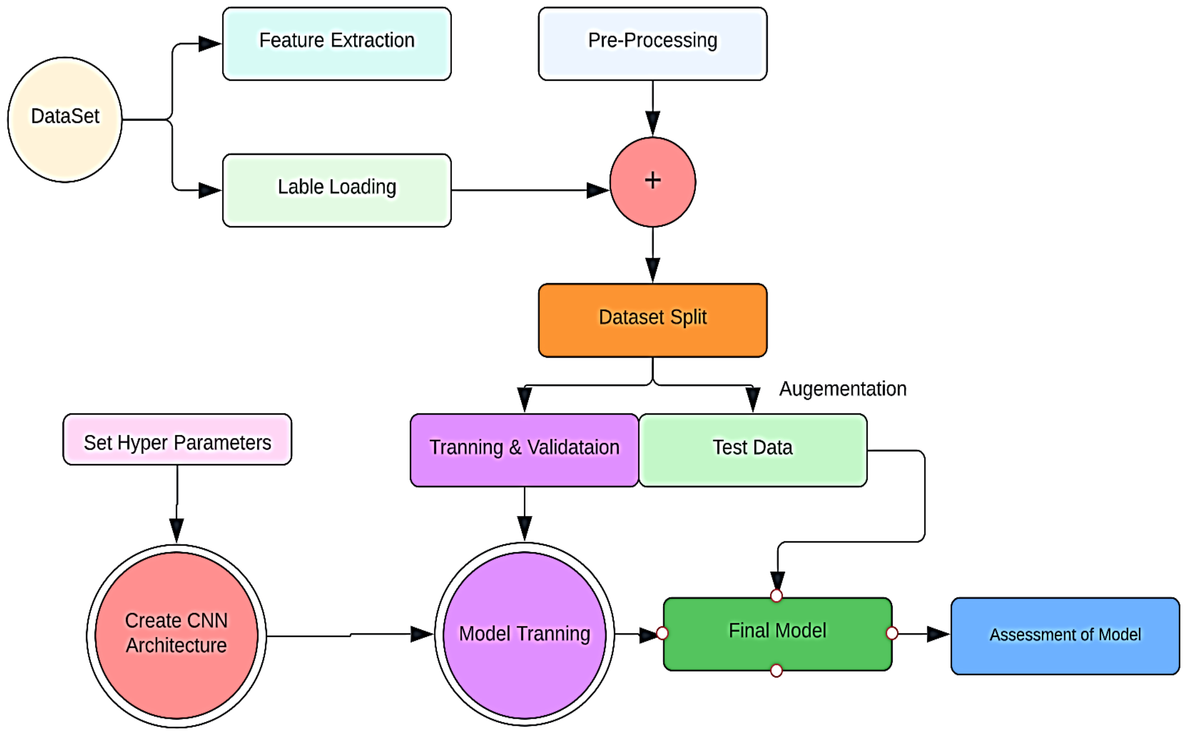

**Figure 1.** Flow diagram of proposed methodology for tumor detection.

A 5-layer convolutional neural network is described and utilized for tumor localization. The model considers the hidden layers and provides the first tangible evidence of the tumor's concern. Highlight extraction is usually carried out by several convolutional modules that make up a CNN. Each module starts with a convolutional layer and ends with a pooling layer [19]. The last convolutional module is followed by at least one thick layer that conducts characterization. To establish a value between 0 and 1 for each hub, one for each target class within the model (i.e., each of the conceivable classes the model may forecast), the last thick layer in a CNN has the SoftMax initiation ability (the amount of these SoftMax values is satisfactory to 1). One way to conceptualize the SoftMax [20] values for a given picture is as relative estimates of the chance that the image belongs to each target class shown in Figure 2.

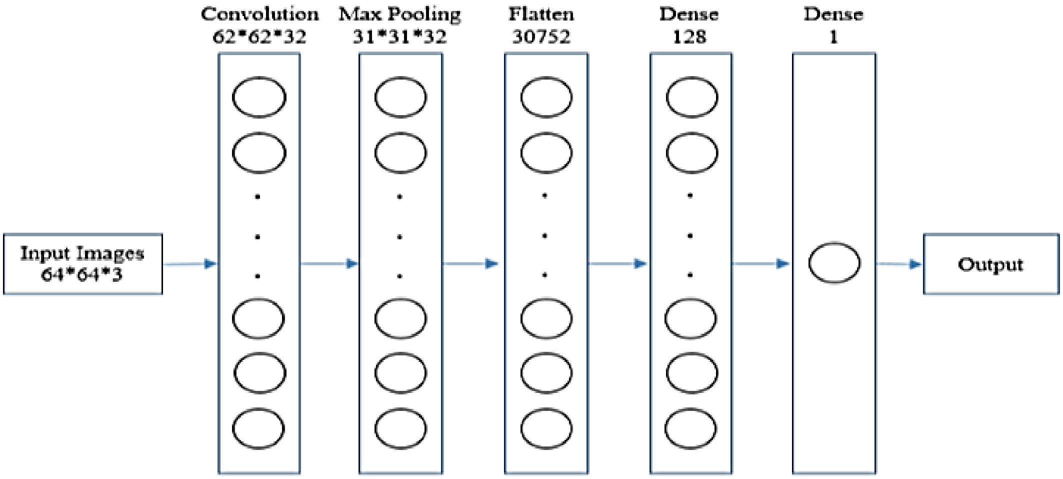

**Figure 2.** Using a 5-Layer convolutional neural network, a method for tumor detection has been proposed.

Using the convolutional layer in conjunction with the fledgling layer, an information state of the MRI images of 64 × 64 × 3 is created, converting each snapshot into a homogenous measurement. After assembling all images at the same angle, we used 32 convolutional channels, each measuring 3 × 3, and three channel tensors to produce a convolutional component entangled with the information layer. ReLU does not need to prove the yield since it is utilized as an enactment task. Reduce the piece of bounds and computational season of the organization by logically abbreviating the spatial size of the representation in this ConvNet model [21]. The pollution of overfitting may also be costed to the Brain MRI image, and our Max Pooling layer flawlessly works for this distinction. We employ MaxPooling2D as the model for spatial information that proves our information picture. This convolutional layer uses measures of 31 × 31 × 32 [22]. The pool size is (2, 2) due to the separation of the information images in both spatial dimensions, which recommends a tuple of two integers to downscale in an upward direction and to a level plane. A pooled highlight map is acquired after the pooling layer. After pooling, one of the most critical layers is leveling. It is essential for planning since we must improve the framework by addressing the information photos in a single vector section. The Neural Network was then tasked with processing, as shown in Figure 3.

We used two layers that were entirely intertwined. Dense-1 and Dense-2 addressed the thick layer. In Keras, the thick capacity is used to handle the Neural Network and the gotten vector functions as a contribution to this layer. Inside the hidden layer, there are 128 hubs. We kept it as low as possible since the number of measurements or hubs we need is proportional to the processing assets we need to match our model, and 128 hubs produce the most liberal outcome. Because the beginning job requires improved intermingling execution [23], ReLU is used. Following the first thick layer, the model's final layer was created using the second entirely associated layer. We used the sigmoid capacity as an enactment job where the total number of the hub is one because we wanted to reduce the number of processing assets used so that a more significant sum reduces the execution time. Even though the actuation work, we scale the sigmoid capacity, and the quantity of hubs is far fewer and easier to deal with for this profound organization [24], there is a probability that the preparation of profound organizations for using the sigmoid will be hampered. The suggested CNN model's functional evolution is seen here.

