Medical & Biological Engineering & Computing (2023) 61:1821–1836
https://doi.org/10.1007/s11517-023-02820-3

ORIGINAL ARTICLE

# Automatic brain tumor detection using CNN transfer learning approach

Vinayak K. Bairagi[1] · Pratima Purushottam Gumaste[2] · Seema H. Rajput[3] · Chethan K. S.[4]

Received: 16 October 2021 / Accepted: 27 February 2023 / Published online: 23 March 2023
© International Federation for Medical and Biological Engineering 2023

**Abstract**
Automatic brain tumor detection is a challenging task as tumors vary in their position, mass, nature, and similarities found between brain lesions and normal tissues. The tumor detection is vital and urgent as it is related to the lifespan of the affected person. Medical experts commonly utilize advanced imaging practices such as magnetic resonance imaging (MRI), computed tomography (CT), and ultrasound images to decide the presence of abnormal tissues. It is a very time-consuming task to extract the tumor information from the enormous quantity of information produced by MRI volumetric data examination using a manual approach. In manual tumor detection, precise identification of tumor along with its details is a complex task. Henceforth, reliable and automatic detection systems are vital. In this paper, convolutional neural network based automated brain tumor recognition approach is proposed to analyze the MRI images and classify them into tumorous and non-tumorous classes. Various convolutional neutral network architectures like Alexnet, VGG-16, GooGLeNet, and RNN are explored and compared together. The paper focuses on the tuning of the hyperparameters for the two architectures namely Alexnet and VGG-16. Exploratory results on BRATS 2013, BRATS 2015, and OPEN I dataset with 621 images confirmed that the accuracy of 98.67% is achieved using CNN Alexnet for automatic detection of brain tumors while testing on 125 images.

**Keywords** Neural networks · Brain tumor · MRI · Transfer learning · Alexnet architecture · VGG-16 architecture

## 1 Introduction

Cancer may be defined as the unrestricted, abnormal, and unnatural growth of the section of the cells or tissue. Occurrence of this abnormal cells raise in the brain is called as brain tumor. Brain tumors are considered fatal cancers. The tumors when originated in brain itself are classified as primary. Gliomas are brain tumors that arise from brain glial cells. Early diagnosis of brain tumor is vital in enhancing treatment opportunities. Medical imaging techniques such as computed tomography (CT), single photon emission computed tomography (SPECT), positron emission tomography (PET), magnetic resonance spectroscopy (MRS), and magnetic resonance imaging (MRI) are altogether utilized to get the important information about outline, length, position, and metabolism of brain tumors [1].

The various research studies used brain MRI imaging because of their high resolution. After capturing the brain MRI, it is essential to separate the tumor region from the MRI brain image. Accurate segmentation of brain MRI images helps the medical practitioners for planning the treatment of the patients. Due to complicated brain tissue structure, manual tumor segmentation from MRI images is hard and complex and it is primarily based on the operator's experience and subjective selection. Therefore, computerized segmentation strategies are required. Because of its high variability in brain tumor's shape, size, regularity, area, and heterogeneous presentation, there are several difficulties in automated algorithms [2].

Artificial neural network (ANN) consists of machine learning (ML) and deep learning (DL). It plays an important role for classification of biomedical images. ANN is

✉ Vinayak K. Bairagi
bairagi1@gmail.com

1 Department of Electronics and Telecommunication, AISSMS Institute of Information Technology, Pune, India

2 Department of Electronics and Telecommunication, JSPM's Jayawantrao Sawant College of Engineering, Pune, India

3 Department of Electronics and Telecommunications, Cummins College of Engineering for Women, Savitribai Phule Pune University, Pune, India

4 RV Institute of Technology and Management, Bangalore, India

consisting of layers namely, Input, Hidden, and Output. The inputs can be radiometric functions that have been extracted from the images. Automated image segmentation, facts analysis, and image reconstruction play a vital role in ML [3, 4]. Figure 1a and b show the samples of brain MRI normal and abnormal images.

In this paper, an algorithm is proposed based on convolutional neural network (CNN) to detect the tumor in brain MRI images. CNNs are an improvement on the general idea of artificial neural networks. Their ability to automatically learn appropriate representations of the data makes problems easier to solve, especially problems involving large amounts of data that would otherwise require a lot of pre-processing.

The organization of the remaining paper is as follows. Section 2 gives an update of previous research work carried out by various researchers, with respect to brain tumor detection, whereas in Section 3 proposed technique is explained in detail along with methodology. The obtained results, findings, and a discussion of the proposed method are explained in Section 4. Section 5 comprises of concluding remarks as well as upcoming opportunities to work in the presented work.

## 2 Literature review

In deep learning, the machine learns useful knowledge and features from raw information [5, 6], bypassing physical and troublesome steps. CNN is an effective method of analyzing good descriptions of images [7, 8]. Various architectures used in CNN are Alexnet [1], ZFNet [10], VGG Net [11], Google Net [12], and ResNet [13]. The AlexNet [9] is better among all the existing algorithms. Table 1 shows the information of various available community architectures.

A typical CNN architecture consists of multiple layers such as convolution, pooling, activation, and classification (fully connected) layers [14]. Convolutional layer produces feature maps by convolving a kernel across the input image to generate the image features [15]. Pooling layer is used to down-sample the output of preceding convolutional layers by using the maximum or average of the defined neighborhood as the value passed to the next layer. Rectified Linear Unit (ReLU) is the most commonly used activation functions [16]. The convolution operation can create uncommon element maps depending on the channels utilized. The pooling layer plays a down sampling activity. Neurons in a fully connected layer are associated with all actuations inside the first layer. The architecture of Alexnet is shown in Table 2.

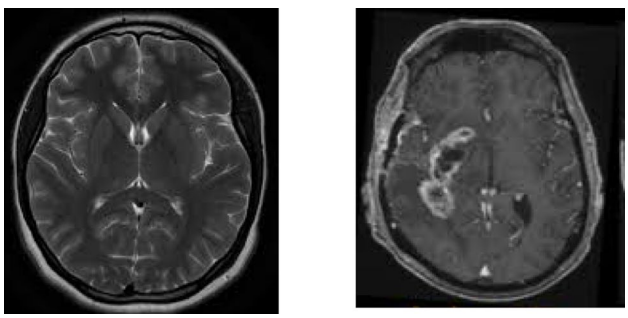


**Fig. 1** Brain MRI images. **a** Normal brain MRI. **b** Brain MRI with tumor

## 3 Methodologies

### 3.1 Brain MRI dataset

Dataset collection is the most important step in any research work. BRATS 2013 and BRATS 2015 are the main dataset used in this study along with Open-I NLM dataset [18]. The dataset consists of both the types of images, i.e., images with tumors and images without tumors. MRI images are collected from MRI machines with different field strength. However, images captured below 1.5 T field intensity are included in this work. This field strength is sufficient to envision the tumors in the images. Large number of dataset is available at Open-I website of dataset portal. Table 3 shows some major depositories used by the researchers to conduct the investigation.

Figure 2 shows splitting of the dataset during the experimentation. MRI images have different modalities like-$T_1$, $T_2$, and FLAIR. In the proposed work, datasets are divided into two parts, 80% for training and validation along with 20% to prepare to test dataset. The splitting of the overall images is as shown in Fig. 2 [19, 20].

The sample images from the dataset are shown in Fig. 3, with three different views.

**Table 1** Brief tabulation of DL network architectures and object Recognition Challenge [8]

| Architecture | Article | Top-5 error rate | Number of parameters |
|---|---|---|---|
| Alexnet [1] | Krizhevsky et al., 2017 | 16.4% | 60 million |
| ZFNet [10] | Zeiler et al., 2013 | 11.7% | N/A |
| VGG Net [11] | Simonyan et al., 2014 | 7.3% | 138 million |
| Google Net [12] | Szegedy et al., 2015 | 6.7% | 5 million (V1) & 23 million (V2) |
| ResNet [13] | He. Kaiming et al., 2016 | 3.57% | 25.6 million (ResNet-50) |

**Table 2** Alexnet architecture [using basic mode transfer learning] [17]

| Layer | Name | Details |
|---|---|---|
| 1 | Input | 227×227×3 |
| 2 | Conv1 | 96 kernels of size 11×11 applied with a stride of [4 4] and padding of [0 0] |
| 3 | Relu1 | ReLU |
| 4 | Norm1 | Cross channel normalization with 5 channels per element |
| 5 | Pool1 | Pooling size of 3×3 and stride [2 2] padding [0 0] |
| 6 | Conv2 | 256 5×5×48 convolutions with stride [1 1] and padding [2 2] |
| 7 | Relu2 | ReLU |
| 8 | Norm2 | Cross-channel normalization with 5 channels per element |
| 9 | Pool2 | 3×3 max pooling with stride [2 2] and padding [0 0] |
| 10 | Conv3 | 384 3×3x256 convolutions with stride [1 1] and padding [1 1] |
| 11 | Relu3 | ReLU |
| 12 | Conv4 | 384 3×3×192 convolutions with stride [1 1] and padding [1 1] |
| 13 | Relu4 | ReLU |
| 14 | Conv5 | 256 3×3×192 convolutions with stride [1 1] and padding [1 1] |
| 15 | Relu5 | ReLU |
| 16 | Pool5 | 3×3 max pooling with stride [2 2] and padding [0 0] |
| 17 | Fc6 | 4096 fully connected layer |
| 18 | Relu6 | ReLU |
| 19 | Drop6 | 50% dropout |