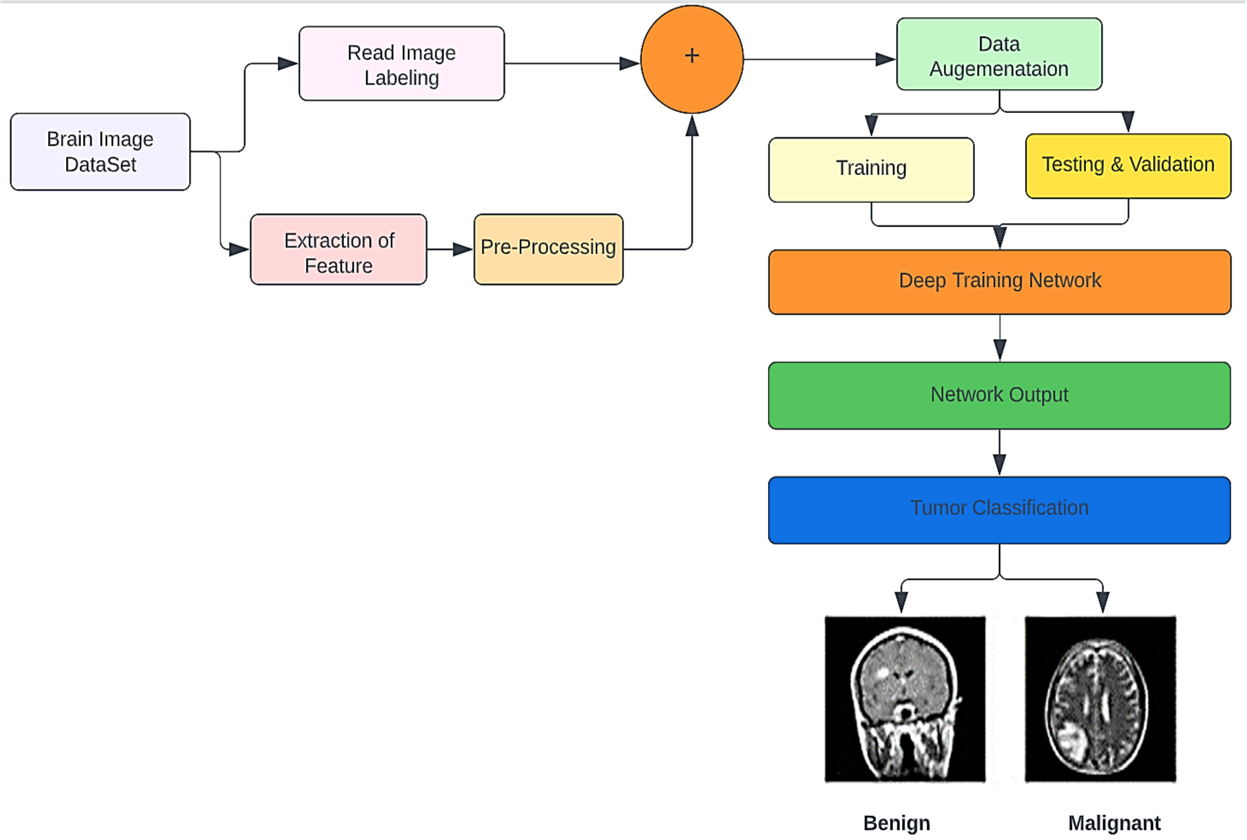

**Figure 3.** Proposed model progressing in its level.

*3.1. Data augmentation*

Information expansion could be a methodology that grants professionals to fundamentally build the scope of data accessible for preparing models without gathering new information. Since this can be a small dataset, there were short guides to mentor the neural organization. Additionally, the information increase helped handle the data unevenness issue inside the information. Before the information increases, the dataset (RADHAMADHAB DALAI, July 1, 2021, "Brain Tumor Dataset", https://dx.doi.org/10.21227/2qfw-bf10.) comprised of:

*i.* 155 positive and 98 negative models, prompting 253 model pictures [25].
*ii.* After the information increase, presently the dataset comprises of:
*iii.* 1085 positive and 980 models, resulting in 2065 model images.
*iv.* Note that these 2065 models also include 253 unique images.

*3.2. Data preprocessing*

The following image preprocessing techniques are used to enhance the image quality:

1) **Rescaling:** Rescaling the image to a desired size is often one of the first steps in preprocessing. This step is essential to ensure that the image has a consistent size, which makes it easier to process.
2) **Denoising:** Removing noise from the image can improve its visual quality and make it easier for the algorithm to process. There are various techniques for denoising an image, including

Gaussian and median filtering.

3) **Color Correction:** Adjusting an image's brightness, contrast, and saturation can improve its visual quality and make it easier for the algorithm to identify patterns and features.
4) **Rotations and translations:** Images may need to be rotated or translated to align with a reference image. It is crucial in applications such as medical imaging, where the images must be registered accurately to obtain meaningful results.
5) **Image segmentation:** Image segmentation is the process of dividing an image into multiple regions, or segments, based on the similarity of pixel values. It is an essential step in object recognition and classification tasks.

The following preprocessing procedures were conducted for each image:

Remove the area of the image that solely depicts the brain (which is the image's most fundamental piece). Adjust the image's size such that (240, 240, 3) = (picture width, picture height, and the number of channels) is the state: since the dataset's photos arrive in different sizes. As a result, all images should have an indistinguishable form to ensure that they are taken care of as a contribution to the neural network. Use standards to scale pixel values from 0 to 1.

### *3.3. Data split*

The information was divided into the following categories:

Information for training = 70%

Information for validation = 15%

Information for testing =15%.

CNNS contains different layers. Each layer has a different function.

**Convolution Layer**: This is where the learning process occurs; it computes the complications between the neurons and the different patches in the input. Image data is stored in a 4D tensor [26] (A tensor is a multidimensional array of components that describe functions relevant to the coordinates of a space), usually processed by 2D convolutional layers.

**Pooling Layer:** This layer decreases the network's count of parameters (weights). A model that fits the training data too well is said to overfit. The model becomes so familiar with the details of the data and the noise in the training data that it performs poorly on the new data. This layer also increases the network's dependability [27]. The pooling layer conserves the essential characteristics while reducing the size of the image and is mainly placed between two convolution layers.

Flattening Layer: Neural networks can learn only 1D data; this layer converts the 2D data (tensor/array) into 1D data.

**Fully Connected Layer**: Here, the input image from the previous layer is fed to the FC layer [28]. This is the last layer placed before the output layer, which comprises the weights along the network's neurons. This layer provides help in connecting the neurons between two different layers. A nonlinear combination of these parameters can be learned by attaching a fully connected layer that is feasible and cost-effective.

**Activation Function:** The activation function's significant parameters of the convolutional neural network model. There are several activation functions, such as SoftMax, ReLU, Sigmoid functions, etc., and now we will discuss the two most adaptable activation functions for our complications.

### *3.4. ReLU (rectified linear unit) correction layer*

Applying any activation function to the output of the preceding layer is the duty of the ReLU Correction Layer. It adds nonlinearity to the network. These activation functions are used to learn and find the accuracy of the continuous and complex relationships between network variables.
Usually defined as ReLU(i)=max (0, i). Visually represented in Figure 4.

f(i)=max (0, j)

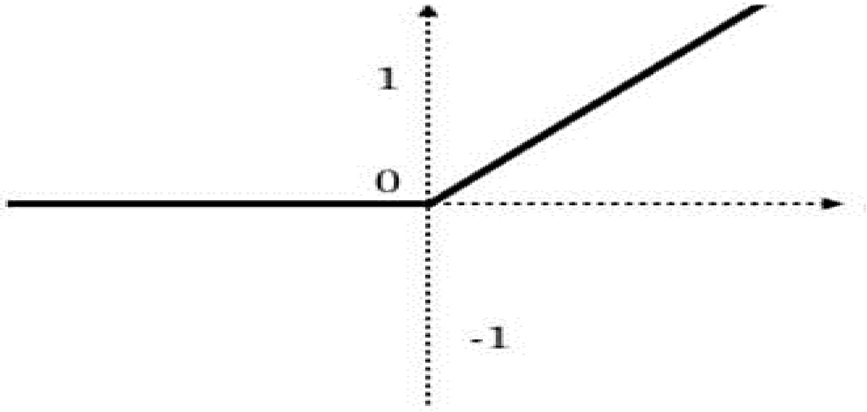

**Figure 4.** ReLU function.

**SoftMax Layer:** SoftMax extends ideas into a multiclass world related to cross-entropy functions. This layer intends to test the model's efficiency by employing a loss function and the cross-entropy (used to measure the randomness in the information being processed) function to maximize the network's performance.
SoftMax function:

$$\boldsymbol{s_i = \frac{exp(a_i)}{\sum_{b=1}^{b=n} exp\ (b_i)}} \tag{1}$$

For i = 1, 2, 3…….., n.

Any neural network's final activation function, the SoftMax function, is often employed to normalize the output of a network.