**Table 3** Summary of various datasets used for brain tumor detection [8]

| Dataset | Train | Test | Image size |
|---|---|---|---|
| Brats2013 | 55 | 10 | 160×216×176 |
| Brats2015 | 274 | 53 | 240×240×155 |
| Open-I | 167 | 62 | Variable size |

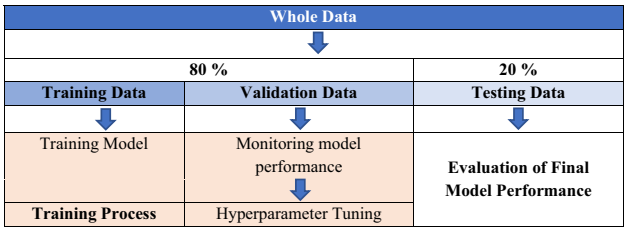


**Fig. 2** Dataset splitting for the experimentation

## 3.2 Proposed method

### 3.2.1 Transfer learning

Knowledge gained while solving one type of problem can be used to solve other similar type of problems. So, previously gained knowledge in the form of pretrained network can be used to learn and solve new similar problems. While solving the first problem, the pretrained network has learned a rich set of features; such learning can be readily used for solving other similar problems [21]. For example, one can take a network which is already trained on millions of images and retrain it for new object classification using only hundreds of images. Such retraining process will be faster and easier than starting to train the network from scratch initial stage. Fine-tuning of pretrained network with transfer learning is the important stage while using pretrain network for new applications.

In CNN, within the convolutional layer, the input image is split into several tiny regions. The output layer is used to produce the class probability. CNN brain tumor classification is split into two stages, namely training and testing. Dataset images are broken into special groups using tumor and non-tumor brain images. Within the training phase, pre-processing, features extraction, and categorization with loss feature are executed to make a prediction. In the pre-processing phase, resizing operation is performed to change the size of the image. The general framework of the brain tumor classification using CNN is shown in Fig. 4.

Brain MRI images are taken from "Open-i Biomedical" image dataset, "BRATs 2013" and "BRATs 2015" dataset. Alexnet is one of the pre-trained convolutional neural networks. A pre-trained model for brain tumor classification is used. Transfer learning is used for fast training process. With respect to the task of classification, first and last three layers of pre-trained networks are modified in order to adapt them. In fully connected layer, the output size represents absence or presence of tumor.

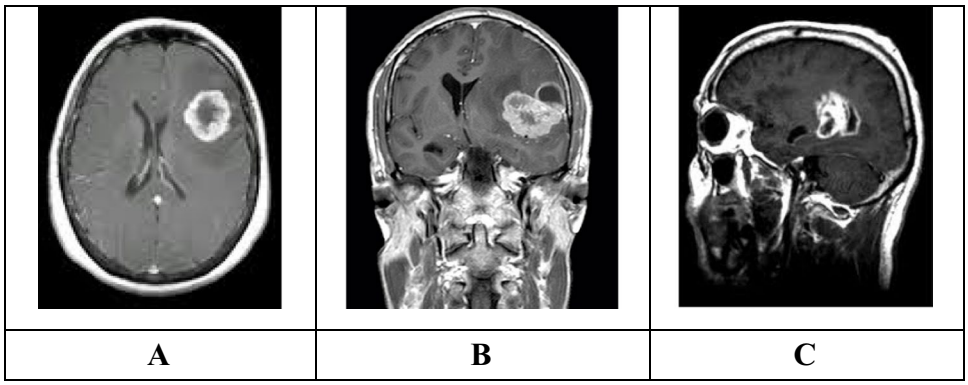


**Fig. 3** MRI image slices showing a patient's brain tumor. **A** Axial view. **B** Coronal view. **C** Sagittal section view

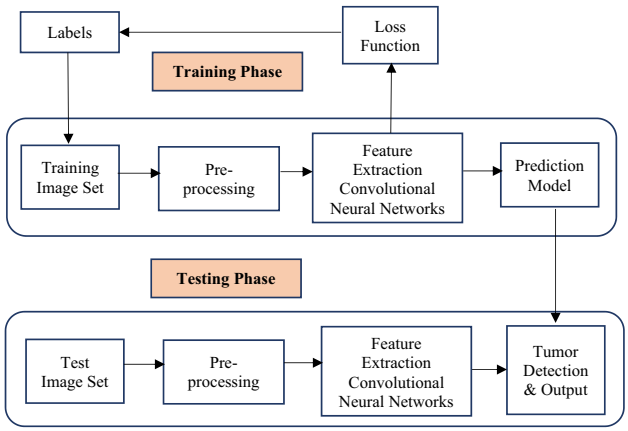


**Fig. 4** Flow of proposed technique

**Algorithm for proposed work:**
**Evaluation Process of CNN model**

1. Load image dataset ();
2. Pre-process image dataset ();
3. Data Augmentation ();
4. Split Data ();
5. Load Model ();
6. **for** *each epoch* in *epoch Number* **do**
7. **for** *each epoch* in *epoch Number* **do**
8. $\hat{y} = model(features)$;
9. $loss = crossentropy(y, \hat{y})$;
10. $optimize\ (loss)$;
11. $accuracy()$;
12. $confmatrix(\ y, \hat{y}\ )$;
13. Return ()

**Fig. 5** Flow of proposed techniques

The loss characteristic is obtained through gradient [slope] descent [Succession] algorithm. The unknown image pixel is mapped to a particular class with its rankings by means of a score characteristic. The usefulness of a selected bunch of constraints is recorded by means of the loss function. The loss characteristic count is extremely important to progress the precision. The algorithm of the proposed work is shown in Fig. 5, which explains the CNN work flow and the steps required for training and calculating the accuracy.

A significant set of 32 intensity and grain textual features are extracted from the segmented region of interest (SROI) of tumor part. These features are First order Statistical Features, Gray Level Co-occurrence Matrix (GLCM), Grey Level Run Length Encoding Matrix (GLRLM), Grey Level Gap Length Matrix (GLGLM), and Grey Level Size Zone Matrix (GLSZM).

## 4 Outcomes and discussion

The tumor detection from imaging is challenging task. In this work, special strategy is used for the tumor detection using the CNN approach. MATLAB 2020 evaluation version is used for simulation and system hardware used is i5-8250U processor, RAM: 8 GB, System type: 64-bit Operating System. Experiments are performed on BRATS 2013 [22], BRATS 2015 [23], and OPEN-I [18] dataset. Table 4 shows how the total image collection available for the experimentation is divided into training and testing portions. Few subjects contain more measurement for some patients and only one measurement per patient is used.

Image data augmentation is used to increase the dataset. Random combination of resizing, cropping, rotation, reflection, shear, and translation transformations are done to increase the dataset. Experimentations are done on actual dataset as well as augmented dataset.

The Alexnet architecture from CNN is used in the proposed work. The transfer learning approach is adopted to minimize the execution time and computational complexity. The different parameters tuned and finalized for the modelling are listed in Table 5.

The sample output obtained during the training phase is shown in Fig. 6. The blue line indicates the smoothen training curve of the CNN network during training phase, faint blue line indicates training curve, and black line indicates validation phase.

Similarly, layer-wise different features are extracted after training of Alexnet architecture, which is shown in Fig. 7. The CNN is next explored through the visual investigation of their transitional layers.

### 4.1 Features on convolutional layer 1

This layer is the second layer in the network and is named 'conv1'. These images mostly contain edges and colors, which indicates that the filters at layer 'conv1' are edge detectors and color filters. The edge detectors are at different angles, which allows the network to construct more complex features in the later layers.

### 4.2 Features on convolutional layer 2

These features are created using the features from layer 'conv2'. The second convolutional layer is named 'conv2', which corresponds to layer 6. Visualization of the first 30 features is learned by this convolutional layer 2, by setting channels to be the vector of indices 1:30. Figure 7 shows the visualization of features of first five features.

Same is the case for the features from layer 'conv3', 'conv4', and 'conv5'.