## 4. Proposed neural network architecture

Architecture detail: Each × (picture) input is supplied into the neural network with a shape of (240, 240, 3). It then moves on to the next layer:

1) A layer with no cushioning and a swimming pool size of (2, 2).
2) A 32-filter convolutional layer with a stride of at least 1 and a filter size of (7, 7) [29].
3) A batch normalization layer that normalizes pixel data to speed up calculations.
4) A ReLU-activating layer.
5) A Max Pooling layer with f and s both equal to four.
6) A Max Pooling layer with f = 4 and s = 4 is used as previously.

7) A layer that flattens the three-dimensional matrix so that it becomes a one-dimensional vector.
8) A dense (output unit) layer that is entirely coupled and has one neuron with sigmoid activation (since this is often a binary classification task).

The training process involves updating the model's parameters to minimize the difference between the model's predicted outputs and the actual outputs (i.e., the labels) in the training data. This process is known as supervised learning and is typically performed using a stochastic gradient descent (SGD) variant. The choice of hyperparameters, such as the learning rate, the number of layers in the model, the number of attention heads, and the model's size, among others, can significantly impact the model's performance. Finding the optimal hyperparameters is often performed through hyperparameter tuning, which involves training multiple models with different hyperparameters and comparing their performance on a validation set. The best-performing model is then selected and used for inference. The number of hyperparameters that can be adjusted and the optimal values for each can vary greatly depending on the specific task and dataset being used. Additionally, many other techniques can be used to improve the performance of language models, such as transfer learning, fine-tuning, and data augmentation.

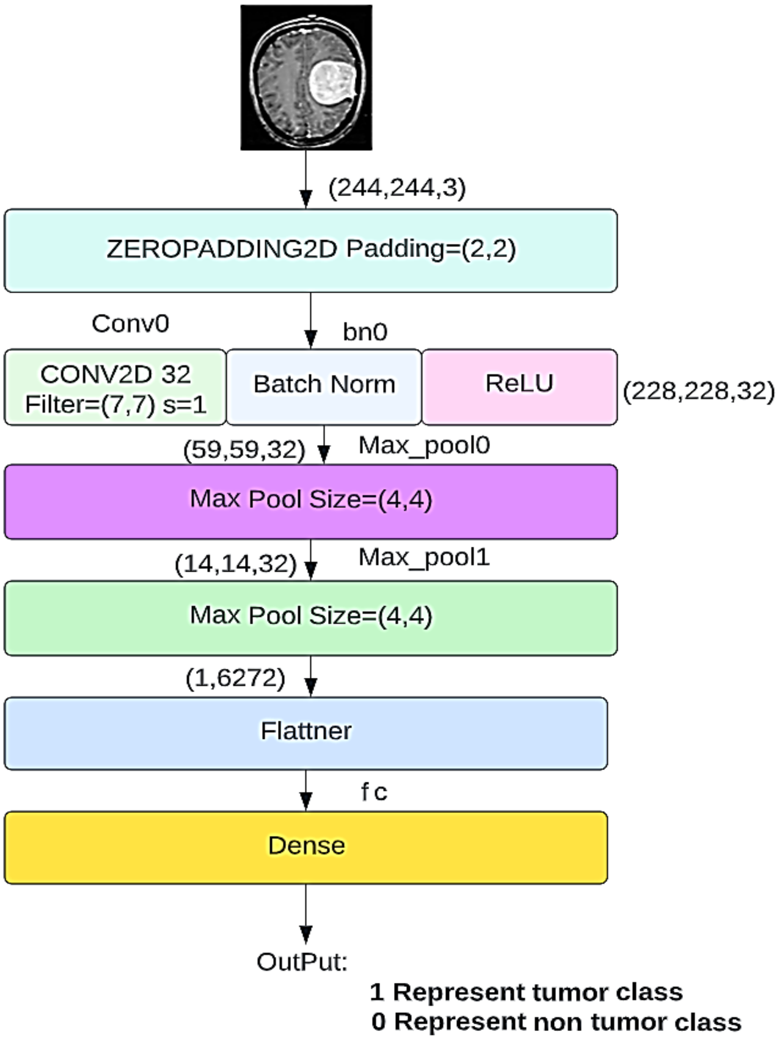

**Figure 5.** Neural network architecture.

**Table 1.** The performance measure of AlexNet for our data set in percentage.

| Factor(s) | Values |
|---|---|
| Number of convolutional layers | 32-filter convolutional layer with a stride of at least 1 and a filter size of (7, 7) |
| Number of cross-channel normalization | 1, 2, 3 |
| Number of drop out layers | 1, 2, 3 |
| Maximum epochs | 20,40,50,60,80,100 |
| Number of fully connected layers | 1,2, 3 |
| Number of convolutional kernels | 8, 16, 32, 64, 128, 256 |
| Kernel size | 1,3, 5, 7, 9, 10, 11 |
| pooling layer | Max pooling, average pooling |
| pooling layer window size | 2, 3,4, 5 |
| Optimizers | GoogLeNet, Adam, VGG |
| Mini-batch size | 1, 4, 8, 16, 32, 64, 128 |
| Dropout rate | 0.1, 0.15, 0.2, 0.25, 0.5 |
| Initial learning rate | 0.01, 0.001, 0.0001 |
| Learning rate drop factor | 0.1, 0.2, 0.3 |

### *4.1. GOOGLENET*

**Table 2.** Configuration details for our GoogLeNet model.

| Type | Patch Size/Stride | Depth | #1x1 | #3x3 Reduce | #3x3 | #5x5 Reduce | #5x5 | Pool Proj | Pramas | Output Size | Ops |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CONVOLUTION | 7x7/2 | 1 | | | | | | | 2.7K | 112X112X64 | 34M |
| MAXPOOL 3x3/2 | | 0 | | | | | | | | 56X56X64 | |
| CONVOLUTION | 3x3/1 | 2 | | 64 | 192 | | | | 112K | 56X56X192 | 360M |
| MAXPOOL | 3x3/2 | 0 | | | | | | | | 28X28X192 | |
| INCEPTION(3a) | | 2 | 64 | 96 | 128 | 16 | 32 | 32 | 159K | 28X28X256 | 128M |
| INCEPTION(3b) | | 2 | 128 | 128 | 192 | 32 | 96 | 64 | 380K | 28X28X480 | 304M |
| MAXPOOL | 3x3/2 | 0 | | | | | | | | 14X14X480 | |
| INCEPTION(4a) | | 2 | 192 | 96 | 208 | 16 | 48 | 64 | 364K | 14X14X512 | 73M |
| INCEPTION(4b)1 | | 2 | 160 | 112 | 224 | 24 | 64 | 64 | 437K | 14X14X512 | 88M |

**Table 2.** Configuration details for our GoogLeNet model.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INCEPTION(4c) 1 | | 2 | 128 | 128 | 256 | 24 | 64 | 64 | 463K | 14 × 14 × 512 | 100M |
| INCEPTION(4d) | | 2 | 112 | 144 | 288 | 32 | 64 | 64 | 580K | 14 × 14 × 528 | 119M |
| MAXPOOL | 3x3/2 | 0 | | | | | | | | 7 × 7 × 832 | |
| INCEPTION(5a) | | 2 | 256 | 160 | 320 | 32 | 128 | 128 | 1072K | 7 × 7 × 832 | 54M |
| INCEPTION(5b) | | 2 | 384 | 192 | 384 | 48 | 128 | 128 | 1388K | 7 × 7 × 1024 | 71M |
| AVERAGE POOL | 7 × 7/1 | 0 | | | | | | | | 1 × 1 × 1024 | |
| DROPOUT Rum | | 0 | | | | | | | | 1 × 1 × 1024 | |
| LINEAR | | 1 | | | | | | | 1000K | 1 × 1 × 1000 | 1M |
| SOFTMAX | | 0 | | | | | | | | 1 × 1 × 1000 | |

The architecture we will proceed with further is GoogLeNet, also known as Inception v1. The GoogLeNet (Inception v1) has nine linearly fitted inception modules. It has 22 layers (including the pooling layers. It will be 27). Without sacrificing accuracy or speed, the Inception net eventually lowers the computing cost to a significant degree. Therefore, GoogleNet shows seven million parameters. Nine inception modules, three SoftMax layers for the main auxiliary classifiers, four max-pooling layers, four convolutional layers, three average-pooling five fully connected layers, and four max-pooling layers make up this model. This architecture uses ReLU [30] operation in all convolutional layers and dropout regularization in the fully connected layer.