**Table 4** Dataset splitting

| | Training | | Testing | |
|---|---|---|---|---|
| | Images without Tumor (Normal) | Images with Tumor (Abnormal) | Images without Tumor (Normal) | Images with Tumor (Abnormal) |
| Actual dataset | 273 | 223 | 62 | 63 |
| Augmented data (X40) | 10,920 | 8920 | 2480 | 2520 |

**Table 5** Hyperparameters set during the training phase

| Hyperparameter | Value Set | Hyperparameter | Value Set |
|---|---|---|---|
| Gradient Decay Factor | 0.9 | GradientThreshold | Inf |
| Squared Gradient Decay Factor | 0.99 | MaxEpochs | 100 |
| Epsilon | 1.00E-08 | MiniBatchSize | 64 |
| Initial Learn Rate | 3.00E-05 | ValidationData | 1 × 1 augmentedImageDatastore |
| L2Regularization | 1.00E-04 | ValidationFrequency | 3 |
| Gradient Threshold Method | 'l2norm' | ValidationPatience | 5 |

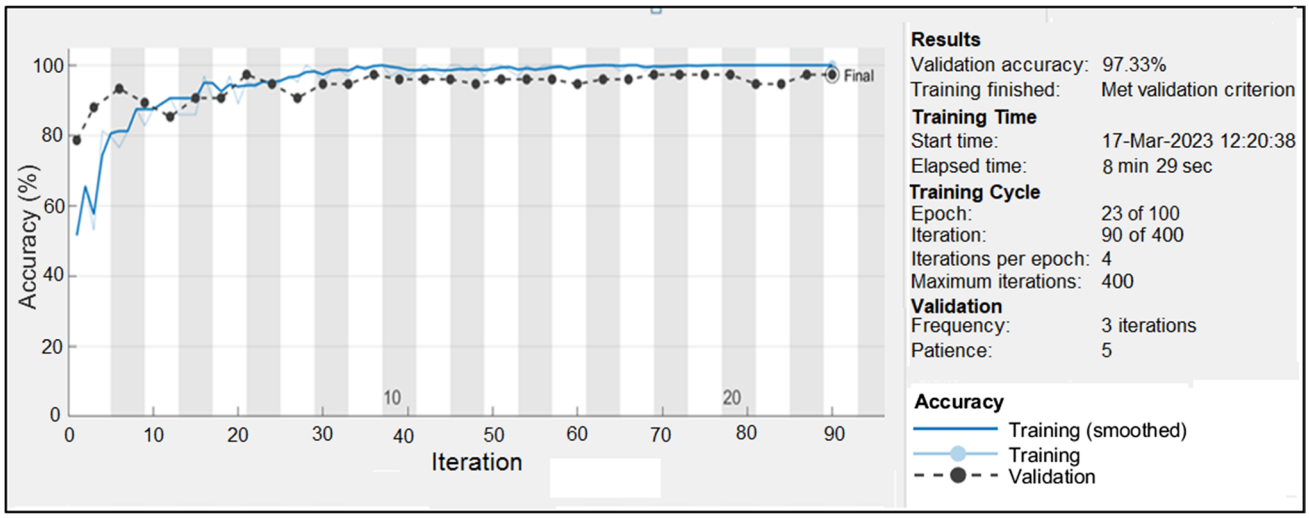


**Fig. 6** Output obtained during the training progress

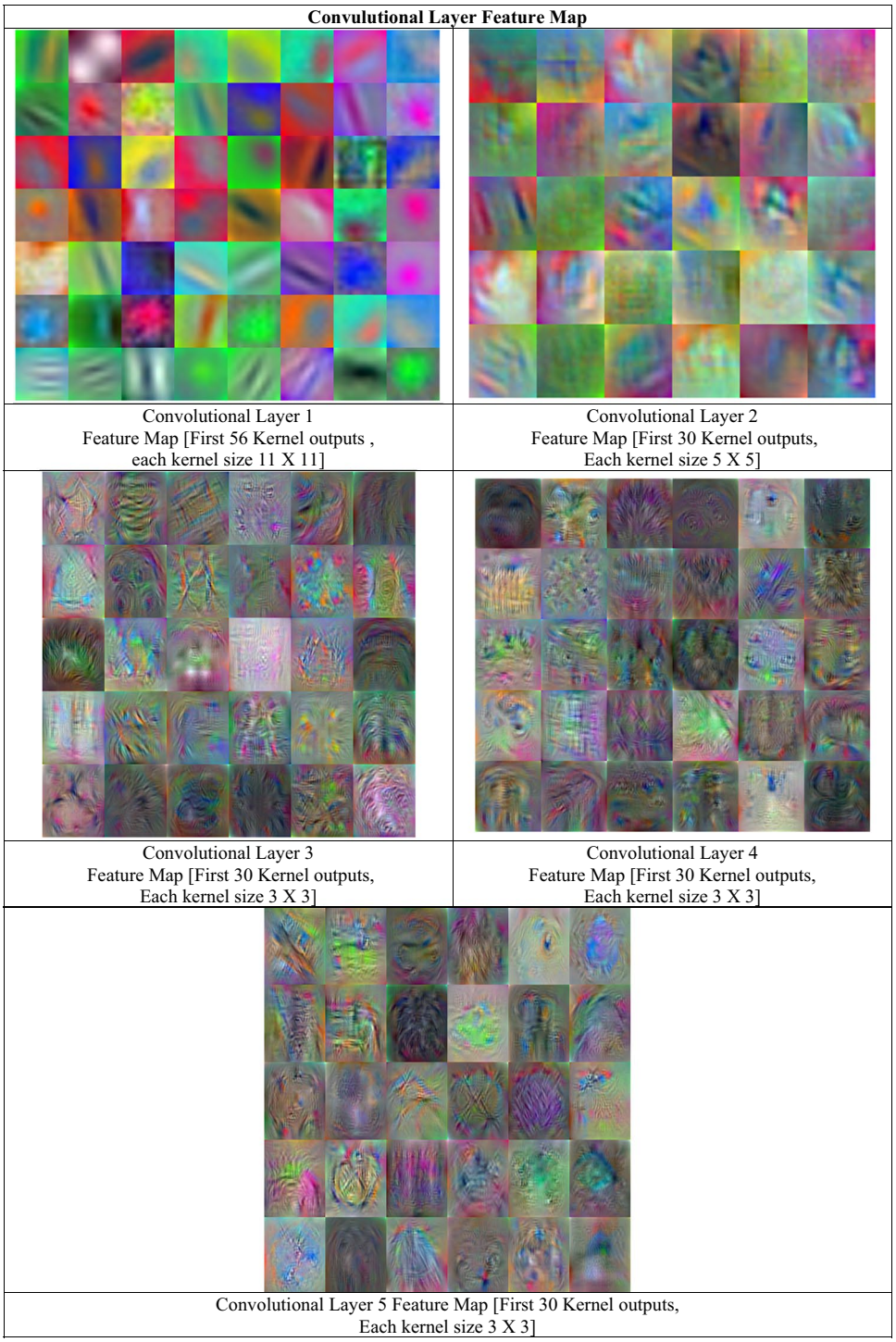


**Fig. 7** Feature map view obtained from the proposed algorithm after CNN training on entire dataset

The following output is obtained when the single image features are observed as input to the Alexnet architecture. Input image size is $227 \times 227 \times 3$. There are usually many kernels of the same size in each convolutional layer. Convolutional layer $C_1$ includes 96 kernels of size $11 \times 11$, applied with a stride of 4 and padding of '0'. So, the output image is of size $55 \times 55 \times 96$ (one channel for each kernel). Convolutional layer $C_2$ includes 256 kernels of size $5 \times 5$ applied with a stride of 1 and padding of 2. So, the output image is of size $27 \times 27 \times 96$ (one channel for each kernel). Convolutional layer $C_3$ includes 384 kernels of size $3 \times 3$ applied with a stride of 1 and padding of 1. So, the output image is of size $11 \times 11 \times 384$ (one channel for each kernel). Convolutional layer $C_4$ includes 384 kernels of size $3 \times 3$ applied with a stride of 1 and padding of 1. So, the output image is of size $13 \times 13 \times 384$ (one

channel for each kernel). Convolutional layer $C_5$ includes 256 kernels of size $3 \times 3$ stride of 1 and padding of 1. So, the output image is of size $13 \times 13 \times 256$ (one channel for each kernel). Various types of kernel (convolutional filters) are applied on the input image to extract the required features. Convolutional layers 1 and 2 describe lower-level image descriptors as shown in Fig. 8.

These images mostly contain edges and colors, which indicates that the filters at layer 'conv1' are edge detectors and color filters. The edge detectors are at different angles, which allow the network to construct more complex features in the later layers. As one moves to further layer, higher layers in the network might build upon these representations

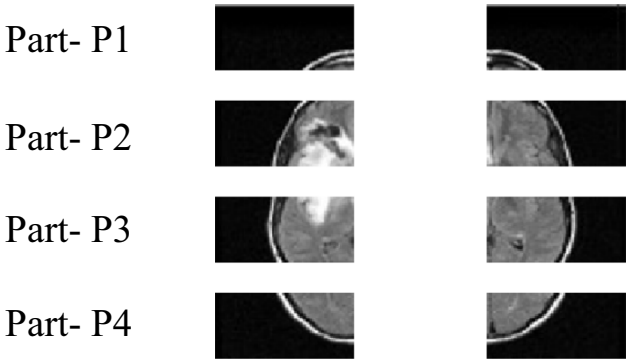


**Fig. 9** Partitioning of the image in eight parts

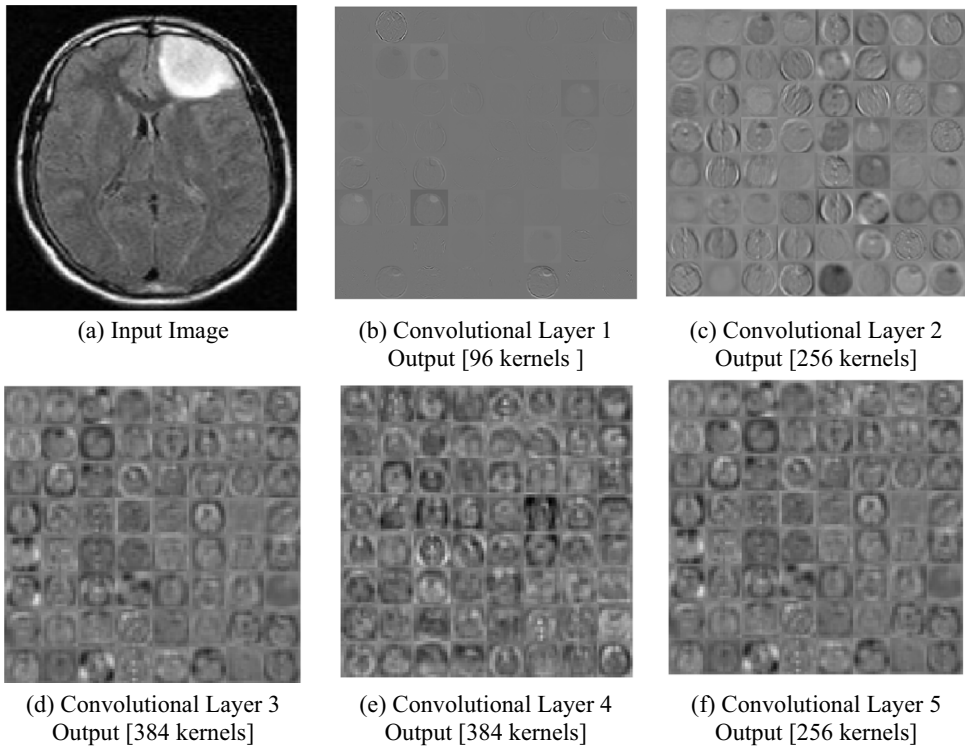

**Fig. 8** Feature maps obtained from the proposed algorithm for single image as input

(a) Input Image

(b) Convolutional Layer 1 Output [96 kernels ]

(c) Convolutional Layer 2 Output [256 kernels]

(d) Convolutional Layer 3 Output [384 kernels]

(e) Convolutional Layer 4 Output [384 kernels]

(f) Convolutional Layer 5 Output [256 kernels]

**Table 6** Statistical features of each part — Sample Image 1

| Part No. | Left Side Statistical Features | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | First Order | | | | | Second Order | | | | |
| | Mean | Standard deviation | Skewness | Kurtosis | Energy | Homogeneity | Abs. Value | Inertial Contrast | Inverse difference | Maximum probability |
| $P_1$ | 18.139 | 293.7688 | -1.3E+11 | -2.2E+10 | 0.097767 | 3085.2833 | 173 | 253 | 3081.3588 | 0 |
| $P_2$ | 72.451 | 5105.191 | -4.85E+16 | 7.19E+19 | 0.029958 | 2788.5333 | 869 | 1325 | 2769.7914 | 71 |
| $P_3$ | 81.919 | 6547.894 | -1.49E+17 | 1.72E+20 | 0.020334 | 2803.1166 | 902 | 1698 | 2782.1873 | 56 |
| $P_4$ | 41.547 | 1644.125 | -3E+14 | 1.23E+16 | 0.050577 | 2966.8333 | 494 | 886 | 2951.4942 | 3 |

to represent larger structures. Convolutional layer 5 (higher layer) might represent whole objects.

To check the significance of selected features, the image is divided into eight partitions as shown in Fig. 9.

In the case of image as shown in Fig. 9, tumor is split into two partitions P2 and P3. The statistical features obtained of each partition are presented in Table 6. The various first-order and second-order statistical features are used for the experimentation.

As shown in Table 6, image shown in Fig. 9 contains tumor spread into partition 2 and 3. In case of partitions 1 and 4, very minor change in statistical features is observed, but in the case of partitions 2 and 3 prominent change in statistical values is observed.

### 4.3 Comparison of the result

Alexnet and VGG-16 architectures are implemented in the proposed work. The effect of variation of the training functions is studied and listed in Table 7. During experimentation, the initial learning rate is also changed; as modelling the architecture (hyperparameter tuning) plays a crucial part in the implementation of CNN algorithms. The analyzed training functions are Adam (adaptive moment estimation) and RMSprop (root mean squared propagation).

The best training parameters are obtained for CNN training using Alexnet architecture with TL being with 100 epochs with the mini-batch size of 64 image instances and

**Table 7** Results of the proposed algorithm using Alexnet architecture [Adam optimizer]

| Parameters / Architecture | | ALEXNET | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Training Function | | Adam [Adaptive Moment Estimation] | | | | | | | |
| Learning Rate | | 3.0E -3 | 3.0E -4 | 3.0E-5 | 3.0E -6 | 3.0E -7 | 2.0E -4 | 2.0E -5 | 2.0E -6 |
| Performance Parameter | Accuracy | 56 | 94.67 | 90.67 | 76 | 98.67 | 93.33 | 90.67 | 94.67 |
| | Normalized Error Rate | 7 | 0.1 | 0.2 | 0.4 | 0.45 | 0.2 | 0.3 | 0.2 |
| | Epochs | 4 | 4 | 10 | 26 | 100 | 18 | 12 | 84 |
| | Iterations | 15 | 66 | 39 | 102 | 400 | 69 | 45 | 336 |
| | Elapsed Time In min | 1.7 | 5.4 | 3.36 | 7.35 | 29.56 | 7.47 | 5.4 | 31.47 |

**Table 8** Results of the proposed algorithm using Alexnet Architecture [Rmsprop optimizer]

| Parameters / Architecture | | ALEXNET | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Training Function | | Rmsprop [Root Mean Square Propagation] | | | | | | |
| Learning Rate | | 3.0E-3 | 3.0E-5 | 3.0E-6 | 2.0E-3 | 2.0E-5 | 2.0E-6 | 2.0E-7 |
| Performance Parameter | Accuracy | 56 | 93.33 | 89.33 | 56 | 89.33 | 90.67 | 80 |
| | Normalized Error | 7.5 | 0.1 | 0.22 | 7 | 0.1 | 0.3 | 0.4 |
| | Rate | | | | | | | |
| | Epochs | 6 | 15 | 19 | 4 | 10 | 21 | 100 |
| | Iterations | 21 | 60 | 75 | 15 | 39 | 84 | 400 |
| | Elapsed Time In min | 2.23 | 4.15 | 5.26 | 1.7 | 2.56 | 6.8 | 31.2 |

**Table 9** Results of the proposed algorithm using Alexnet architecture

| Network | Training Function (Optimizer) | Learning Rate | Performance Parameter | | | | |
|---|---|---|---|---|---|---|---|
| | | | Accuracy | Normalized Error Rate | Epochs | Iterations | Elapsed Time (min) |
| ALEXNET | ADAM | 3.00E-03 | 56.00 | 7 | 4 | 15 | 1.7 |
| | | 3.00E-04 | 94.67 | 0.1 | 4 | 66 | 5.4 |
| | | 3.00E-05 | 90.67 | 0.2 | 10 | 39 | 3.36 |
| | | 3.00E-06 | 76.00 | 0.4 | 26 | 102 | 7.35 |
| | | 3.00E-07 | 98.67 | 0.45 | 100 | 400 | 29.56 |
| | | 2.00E-02 | 44.00 | 7.4 | 6 | 24 | 2.42 |
| | | 2.00E-03 | 44.00 | 8.4 | 7 | 27 | 2.15 |
| | | 2.00E-04 | 93.33 | 0.2 | 18 | 69 | 7.47 |
| | | 2.00E-05 | 90.67 | 0.3 | 12 | 45 | 5.4 |
| | | 2.00E-06 | 94.67 | 0.2 | 84 | 336 | 31.47 |