**1x1 Convolution:** The 1 × 1 convolution is used in the conception architecture's architecture. These convolutions are used to improve the depth of the architecture while reducing the number of parameters in the design.

**Global average pooling**: This is used to lower the number of trainable parameters at the network's end to zero and increase the top-1 accuracy by 0.6%.

**Inception model:** The inception model 3 × 3 max pooling and 1 × 1, 3 × 3, 5 × 5 convolution are performed similarly at the input, and the outputs are piled together to produce the final output. As shown in Figure 5. The first convolutional layer uses a patch of size 7 × 7, which is moderately massive compared to other patch sizes within the network. The main intention of this layer is to lessen the input image but not to lose the information and necessary details. The input image is decreased to a component of four at the second convolutional layer and a factor of eight before thrusting out to the first inception model. Still, the feature maps are generated in a vast amount.

GoogLeNet requires a particular type of configuration to produce accurate results, and the configuration details are mentioned in Table 2. Using nine inception modules makes it unique from other architectures, and Figure 6 delineates a straightforward inception model for Google net architecture Figure 7 Portrays the Google net.

The algorithm mentioned below is used to get the input which is the preprocessed image set, and implements all inception modules upon the image set and gives the output, which is the percentage of accuracy or 1000 as specified by us. If it is 100% or 1000, the image is accurate. Then we can proceed with the further prediction of the image. Four convolutional layers, four max-pooling layers, nine inception modules, three SoftMax layers for the primary auxiliary classifiers, three average pooling layers, and five fully connected layers make up the class GoogLeNet [31].

**Algorithm 2:** Preprocessed image set and implements all inception modules.

```
Input: preprocessed image
Output: array (according to the batch size) #1000 as the output size
Import torch and all neural network modules from the torch as nn
    begin
    Define Function googlenet():
        If the model is pretrained:
            If the transform_input is not in kwargs:
                set kwargs ← true
                    if aux_logits not in kwargs:
                        set kwargs ← false
                         if kwargs is aux_logits:
                              send a warning (model is not pretrained)
                         if model is not original_aux_logits:
                            set model.aux_logits ← False
                                       Set model. aux1 ←None
                                       Set model. aux2 ←None
                          return model
                return GoogLeNet(**kwargs)
    #Create class GoogLeNet:
    Test model with attributes and member functions
    This class involves all the inception layers, as mentioned already.
    GoogLeNetOutputs:
       if self.training and self.aux_logits:
            return _GoogLeNetOutputs(x, aux2, aux1)
       else:
             return x     # type: ignore[return-value]
```

## 5. Results and analysis

In this section, we analyze the result and compare the result with different parameters.

### *5.1. Comparison of AlexNet and GoogLeNet based on performance*

We use six evaluation matrices to observe their performance: Time cost, area under the curve, Sensitivity, Specificity, Accuracy, and Loss. The five matrices, except the time cost, are calculated using the formula below:

**Sensitivity:** The exactness of positive models is correlated with sensitivity. It alludes to how numerous positive classes were labeled effectively; this can be determined with the equation below.

$$Sensitivity = TP/(TP + FN)$$

False negatives are instances when positive cases are mistakenly classified as unfavorable, whereas real positives are instances where positive situations are appropriately identified.

**Specificity:** The conditional probability of true negatives, which may be calculated using the method below, and specificity are correlated.

$$Specificity = TN/(TN + FP)$$

Where FP is the number of false positives, which are defined as the negative instances that are mistakenly categorized as positive cases, and TN is the number of true negatives or negative cases that are negative and classed as unfavorable.

**Accuracy:** It is the number of correct assessments divided by the total assessments

$$Accuracy = (TP + TN)/\,(TP + TN + FP + FN)$$

Loss = -Σ, yj, log (ˆ$y_i$)

(y is the actual value, and ˆ$y_i$ is the predicted value)

AUC: AUC corresponds to the area under the curve

AUC = 0.5 (Sensitivity + Specificity)

*5.2. Performance of AlexNet in (%)*

1) The performance of AlexNet has been measured using the measures mentioned above and arranged in Table 3.

**Table 3.** The performance measure of AlexNet for our data set in percentage.

| Performance measures | Alex Net |
|---|---|
| Accuracy | 98.95 |
| Sensitivity | 98.4 |
| Specificity | 99.58 |
| The Area under the curve | 99.05 |
| Time | 95.01 |

2) Performance of GoogLeNet in (%):

The performance of GoogLeNet has been measured using the measures mentioned earlier and arranged in Table 4.

From the above two Tables 4 and 5, we can see the difference between the performance matrices of both AlexNet and GoogLeNet; that is, the accuracy measure of AlexNet is found to be 98.95. In contrast, the accuracy measure of GoogLeNet is found to be 99.45. Furthermore, the sensitivity of AlexNet is 98.4, and GoogLeNet is 99.75, so from these values, we can infer from the above table that the GooGleNet is highly accurate.

**Table 4.** The performance measure of GoogLeNet for our data set in percentage.

| Performance measures | Google Net |
|---|---|
| Accuracy | 99.45 |
| Sensitivity | 99.75 |
| Specificity | 99.91 |
| The area under the curve | 99.8 |
| Time | 135.40 |

**Table 5.** The performance measure of GoogLeNet for our data set in percentage.

| Model | FLOPS | # Params | fps | Latency | Accuracy |
|---|---|---|---|---|---|
| CNN | 83.811 | 32.464 | 1406.23 | 0.0465 | 76.37 |
| AlexNet | 167.685 | 32.464 | 860.84 | 0.0764 | 77.85 |
| GoogleNet | 85.685 | 60.369 | 1114.82 | 0.0564 | 76.14 |

The number of parameters that GoogLeNet consumes is significantly less; that is, the depth of AlexNet is 8, it takes 60 million parameters, and the image input size is 22 × 227. At the same time, GoogLeNet has a depth of 22 while it takes 7 million parameters, and the input size is 244 × 24.

Our model has been tested with three different learning rates, and when we compare it with the performance results of Alex Net, the GoogLeNet is proved to be more effective, and the results are shown in Tables 4 and 5.

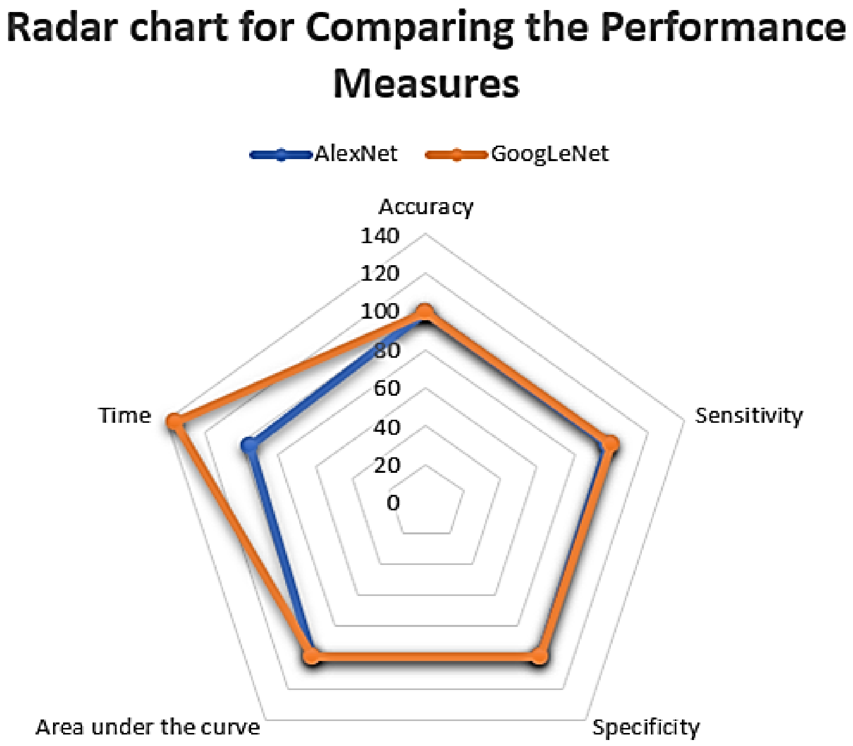

**Figure 6.** Comparative analysis of the performance of proposed work.

**Table 6.** Performance measure of Google Net after testing with three different learning rates.

| PERFORMANCE MEASURES | LR = 0.01 | | LR = 0.001 | | LR = 0.001 | |
|---|---|---|---|---|---|---|
| | TRAINING SET | TESTING SET | TRAINING SET | TESTING SET | TRAINING SET | TESTING SET |
| Accuracy | 74.01% | 77.10% | 100% | 98.10% | 100% | 100% |
| Loss | 2.81 | 2.25 | 0.021 | 0.001 | $1.0095 \times 10^{-6}$ | $9.6568 \times 10^{-8}$ |
| Specificity | 0 | 0 | 1 | 1 | 1 | 1 |
| Sensitivity | 0.71 | 0.75 | 1 | 1 | 1 | 1 |
| AUC | 0.50 | 0.50 | 1 | 0.9756 | 1 | 1 |