**Table 10** Results of the proposed algorithm using VGG16 with SGDM training function

| Network | | VGG 16 | | | | |
|---|---|---|---|---|---|---|
| Batch Size | | 10 | | | | |
| Training Function (Optimizer) | | SGDM | | | | |
| Learning rate | | 1.00E-03 | 1.00E-04 | 1.00E-05 | 1.00E-06 | 1.00E-07 |
| Performance Parameter | Accuracy | 56.00 | 89.33 | 88.00 | 68.00 | 72.00 |
| | Elapsed Time (min) | 5.41 | 26.5 | 27.55 | 13.7 | 104.19 |
| | Epoch | 1 | 2 | 3 | 2 | 5 |
| | Iterations | 14 | 50 | 74 | 34 | 150 |
| | Validation error rate | 7 | 0.27 | 0.3 | 0.58 | 0.52 |

**Table 11** Results of the proposed algorithm using VGG16 with ADAM training function

| Network | | VGG 16 | | | | |
|---|---|---|---|---|---|---|
| Batch Size | | 10 | | | | |
| Training Function (Optimizer) | | ADAM | | | | |
| Learning rate | | 1.00E-03 | 1.00E-04 | 1.00E-05 | 1.00E-06 | 1.00E-07 |
| Performance Parameter | Accuracy | 44.00 | 68.00 | 90.67 | 77.33 | 42.67 |
| | Elapsed Time (min) | 7.53 | 24.50 | 53.47 | 60.29 | 113.5 |
| | Epoch | 1 | 2 | 3 | 5 | 5 |
| | Iterations | 16 | 52 | 82 | 150 | 150 |
| | Validation error rate | 0.33 | 0.6 | 0.25 | 0.41 | 0.8 |

**Table 12** Results of the proposed algorithm using VGG16 with RMSPROP training function

| Network | | VGG 16 | | | | |
|---|---|---|---|---|---|---|
| Batch Size | | 10 | | | | |
| Training Function (Optimizer) | | RMSPROP | | | | |
| Learning rate | | 1.00E-03 | 1.00E-04 | 1.00E-05 | 1.00E-06 | 1.00E-07 |
| Performance Parameter | Accuracy | 56.00 | 44.00 | 88.00 | 84.00 | 61.33 |
| | Elapsed Time (min) | 14.00 | 8.40 | 25.58 | 52.12 | 59.24 |
| | Epoch | 1 | 1 | 2 | 4 | 5 |
| | Iterations | 28 | 12 | 60 | 108 | 150 |
| | Validation error rate | 3 | 1.8 | 0.3 | 0.4 | 0.45 |

**Table 13** Optimizer performance

| Identifier used => | SVM | ANN | VGG16 | | | Alexnet | | Googlenet | |
|---|---|---|---|---|---|---|---|---|---|
| Optimizer | | | Adam | SGDM | RMSPROP | Adam | RMSPROP | Adam | SGDM |
| Accuracy | 89.53 | 87.7 | 90.67 | 89.33 | 88.00 | 98.67 | 93.33 | 91.49 | 90.43 |
| Learning Rate | - | - | 1.00E-05 | 1.00E-04 | 1.00E-05 | 3.0E-07 | 3.00E-05 | 3.00E-04 | 3.00E-03 |

initial learning rate of 3.00E-07. In Alexnet architecture, the maximum accuracy of 98.67% is achieved with training function of ADAM. Accuracy is calculated using following formula

$$\text{Accurancy} = [\text{TP} + \text{TN}]/[\text{TP} + \text{TN} + \text{FP} + \text{FN}] \quad (1)$$

- TP: True positive: Images with tumor correctly identified as images with tumor
- FP: False positive: Images without tumor incorrectly images with tumor
- TN: True negative: Images without tumor correctly identified as Images without tumor

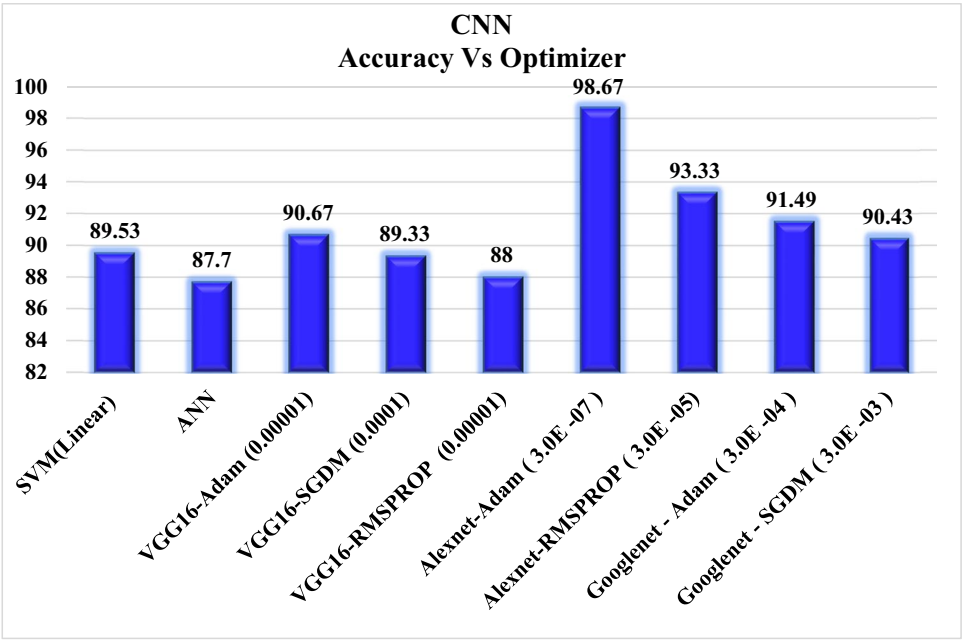


**Fig. 10** Performance of various techniques used for experimentation

- FN: False negative: Images without tumor incorrectly identified as images with tumor

ADAM optimizer effectively minimizes cost function without any parameter tuning [24]. Dropout is applied to improve generalization and performance on the test set. Convolutional layers are always followed by the pooling layers, which limits the capabilities of this network due to the aggressive information loss in pooling. Table 8 shows the results of proposed algorithm using Alexnet architecture.

The best training parameters are obtained for CNN training using Alexnet architecture with transfer learning with 15 epochs with the mini-batch size of 64 image instances and initial learning rate is 3.00E-05. In Alexnet architecture, the maximum accuracy of 98.67 is achieved with training function of ADAM as shown in Table 9. In Alexnet architecture, the maximum accuracy of 93.33 is achieved with training function of rmprop.

The hyperparameters of the VGG-16 ConvNet are tuned, and the obtained results are tabulated as shown in Table 10. The training functions referred in this experiment are SGDM, RMSprop, and ADAM. The VGG 16 ConvNet architecture and optimizers, SGDM, ADAM, and RMSprop are listed in Tables 10, 11, and 12 respectively.

**Table 14** Ten-fold cross validation results (Alexnet)

| Round number | Validation accuracy in % | Normalized error rate training | Epochs | Iterations | Elapsed time (min) |
|---|---|---|---|---|---|
| 1 | 91.89 | 0.012 | 9 | 45 | 3.9 |
| 2 | 89.19 | 0.011 | 8 | 39 | 2.40 |
| 3 | 91.89 | 0.011 | 11 | 51 | 4.8 |
| 4 | 86.49 | 0.013 | 6 | 30 | 2.57 |
| 5 | 97.30 | 0.011 | 10 | 48 | 5.8 |
| 6 | 97.30 | 0.011 | 15 | 75 | 6.22 |
| 7 | 86.49 | 0.012 | 9 | 42 | 4.46 |
| 8 | 100 | 0.010 | 32 | 159 | 13.25 |
| 9 | 89.19 | 0.015 | 10 | 48 | 4.4 |
| 10 | 91.89 | 0.010 | 13 | 63 | 5.34 |

**Table 15** Comparison of obtained results for the convolutional neural network architecture

| Parameters/architectures | Alexnet – TL | | VGG-16 | | |
|---|---|---|---|---|---|
| Training function | ADAM | Rms prop | SGDM | ADAM | Rms prop |
| Learning rate | 3.00E-07 | 3.0E-5 | 1.00E-04 | 1.00E-05 | 1.00E-05 |
| Accuracy | 98.67 | 93.33 | 89.33 | 90.67 | 88.00 |
| Elapsed time in min | 29.56 | 0.1 | 26.5 | 53.47 | 25.58 |
| Epoch | 100 | 15 | 2 | 3 | 2 |
| Iterations | 400 | 60 | 50 | 82 | 60 |
| Error rate | 0.45 | 4.15 | 0.27 | 0.25 | 0.3 |