**Table 7.** Performance measure of Alex Net after testing with three different Learning rates.

| PERFORMANCE MEASURES | LR = 0.01 | | LR = 0.001 | | LR = 0.001 | |
|---|---|---|---|---|---|---|
| | TRAINING SET | TESTING SET | TRAINING SET | TESTING SET | TRAINING SET | TESTING SET |
| Accuracy | 74.01% | 77.10% | 100% | 100% | 100% | 100% |
| Loss | 1.09 | 3.65 | $3.67 \times 10^{-8}$ | $9.1 \times 10^{-8}$ | $3.9714 \times 10^{-8}$ | $3.1243 \times 10^{-8}$ |
| Specificity | 0 | 0 | 1 | 1 | 1 | 1 |
| Sensitivity | 0.71 | 0.75 | 1 | 1 | 1 | 1 |
| AUC | 0.5 | 0.5 | 1 | 1 | 1 | 1 |

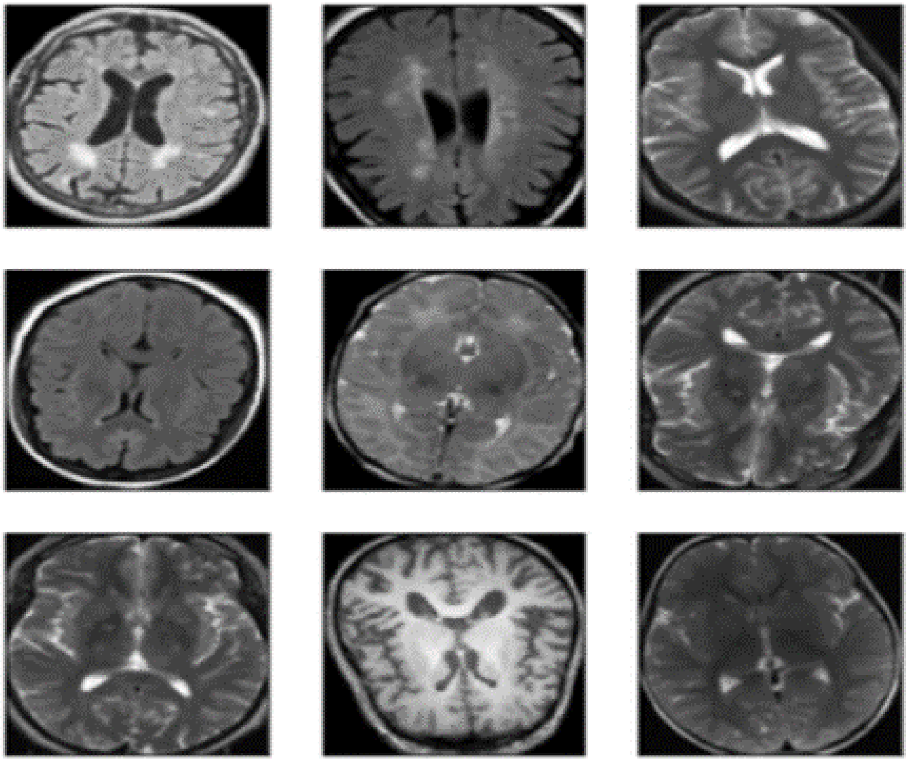

**Figure 7.** Brain tumor NO-Class dataset.

Accuracy is a metric that measures the fraction of correct predictions made by the model over the total number of predictions. It is a value between 0 and 1, where 1 indicates that the model is making perfect predictions, and 0 indicates that the model is making no correct predictions. Accuracy is an excellent metric to use when the classes are balanced and have roughly equal numbers of samples.

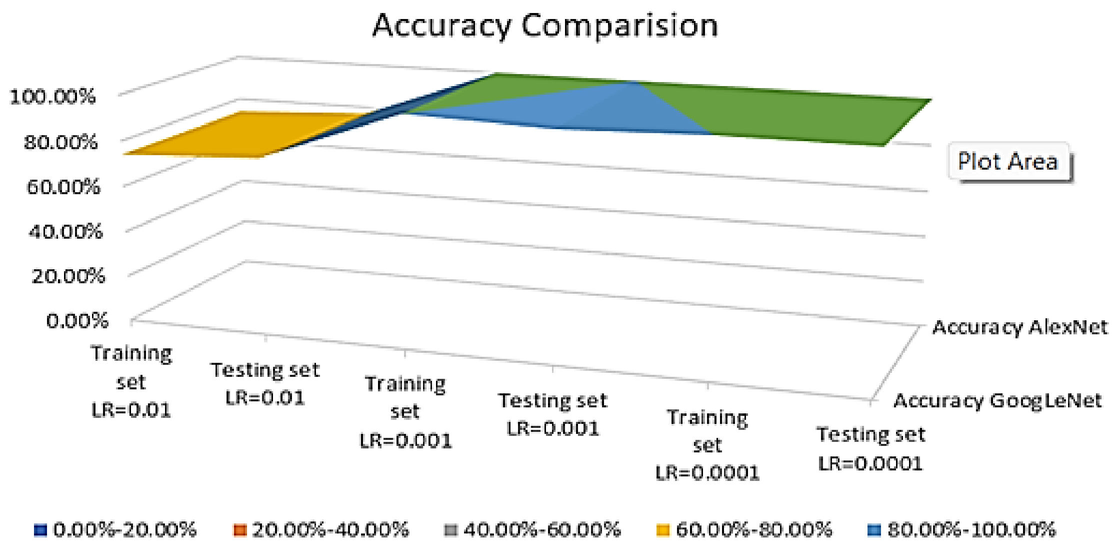

**Figure 8.** 3D-surface graph showing the accuracy comparison.

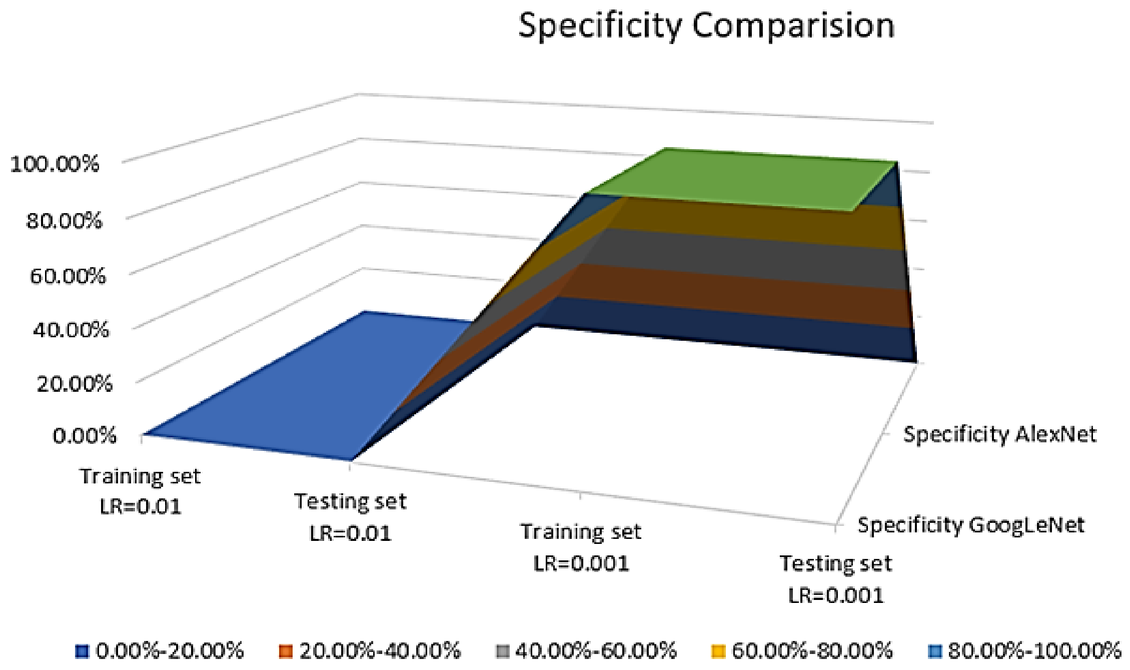

**Figure 9.** 3D-surface graph showing the specificity comparison.

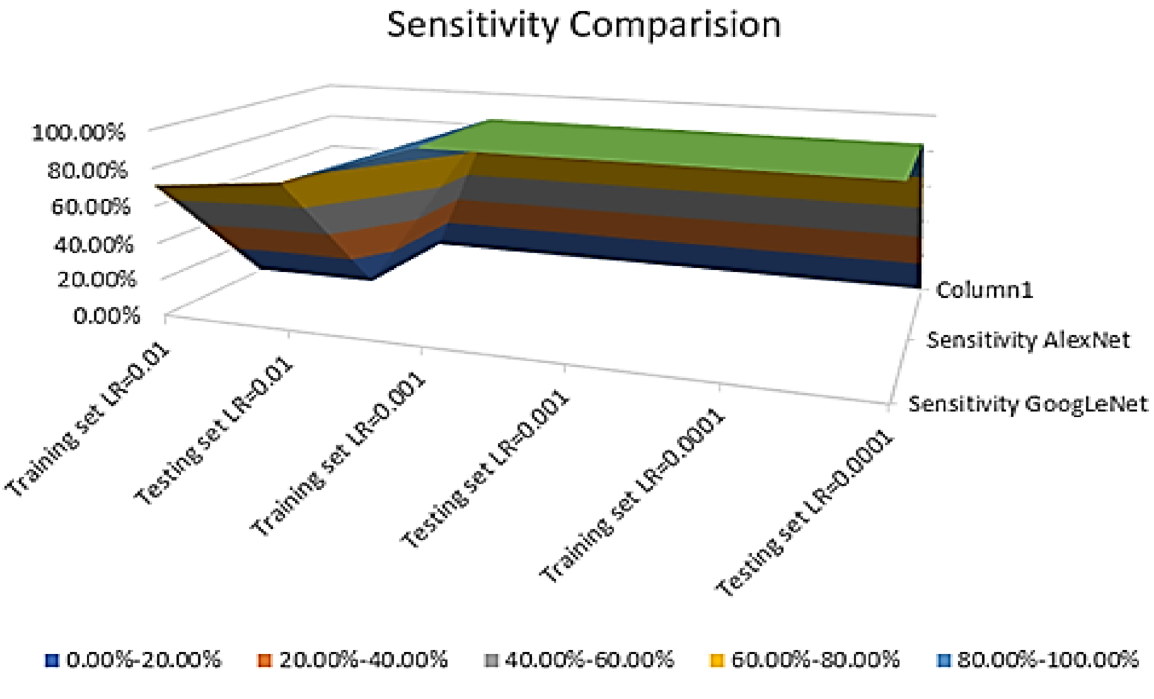

**Figure 10.** 3D-surface graph showing the sensitivity comparison.

The above-given Figures 8–10 are the graphs depicting the accuracy comparison, specificity comparison, and sensitivity comparison, respectively.

Both loss and accuracy must be relative to the problem being solved and the data being used. A model with a low loss and high accuracy on the training data does not necessarily mean that the model will perform well on new, unseen data. This is why it is common to split the data into training and validation sets, and to monitor both loss and accuracy on the validation set as the model is trained.

**Table 8.** Performance evaluation of Exiting model with proposed work.

| Reference | Dataset | Approach | Accuracy (%) |
|---|---|---|---|
| W. Ayadi et al. (2021) [5] | publicly available database | CNN | 92.98 |
| R. L. Kumar et.al. (2021) [15] | Figshare | GoogLeNet, AlexNet and VGGNet | 98.69 |
| F. Abdolkarimzadeh et.al. (2021) [1] | Harvard Medical School | DL, K-NN, LDA | 95.45–96.97 |
| S. N. Shivhare et.al. (2021) [27] | Figshare, Brainweb, Radiopaedia | LeNet | 88 |
| V. V. S. Sasank et.al. (2021) [26] | Kaggle | CNN | 96 |

**Table 8.** Performance evaluation of Exiting model with proposed work.

| | | | |
|---|---|---|---|
| O. Polat and C. Gungen (2021) [20] | figshare | Fuzzy C Mean+ CNN | 97.5 |
| X. L. Lei, (2021) [18] | Nanfang Hospital and General Hospital, Tianjin Medical University | CapsNet | 86.5 |
| V. V. S. Sasank and S. Venkateswarlu (2021) [25] | Brats-2015 | CNN | 92.13 |
| P. Wang and A. C. S. Chung (2021) [29] | Kaggle | CNN | 92.00 |
| M. M. Badža and M. Č. Barjaktarović (2022) [38] | Radhamadhab Dalai, IEEE, FIGSHARE | convolutional neural network (CNN) | 97.28% |
| Y. Guan et al. (2021) [36] | Radhamadhab Dalai, IEEE, FIGSHARE | Capsule Networks | 86.56% |
| M. Aamir et al. (2022) [37] | Radhamadhab Dalai, IEEE, FIGSHARE | Capsulenet +SVM | 92.60 |
| M. M. Badža and M. Č. Barjaktarović (2020) [38] | Radhamadhab Dalai, IEEE, FIGSHARE | GLCM and Wavelet Packets. | 93.30 |
| N. Zheng et al. (2023)[39] | Radhamadhab Dalai, IEEE, FIGSHARE | Arithmetic Optimization Algorithm | 96.8 |
| Q. Zhou (2023) [40] | Radhamadhab Dalai, IEEE, FIGSHARE | light-weight convolutional neural network CNN) with SCM-GL | 95.8 |
| S. Deepak and P. M. Ameer (2023) [41] | Radhamadhab Dalai, IEEE, FIGSHARE | KNN+SVM | 98.2 |
| G. Xiao et al. (2023) [42] | Radhamadhab Dalai, IEEE, FIGSHARE | jigsaw puzzle | 97.4 |
| Proposed Work | Radhamadhab Dalai, IEEE, FIGSHARE | CNN, AlexNet & GoogLeNet | 99.45 |

Table 8 shows that the proposed work outperformed in comparison to the existing method and active more accuracy. The runtime cost for the learning process of Alex Net for the learning rate (LR) 0.0001 is 19:01 minutes, for LR 0.001 is 15 minutes, and for LR 0.01 is 8:95 minutes, while in the case of Google Net for LR 0.0001, the runtime cost is 34:45 minutes for LR 0.001 it is 35:25 minutes, and for LR 0.01 it is 33:21 minutes. Hence, we can conclude that our model is best for this disease.

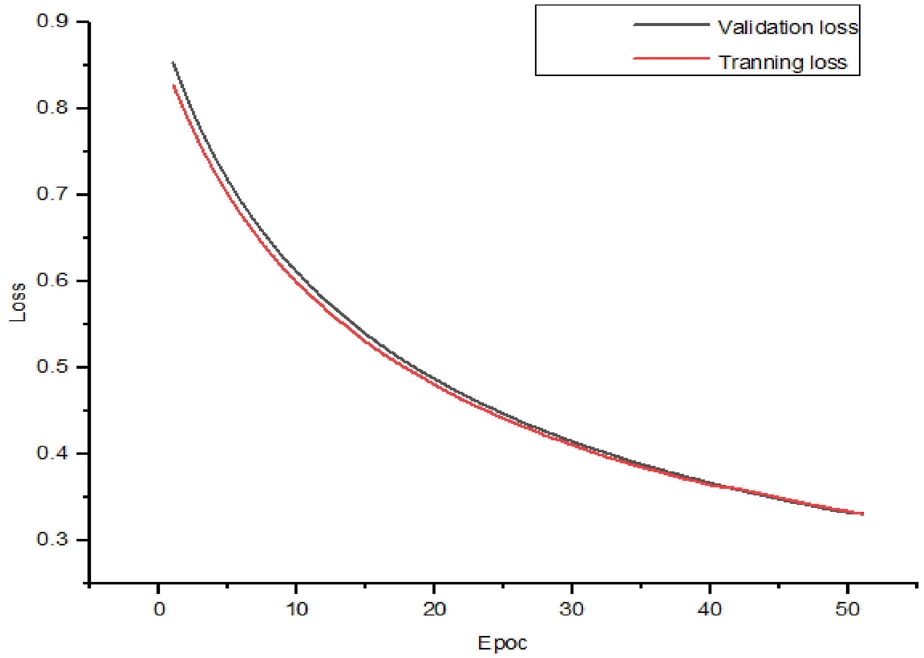

**Figure 11.** Training and validation performance of loss.

Figure 11 shows the comparative analysis of loss during training and validation of the proposed model.