**Table 16** Comparison of obtained results with approaches used in literature

| Year and Ref | Methodology | Dataset used | Details of testing samples/validation | Parameters Accuracy |
|---|---|---|---|---|
| 2015 [25] | CNN | BRATS 2013<br>BRATS 2015 | Training 80%,<br>Testing 20%, | 0.67 |
| 2018 [26] | U-Net Model and XGBoost | BRATS 2018 | Training 285,<br>Testing 66,<br>and tenfold cross validation | 0.650 |
| 2018 [27] | CNN | BRATS 2018 | Training 285,<br>Testing 66 | 0.710 |
| 2018 [28] | ANN | BRATS 2018 | Training 285,<br>Testing 66 | 0.679 |
| 2018 [29] | CNN | BRATS 2018 | Training 285,<br>Testing 66 | 0.536 |
| 2019 [30] | Deep learning approach training of neural networks | BRATS 2018 (285 patients) | Training 80%,<br>Testing 20%,<br>and fivefold cross validation | 0.765 |
| 2019 [31] | KE-CNN | 233 Patients# | Training 70%,<br>Testing 30%,<br>and tenfold cross validation | 0.936 |
| 2020 [32] | VGG-16 | ImageNet | 1.2 million training images, and 150,000 test images | 0.930 |
| 2020 [33] | CNN | BRATS 2013<br>BRATS 2015 | Training: 395, Testing: 182<br>Total: 577 | 0.961 |
| 2021 [34] | CNN VGG16 | BRATS 2013<br>BRATS 2015 | Training:395, Testing: 182<br>Total: 577 | 0.967 |
| 2021 [35] | Multiscale CNN | 233 Patients# | Training 80%,<br>Testing 20%,<br>fivefold cross validation | 0.973 |
| 2022 [36] | VGG-16 | BRATS 2015 | Training 80%,<br>Testing 20%, | 0.854 |
| 2022 [37] | DCNN | BRATS 2015 | Training 70%,<br>Testing and Validation 30%,<br>tenfold cross validation | 0.886 |
| 2022 [38] | CNN | BRATS 2015 BRATS 2018 | tenfold cross validation | 0.942<br>0.935 |
| 2022 [39] | CNN | BRATS 2018 | Training 80%,<br>Testing 20%,<br>and fivefold cross validation | 0.963 |
| 2022 [40] | CNN | BRATS 2018 | 80% training set, 10% of each validation, and testing set | 0.9790 (T2),<br>0.961 (T1) |
| 2023 [42] | VGG-16,<br>Resnet50,<br>InceptionV3 | BRATS 2013<br>BRATS 2015 | Training 80%,<br>Testing 20%, | 0.975 for VGG-16, 0.95 for Resnet50,<br>0. 915 for InceptionV3 |
| Proposed method [Alexnet-TL] | CNN | BRATS 2013<br>BRATS 2015<br>OPEN I | Training: 496, Testing: 125<br>Total: 621<br>With data augmentation,<br>tenfold cross validation | 0.986 |

# Dataset collected from Nanfang Hospital, Guangzhou, China, and General Hospital, Tianjing Medical University, China [41]

In VGG-16, using training function SGDM maximum accuracy achieved is 89.33 with the Learning rate = 1.00E-4.

The results obtained accuracy = 90.67% with learning rate = 1.00e-05, Number of Epochs = 3, Iteration = 82, and Normalized error rate = 0.25 for VGG-16-TL network with training function ADAM.

In VGG-16, using training function rmsprop, maximum accuracy achieved is 88.00 with the Learning rate = 1.00E-5.

The proposed work is tested with the various optimizers. The results obtained from the optimizer are compared in terms of the accuracy achieved as shown in Table 13 and Fig. 10. The performance of various optimizers used is tabulated in Table 13.

In case of Alexnet architecture, ADAM and RMSPROP optimizers are compared. Maximum accuracy of 98.67% is achieved with ADAM optimizer with Alexnet [24]. In case of VGG-16 architecture, Adam, SGDM, and rmsprop are compared. Maximum accuracy of 90.67 is achieved with Adam optimizer in the case of VGG-16.

The important step in the result authentication is cross validation of experimental results. The cross validation proportion can be varied. Table 14 shows the results obtained for random tenfold cross validation. Table 15 shows comparison of obtained results for convolutional neural network architecture.

The proposed convolutional neural network architecture using Alexnet architecture is modelled using transfer learning approach. In any neural network, hyperparameter tuning is the most vital step. Hence, delicate decision needs to be taken while selecting the values. Understanding the input image characteristics and applying the appropriate hyperparameters is a must.

The kernels available in the Alexnet architecture are found suitable for the brain tumor detection from MRI images. The features extracted using these kernels are also appropriate for the characterization of the tumor. Hence, the proposed work has reached the maximum accuracy. Various researchers' findings are compared with the developed approach shown in Table 16. The investigation is completed on the standard image dataset. The results obtained are approved by the two medical experts. The research in this domain is scattered at various points, like database used, cross validation methods, and performance parameter used (many have used Sensitivity, Dice coefficient, Tanimoto, Jaccard similarity coefficient). Apart from all this diversity, it is tried to give an overall picture of the state of art research carried in this domain. For this purpose, common performance parameter is fixed as "accuracy of the system" and comparison of other developed methods is done with respect to accuracy of system.

In the recent literature, it is found that AlexNet, GoogLeNet, and VGG are most popular pre-trained CNN models and are used in many classification applications. Different approaches were used for the identification of brain tumor. To overcome the drawback of the machine learning approach, CNN architecture is used. In this type of approach, the kernels defined in the convolutional layers are extracting the required features from the input images. The features extracted in this layer are combination of all types of features.

Transfer learning is better than the random initialization to train the pre-trained CNN model when datasets are small. As convolution layer increases, the accuracy increases but at the other side training time also increases.

Further experimentations are done to calculate the dimensions of tumor (i.e., tumor parameters). Few sample image dataset is shown in 11 [43]. The tumor-describing parameters like diameter of tumor, area of tumor, perimeter of tumor, eccentricity of tumor, and circularity of tumor are calculated as shown in Table 17. Diameter gives the mean of major axis and minor axis. It is a scalar value. Area defines quantity of pixels in the region. Perimeter provides the definite figure of

**Table 17** Tumors' dimensions and classifications

| Image | Diameter (mm) | Area ($mm^{2)}$ | Perimeter (mm) | Eccentricity | Circularity |
|---|---|---|---|---|---|
| 1 | 35.1843 | 775 | 166.432 | 0.547083986 | 0.351412882 |
| 2 | 33.37298 | 842 | 108.34 | 0.634372178 | 0.900998501 |
| 3 | 45.74384 | 1579 | 160.112 | 0.659956384 | 0.773613437 |
| 4 | 70.518 | 2335 | 308.445 | 0.874697034 | 0.308262755 |
| 5 | 76.61899 | 4201 | 271.326 | 0.590406902 | 0.716736458 |
| 6 | 42.877 | 1411 | 134.308 | 0.62799665 | 0.982455502 |
| 7 | 40.07757 | 935 | 153.348 | 0.898881309 | 0.499395952 |
| 8 | 35.44647 | 947 | 111.896 | 0.68916205 | 0.949971322 |
| 9 | 44.73175 | 441 | 165.653 | 0.974143244 | 0.201850402 |
| 10 | 22.66749 | 231 | 73.993 | 0.970727683 | 0.529932247 |
| 11 | 84.67897 | 5131 | 300.508 | 0.762677427 | 0.713640646 |
| 12 | 183.9096 | 22,959 | 824.066 | 0.512626257 | 0.424637765 |
| 13 | 80.91509 | 3928 | 675.98 | 0.57064709 | 0.107967594 |
| 14 | 59.98479 | 2335 | 262.82 | 0.683236031 | 0.424580166 |

the pixels in the shape of the nodule. The eccentricity is the proportion of the distance among the foci of the ellipse and its major axis length. The value is ranging from 0 to 1. An ellipse whose eccentricity is 0 is a circle, while 1 is a line segment. Circularity is the roundness of shape which is to 1 only for roundness and it is < 1 for any other shape.

Tumors are classified on the basis of their growth rate. To calculate the growth rate, one must have at least 2 samples of same patients to come across the proper conclusion. In present system, because of dataset limitation, the size of tumor is measured and interpreted in Table 17 (for sample dataset shown in Fig. 11), which shows the dimensions of the nodules (one tumor) in mm. Tumor parameters are calculated for classifying it into various classes. Tumors diameter greater than 10 mm will require special attention of radiologist.

Benign tumors have clearly defined borders and they are composed of harmless cells. Nearby tissues are not infiltrated by the benign brain tumors. On the other hand, distinct borders are absent in the case of malignant brain tumors. The malignant brain tumors tend to grow rapidly and infect other parts of the brain. The brain tumor classification depends on "how rapidly it is growing" and "how likely it is to invade other tissues." World Health Organization grading system classified the brain tumors on the basis of rate of growth into four categories, grades I, II, III, and IV. Grade I tumors are the least malignant and grow slowly. But even a grade I tumor may be life-threatening if it is inaccessible for surgery.