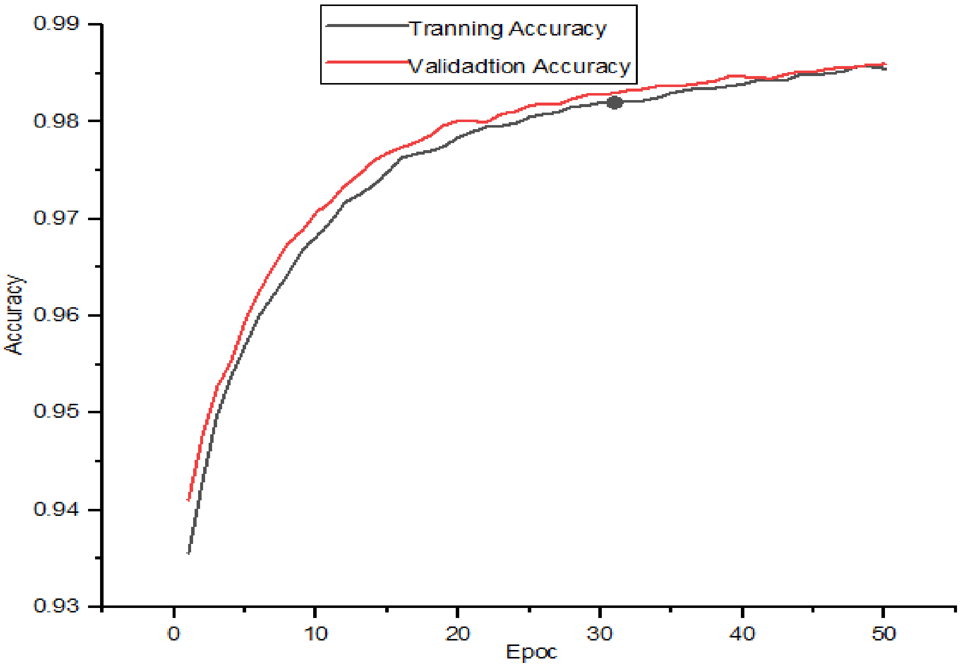

**Figure 12.** Training and validation performance of accuracy.

Figure 12 shows the comparative analysis of accuracy during training and validation of the proposed model.

A total of 70% of the dataset is used for training, 20% for validation, and 10% for testing. Python is used for developing the existing system. Figure 9 displays how the proposed approach improves validation accuracy for segmentation to 99.45% and validation loss to 0.01%. it suggested that the proposed model outperforms the state-of-the-art methods in classifying and segmenting images of brain tumors.

## 6. Conclusions

This study presented a new technique for detecting tumors at an early stage. We used an image edge identification strategy to locate the high-value region in MRI images. We then used the information expansion strategy to expand our preliminary data. We propose a direct CNN organization as a practical tumor clustering method as a second step. However, our experimental results show we can achieve full precision even on a minimal, low dataset. Our exactness rate is exceptionally high compared to the VGG-16, resnet-50, and Inception-v3 models. This is due to the neural organization requiring a greater-than-usual amount of information to mentor on for a la mode and precise results. Because of the initial investment, our model has less stringent computational requirements. The precision of our model is also significantly higher than that of VGG-16, ResNet-50, and Inception-v3. Our proposed framework could improve the prognosis of patients with brain tumors. Widespread hyper-boundary tuning and an improved pre-landing procedure are frequently considered potential methods for improving lift model proficiency. However, our proposed framework is for paired-order problems. In the future, proposed strategies can be used for apparent characterization problems, such as recognizable proof of tumor types like Glioma, meningioma, and pituitary or is also acclimated and distinguishes other mind anomalies. Cancers such as carcinoma and carcinoma are becoming increasingly common worldwide, and our proposed framework can aid in the early detection of these dangerous infections in other clinical areas related to clinical imaging. We will apply this methodology to other logical domains in addition to the current debate over the availability of massive data, or we can use alternative exchange learning strategies with the same proposed approach.

## Conflict of interest

The authors declare there is no conflict of interest.

## References

1. F. Abdolkarimzadeh, M. R. Ashory, A. Ghasemi-Ghalebahman, A. Karimi, Inverse dynamic finite element-optimization modeling of the brain tumor mass-effect using a variable pressure boundary, *Comput. Methods Programs Biomed.*, **212** (2021), 106476, https://doi.org/10.1016/j.cmpb.2021.106476
2. R. R. Agravat, M. S. Raval, A survey and analysis on automated glioma brain tumor segmentation and overall patient survival prediction, *Arch. Comput. Methods Eng.*, **28** (2021), 4117–4152. https://doi.org/10.1007/s11831-021-09559-w

3. A. M. Alhassan, W. Zainon, Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network, *Neural Comput. Appl.*, **33** (2021), 9075–9087. https://doi.org/10.1007/s00521-020-05671-3
4. M. Alshayeji, J. Al-Buloushi, A. Ashkanani, S. Abed, Enhanced brain tumor classification using an optimized multi-layered convolutional neural network architecture, *Multimedia Tools Appl.*, **80** (2021), 28897–28917. https://doi.org/10.1007/s11042-021-10927-8
5. W. Ayadi, W. Elhamzi, I. Charfi, M. Atri, Deep CNN for brain tumor classification, *Neural Process. Lett.*, **53** (2021), 671–700. https://doi.org/10.1007/s11063-020-10398-2
6. F. Bashir-Gonbadi, H. Khotanlou, Brain tumor classification using deep convolutional autoencoder-based neural network: Multi-task approach, *Multimed. Tools Appl.*, **80** (2021), 19909–19929. https://doi.org/10.1007/s11042-021-10637-1
7. B. S. Chen, L. L. Zhang, H. Y. Chen, K. W. Liang, X. Z. Chen, A novel extended Kalman filter with support vector machine based method for the automatic diagnosis and segmentation of brain tumors, *Comput. Methods Prog. Biomed.*, **200** (2021), 105797. https://doi.org/10.1016/j.cmpb.2020.105797
8. N. V. Dhole, V. V. Dixit, Review of brain tumor detection from MRI images with hybrid approaches, *Multimed. Tools Appl.*, **81** (2022), 10189–10220. https://doi.org/10.1007/s11042-022-12162-1
9. B. V. Isunuri, J. Kakarla, Three-class brain tumor classification from magnetic resonance images using separable convolution based neural network, *Concurr. Comput. Prac. Experience*, **34** (2022), e6541–e6549. https://doi.org/10.1002/cpe.6541
10. T. A. Jemimma, Y. J. V. Raj, Significant LOOP with clustering approach and optimization enabled deep learning classifier for the brain tumor segmentation and classification, *Multimed. Tools Appl.*, **81** (2022), 2365–2391. https://doi.org/10.1007/s11042-021-11591-8
11. B. Jena, G. K. Nayak, S. Saxena, An empirical study of different machine learning techniques for brain tumor classification and subsequent segmentation using hybrid texture feature, *Mach. Vision Appl.*, **33** (2022), 1–16. https://doi.org/10.1007/s00138-021-01262-x
12. M. Jiang, F. H. Zhai, J. Kong, A novel deep learning model DDU-net using edge features to enhance brain tumor segmentation on MR images, *Artif. Intell. Med.*, **121** (2021), 102180. https://doi.org/10.1016/j.artmed.2021.102180
13. S. Kadry, V. Rajinikanth, N. S. M. Raja, D. J. Hemanth, N. M. S. Hannon, A. N. J. Raj, Evaluation of brain tumor using brain MRI with modified-moth-flame algorithm and Kapu's thresholding: A study, *Evol. Intell.*, **14** (2021), 1053–1063. https://doi.org/10.1007/s12065-020-00539-w
14. S. Kokkalla, J. Kakarla, I. B. Venkateswarlu, M. Singh, Three-class brain tumor classification using deep dense inception residual network, *Soft Comput.*, **25** (2021), 8721–8729. https://doi.org/10.1007/s00500-021-05748-8
15. R. L. Kumar, J. Kakarla, B. V. Isunuri, M. Singh, Multiclass brain tumor classification using residual network and global average pooling, *Multimed. Tools Appl.*, **80** (2021), 13429–13438. https://doi.org/10.1007/s11042-020-10335-4
16. Y. Guan, M. Aamir, Z. Rahman, A. Ali, W. A. Abro, Z. A. Dayo, et al., A framework for efficient brain tumor classification using MRI images, *Math. Biosci. Eng.*, **18** (2021), 5790–5815. https://doi.org/10.3934/mbe.2021292