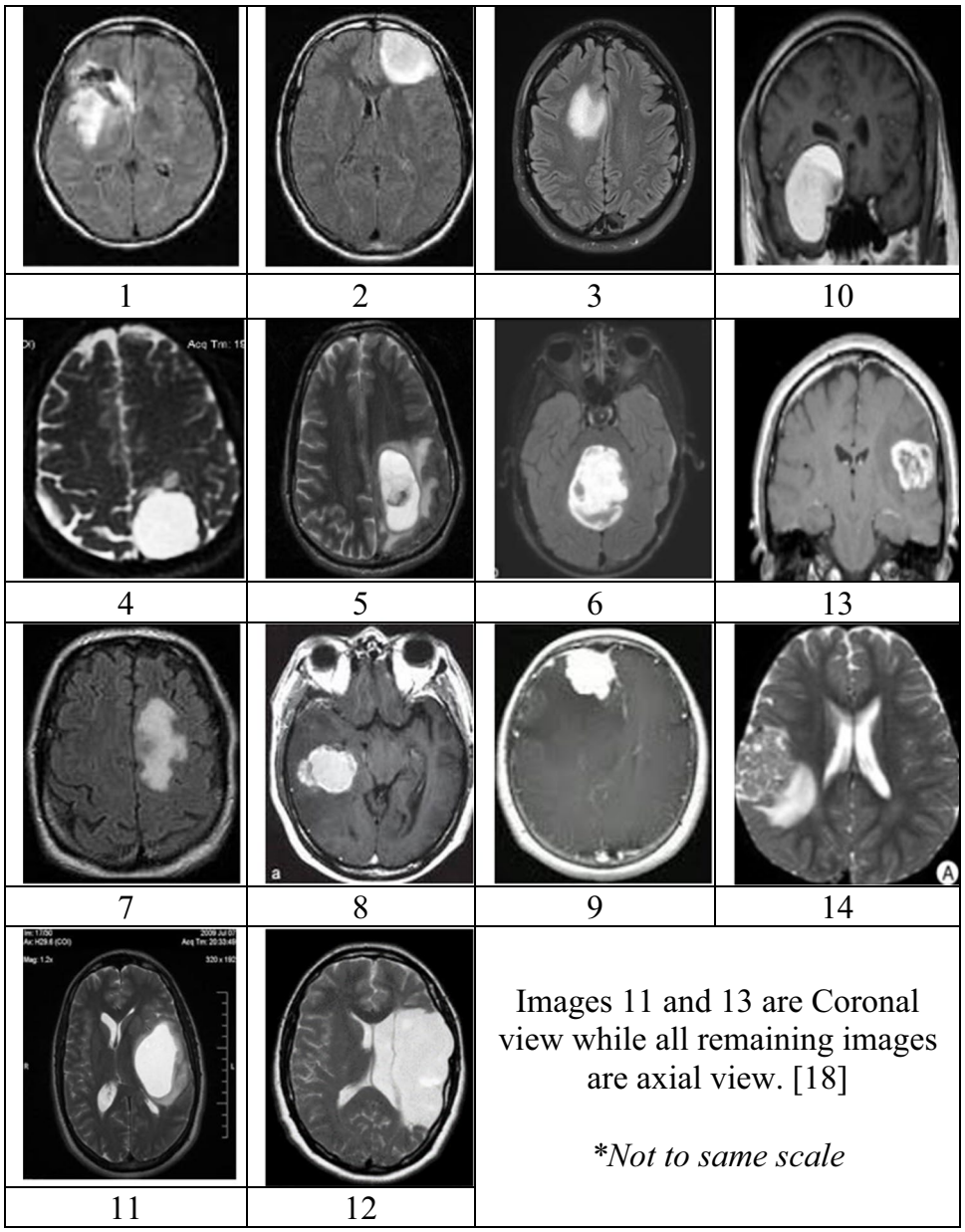


**Fig. 11** Few sample images under test to calculate tumor parameters

Grade II tumors grow slightly faster than grade I tumors and have a little abnormal microscopic appearance. These tumors may attack surrounding normal tissue, and may reappear as a grade III or higher tumor. Grade III tumors are malignant. The chances of recurrence of these tumors are quite high. Grade IV tumors are the most malignant and invade wide areas of surrounding normal tissue.

## 5 Conclusion and future scope

Brain tumors are relatively diverse in their spatial location and structure. Data augmentation is used to explore this variability. A robust CNN-based image processing algorithm is presented for the classification of brain tumor images into normal and abnormal type. The algorithm has been successfully tested on the dataset BRATs 2013, BRATs 2015 and Open-I images.

The presented method is based on CNN which is constructed using convolutional layer with $11 \times 11$ kernels to permit specific features of the images. The results are obtained with accuracy = 90.67% with learning rate = 1.00e-05, No of Epochs = 3, Iteration = 82, Normalized error rate = 0.25 for VGG-16-TL network with training function as ADAM. In VGG-16, using training function rmsprop, maximum accuracy achieved is 88.00 with the Learning rate = 1.00E-5. In VGG-16, using training function SGDM maximum accuracy achieved is 89.33 with the Learning rate = 1.00E-4. The best-performing classifier had an accuracy of 98.67%, with learning rate = 3.00e-07, Normalized error rate = 0.45, No of Epochs = 100, Iteration = 400 for AlexNet-TL network.

From simulation results, it is observed that the highest classification accuracy of 98.67% has been achieved in Alexnet architecture, with training function of ADAM. The proposed methodology is valid for axial, coronal, and sagittal slice images of the brain. Tumor parameters will help the doctors to classify the tumors in various grades defined by WHO. In upcoming algorithm development, the proposed system may be verified with real time images from other dataset along with multiple tumors, to confirm the results in a more general way. The proposed work can be further extended for finding the brain tumor at an early stage, using combination of two CNNs for increased accuracy. Researchers can also explore the use of bio-inspired algorithms in the process of brain tumor detection.

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1007/s11517-023-02820-3.

**Acknowledgements** We thank Dr. Balaji Anvekar, Consultant Neuroradiologist, MRI Dept, S P Institute of Neurosciences, Solapur, and Dr. Mahesh Kasat, Radiologist, Pune, who helped in the authentication of the results obtained through the presented work.

**Author contribution** Study conception, design, manuscript preparation: V K Bairagi; data collection, coding, analysis and interpretation of results: P P Gumaste; analysis and interpretation of results, draft manuscript preparation: Dr. S H. Rajput, Chethan K. S. All authors reviewed and contributed to the final manuscript.

**Data availability** Publically available dataset is used and cited.

**Code availability** Not applicable.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Ethics approval** Not applicable as publically available dataset is used.

## References

1. Krizhevsky A, Sutskever I, Hinton G (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
2. Malathi M, Sinthia P (2019) Brain tumour segmentation using convolutional neural network with tensor flow. Asian Pac J of Cancer Prev 20(7):2095. https://doi.org/10.31557/APJCP.2019.20.7.2095
3. Currie G, Hawk KE, Rohren E, Vial A, Klein R (2019) Machine learning and deep learning in medical imaging: intelligent imaging. J Med Imaging Radiat Sci 50(4):477–487. https://doi.org/10.1016/j.jmir.2019.09.005
4. Bairagi VK (2015) Symmetry-based biomedical image compression. J Digit Imaging 28(6):718–726. https://doi.org/10.1007/s10278-015-9779-3
5. Sobhaninia Z, Rezaei S, Noroozi A, Ahmadi M, Zarrabi H, Karimi N, Emami A, Samavi S (2021) Brain tumor segmentation using deep learning by type specific sorting of images. arXiv preprint arXiv: 1809.07786, 1–4. https://doi.org/10.48550/arXiv.1809.07786
6. Razzak MI, Imran M, Xu G (2018) Efficient brain tumor segmentation with multiscale two-pathway-group conventional neural networks. IEEE J Biomed Health Inform 23(5):1911–1919. https://doi.org/10.1109/JBHI.2018.2874033
7. Lundervold AS, Lundervold A (2019) An overview of deep learning in medical imaging focusing on MRI. Z Med Phys 29(2):102–127. https://doi.org/10.1016/j.zemedi.2018.11.002
8. Zhou T, Ruan S, Canu S (2019) A review: deep learning for medical image segmentation using multi-modality fusion. Array 3–4(100004):1–11. https://doi.org/10.1016/j.array.2019.100004
9. Vinoth R, Venkatesh C (2018)Segmentation and detection of tumor in MRI images using CNN and SVM classification, Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, India, J Phys: Conf Ser, pp 21–25. https://doi.org/10.1109/ICEDSS.2018.8544306
10. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision – ECCV 2014. Lect Notes Comput Sci, vol. 8689. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53
11. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409:1556. https://doi.org/10.48550/arXiv.1409.1556

12. Szegedy C et al (2015) Going deeper with convolutions. 2015 IEEE Conf Comp Vis Patt Recognit (CVPR), Boston, MA, 1–9. https://doi.org/10.1109/CVPR.2015.7298594
13. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision – ECCV 2016. Lect Notes Comput Sci, 9908. Springer, Cham. pp 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
14. Mzoughi H, Njeh I, Wali A, Slima MB, BenHamida A, Mhiri C, Mahfoudhe KB (2020) Deep multi-scale 3D convolutional neural network (CNN) for MRI gliomas brain tumor classification. J Digit Imaging 33:903–915. https://doi.org/10.1007/s10278-020-00347-9
15. Hoseini F, Shahbahrami A, Bayat P (2019) AdaptAhead optimization algorithm for learning deep CNN applied to MRI segmentation. J Digit Imaging 32(1):105–115. https://doi.org/10.1007/s10278-018-0107-6
16. Zhao J, Zhang C, Li D, Niu J (2020) Combining multi-scale feature fusion with multi-attribute grading, a CNN model for benign and malignant classification of pulmonary nodules. J Digit Imaging 33(4):869–878. https://doi.org/10.1007/s10278-020-00333-1
17. Paluszek M, Thomas (2020) Practical MATLAB Deep Learning: a project-based approach. Apress. https://doi.org/10.1007/978-1-4842-5124-9
18. NIH.Open-I Medical Dataset. Available at https://openi.nlm.nih.gov/. Accessed 27 Jan 2020
19. Banerjee S, Mitra S, Masulli F, Rovetta S (2018) Brain tumor detection and classification from multi-sequence MRI: study using convnets. Proc. 7th MICCAI BraTS Challenge, Granada, Spain, 170–179
20. Sultan HH, Salem NM, Al-Atabany W (2019) Multi-classification of brain tumor images using deep neural network. IEEE Access 7:69215–69225. https://doi.org/10.1109/ACCESS.2019.2919122
21. Isin A, Direkoğlu C, Şah M (2016) Review of MRI-based brain tumor image segmentation using deep learning methods. Procedia Computer Sci 102:317–324. https://doi.org/10.1016/j.procs.2016.09.407
22. Multimodal Brain Tumor Segmentation BRATS (2013) Available at https://www.smir.ch/BRATS/Start2013. Accessed 27 Jan 2020
23. Multimodal Brain Tumor Segmentation BRATS (2015) Available at https://www.smir.ch/BRATS/Start2015. Accessed 27 Jan 2020
24. Kingma DP, J BA (2014) Adam: a method for stochastic optimization. ICLR 2015, arXiv preprint arXiv:1412.6980. 1–15 https://doi.org/10.48550/arXiv.1412.6980
25. Rao V, Sarabi MS, Jaiswal A (2015) Brain tumor segmentation with deep learning. Proc MICCAI-BRATS Munich Germany 4:56–59
26. Xu X, Kong X, Sun G, Lin F, Cui X, Sun S, Wu Q, Liu J (2018) Brain tumor segmentation and survival prediction based on extended U-Net model and XGBoost. Proc 7th MICCAI BraTS Challenge, Granada, Spain, 525–533
27. Han W-S, Song Han II (2018) Neuromorphic neural network for multimodal brain tumor segmentation and survival analysis. Proc 7th MICCAI BraTS Challenge, Granada, Spain, 171–178
28. Islam M, Jose VJM, Ren H (2018) Batch normalized PixelNet for brain tumor segmentation and survival prediction. Proc 7th MICCAI BraTS Challenge, Granada, Spain, 232–239
29. Gates E, Pauloski JG, Schellingerhout D, Fuentes D (2018) Glioma segmentation and a simple accurate model for overall survival prediction. Proc 7th MICCAI BraTS Challenge, Granada, Spain, 144–152
30. Mlynarski P, Delingette H, Criminisi A, Ayache N (2019) Deep learning with mixed supervision for brain tumor segmentation. J Med Imaging 6:34002. https://doi.org/10.1117/1.JMI.6.3.034002
31. Pashaei A, Sajedi H, Jazayeri N (2018) Brain tumor classification via convolutional neural network and extreme learning machines. Proc Int Conf Comput Knowl Engg Mashhad Iran, 314–319. https://doi.org/10.1109/ICCKE.2018.8566571
32. Singh A, Deshmukh R, Jha R, Shahare N, Verma S, Nilawar A (2020) Brain tumor classification using CNN and VGG16 model. Int J Adv Res Innov Ideas Educ 6(2):1331–1336
33. Sajja VR, Kalluri HK (2020) Classification of brain tumors using convolutional neural network over various SVM methods. Ingenierie des Systemes d'Information 25(4):489–495. https://doi.org/10.18280/isi.250412
34. Sajja VR (2021) Classification of brain tumors using Fuzzy C-means and VGG16. Turk J Comput Math Educ (TURCOMAT) 12(9):2103–2113. https://doi.org/10.17762/turcomat.v12i9.3680
35. Diaz-Pernas FJ, Martinez-Zarzuela M, Anton-Rodríguez M, Gonzalez-Ortega D (2021) A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network. Healthcare (Basel) 9(2):153. https://doi.org/10.3390/healthcare9020153
36. Parmar A (2020) Brain tumor detection using deep learning, MTech thesis Gujarat Technological University, 1–40, https://bvmengineering.ac.in/NAAC/Criteria1/1.3/1.3.4/18CP814_Thesis.pdf
37. Samee NA, Ahmad T, Mahmoud NF, Atteia G, Abdallah HA, Rizwan A (2022) Clinical decision support framework for segmentation and classification of brain tumor MRIs using a U-Net and DCNN cascaded learning algorithm. Healthcare 10(12):2340. https://doi.org/10.3390/healthcare10122340
38. Rashid SN, Hanif M, Habib U, Khalil A, Inam O et al (2022) Early-stage segmentation and characterization of brain tumor. Comput Mater Continua 73(1):1001–1017. https://doi.org/10.32604/cmc.2022.023135
39. Latif G (2022) DeepTumor: framework for Brain MR image classification, Segmentation and Tumor Detection. Diagnostics 12(11):2888. https://doi.org/10.3390/diagnostics12112888
40. Ahuja S, Panigrahi BK, Gandhi TK (2022) Enhanced performance of Dark-Nets for brain tumor classification and segmentation using colormap-based superpixel techniques. J Mach Learn Appl 7:100212
41. [Brain Tumor Dataset] https://figshare.com/articles/dataset/brain_tumor_dataset/1512427
42. Patel M (2023) Brain tumor detection using MRI images, MS Thesis, California State University, San Bernardino, Electronic Theses Projects, and Dissertations. 1602 https://scholarworks.lib.csusb.edu/etd/1602
43. Gumaste PP, Bairagi VK (2020) A hybrid method for brain tumor detection using advanced textural feature extraction. Biomed Pharmacol J 13(1):145–157. https://doi.org/10.13005/bpj/1871


**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

**Vinayak K. Bairagi** has completed M.E. in Electronic in 2007. University of Pune has awarded him a PhD degree in Engineering in 2013. He has teaching experience of 15 years and research experience of 10 years. He has filed 12 patents and 5 copyrights in technical field. He has published more than 61 papers, of which 38 papers are in international journals. His 46 papers are indexed in Scopus and 29 papers are indexed in SCI. He was invited as session chair for 21 national and international conferences. He has also been invited as resource person by 31 colleges for invited talk. He has received four research grants. He is the Chair of IEEE Signal processing society Pune chapter (2020–2023).

**Pratima Purushottam Gumaste** received PhD degree in Electronics Engineering in 2021 from Savitribai Phule Pune University. She is Associate Professor in the Department of Electronics and Telecommunication Engineering at Jayawantrao Sawant College of Engineering, Pune, SPPU. Her principal research interest area is signal processing, embedded systems, and artificial intelligence. She has served as receiver for IEEE-sponsored conferences. She has published five research articles.

**Seema H. Rajput** is presently working as an associate Professor at CCOEW, Pune. She has Completed ME (Electronics) in 2008 and PhD (Electronics & Telecommunications) in 2016 from Nagpur University. She has total working experience of more than 17 years. She has published several papers in reputed international and national journals. She has received grant of Rs. 180,000/- by SPPU, Pune, on topic: Security Issues in Cognitive Radio. She is Start up and innovation cell, faculty coordinator at CCOEW. Awarded with "Best Teacher Award" in 2016 at Sinhgad Academy of Engineering, KondhwaPune.

**Chethan K S** Obtained his B.E in Electronics & Communication in the year 2008 from VTU, Karnataka. Later he obtained his M. Tech in the year 2010 in the stream of Digital Electronics & Communication Systems from VTU, Karnataka. He obtained his Ph.D. degree in 2019 in the field of Biomedical Signal Processing from VTU, Karnataka. He has several reputed publications in his account. Currently he is serving as an Assistant Professor RVITM, Bangalore. His other areas of interest include Artificial Intelligence, Machine Learning, and Data Science.