17. M. Aamir, Z. Rahman, Z. A. Dayo, W. A. Abro, M. I. Uddin, I. Khan, et al., A deep learning approach for brain tumour classification using MRI images, *Comput. Electr. Eng.*, **101** (2022), 108105. https://doi.org/10.1016/j.compeleceng.2022.108105
18. X. L. Lei, X. S. Yu, J. N. Chi, Y. Wang, J. S. Zhang, C. D. Wu, Brain tumor segmentation in MR images using a sparse constrained level set algorithm, *Expert Syst. Appl.*, **168** (2021), 114262. https://doi.org/10.1016/j.eswa.2020.114262
19. G. L. Li, J. H. Sun, Y. L. Song, J. F. Qu, Z. Y. Zhu, M. R. Khosravi, Real-time classification of brain tumors in MRI images with a convolutional operator-based hidden Markov model, *J. Real Time Image Process.*, **18** (2021), 1207–1219. https://doi.org/10.1007/s11554-021-01072-4
20. O. Polat, C. Gungen, Classification of brain tumors from MR images using deep transfer learning, *J. Supercomput.*, **77** (2021), 7236–7252. https://doi.org/10.1007/s11227-020-03572-9
21. S. Preethi, P. Aishwarya, An efficient wavelet-based image fusion for brain tumor detection and segmentation over PET and MRI image, *Multimed. Tools Appl.*, **80** (2021), 14789–14806. https://doi.org/10.1007/s11042-021-10538-3
22. S. Ramesh, S. Sasikala, N. Paramanandham, Segmentation and classification of brain tumors using modified median noise filter and deep learning approaches, *Multimed. Tools Appl.*, **80** (2021), 11789–11813. https://doi.org/10.1007/s11042-020-10351-4
23. C. S. Rao, K. Karunakara, A comprehensive review on brain tumor segmentation and classification of MRI images, *Multimed. Tools Appl.*, **80** (2021), 17611–17643. https://doi.org/10.1007/s11042-020-10443-1
24. A. S. Reddy, P. C. Reddy, MRI brain tumor segmentation and prediction using modified region growing and adaptive SVM, *Soft Comput.*, **25** (2021), 4135–4148. https://doi.org/10.1007/s00500-020-05493-4
25. V. V. S. Sasank, S. Venkateswarlu, Brain tumor classification using modified kernel based softplus extreme learning machine, *Multimed. Tools Appl.*, **80** (2021), 3513–13534. https://doi.org/10.1007/s11042-020-10423-5
26. V. V. S. Sasank, S. Venkateswarlu, Hybrid deep neural network with adaptive rain optimizer algorithm for multi-grade brain tumor classification of MRI images, *Multimed. Tools Appl.*, **81** (2022), 8021–8057. https://doi.org/10.1007/s11042-022-12106-9
27. S. N. Shivhare, N. Kumar, Tumor bagging: A novel framework for brain tumor segmentation using metaheuristic optimization algorithms, *Multimed. Tools Appl.*, **80** (2021), 26969–26995. https://doi.org/10.1007/s11042-021-10969-y
28. J. J. Wang, J. Gao, J. Ren, Z. Luan, Z. Yu, Y. Zhao, et al., DFP-ResUNet: Convolutional neural network with a dilated convolutional feature pyramid for multimodal brain tumor segmentation, *Comput. Methods Prog. Biomed.*, **208** (2021), 106208. https://doi.org/10.1016/j.cmpb.2021.106208
29. P. Wang, A. C. S. Chung, Relax and focus on brain tumor segmentation, *Medical Image Anal.*, **75** (2022), 102259. https://doi.org/10.1016/j.media.2021.102259
30. Y. Wang, J. L. Peng, Z. D. Jia, Brain tumor segmentation via C-dense convolutional neural network, *Prog. Artif. Intell.*, **10** (2021), 147–156. https://doi.org/10.1007/s13748-021-00232-8
31. Y. L. Wang, Z. J. Zhao, S. Y. Hu, F. L. Chang, CLCU-Net: Cross-level connecte d U-shape d network with selective feature aggregation attention module for brain tumor segmentation, *Comput. Methods Prog Biomed.*, **207** (2021), 106154. https://doi.org/10.1016/j.cmpb.2021.106154

32. X. H. Wu, L. Bi, M. Fulham, D. D. Feng, L. P. Zhou, and J. Kim, Unsupervised brain tumor segmentation using a symmetric-driven adversarial network, *Neurocomputing*, **455** (2021), 242–254. https://doi.org/10.1016/j.neucom.2021.05.073
33. D. W. Zhang, G. H. Huang, Q. Zhang, J. G. Han, J. W. Han, Y. Z. Yu, Cross-modality deep feature learning for brain tumor segmentation, *Pattern Recogn.*, **110** (2021), 107562. https://doi.org/10.1016/j.patcog.2020.107562
34. H. ZainEldin, S. A. Gamel, E. M. El-Kenawy, A. H. Alharbi, D. S. Khafaga, A. Ibrahim, et al., Brain tumor detection and classification using deep learning and sine-cosine fitness grey wolf optimization, *Bioengineering*, 1 (2023), 10–18. https://doi.org/10.3390/bioengineering10010018
35. V. Kushwaha, P. Maidamwar, BTFCNN: Design of a brain tumor classification model using fused convolutional neural networks, in *2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-22)*, (2022), 1–6. https://doi.org/10.1109/ICETET-SIP-2254415.2022.9791734
36. Y. Guan, M. Aamir, Z. Rahman, A. Ali, W. A. Abro, Z. A. Dayo, et al., A framework for efficient brain tumor classification using MRI images, *Math. Biosci. Eng.*, **18** (2021), 5790–5815. https://doi.org/10.3934/mbe.2021292
37. M. Aamir, Z. Rahman, Z. A. Dayo, W. A. Abro, M. I. Uddin, I. Khan, et al., A deep learning approach for brain tumour classification using MRI images, *Comput. Electr. Eng.*, **1** (2022), 108105–108120. https://doi.org/10.1016/j.compeleceng.2022.108105
38. M. M. Badža, M. Č. Barjaktarović, Classification of brain tumors from MRI images using a convolutional neural network., *Appl. Sci. (Basel)*, **10** (2020), 1999–2020. https://doi.org/10.3390/app10061999
39. N. Zheng, G. Zhang, Y. Zhang, F. R. Sheykhahmad, Brain tumor diagnosis based on Zernike moments and support vector machine optimized by chaotic arithmetic optimization algorithm, *Biomed. Signal Process. Control.*, **82** (2023), 104543, 104543–104553. https://doi.org/10.1016/j.bspc.2022.104543
40. Q. Zhou, Medical image classification using light-weight CNN with spiking cortical model based attention module, *IEEE J. Biomed. Health Inform.*, **1** (2023), 1–13. https://doi.org/10.1109/JBHI.2023.3241439
41.S. Deepak, P. M. Ameer, Brain tumor categorization from imbalanced MRI dataset using weighted loss and deep feature fusion, *Neurocomputing*, **520** (2023), 94–102. https://doi.org/10.1016/j.neucom.2022.11.039
42. G. Xiao, H. Wang, J. Shen, Z. Chen, Z. Zhang, X. Ge, Contrastive learning with dynamic weighting and jigsaw augmentation for brain tumor classification in mrI, *Neural Process Lett.*, **1** (2023) ,1–29. https://doi.org/10.1007/s11063-022-11108-w

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)