International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Brain Tumor Detection Using Convolutional Neural Network

Harsh Raj [a], Yugal Sahu [a], Yalagiri Swetha Vani [a], Yamuna B [a], Yatalaboina Siri Praseeda [a]

a Department of Computer Science and Engineering, Presidency University, Bangalore -560064, India

Article Information

Keywords:

Brain Tumor Detection
Magnetic resonance imaging
Convolutional Neural Network
Deep Learning Algorithm

Abstract

Brain tumors refer to the abnormal growth of cells in the brain's tissues. Detecting these tumors and accurately determining their size can be challenging when planning treatment. Magnetic resonance imaging (MRI) utilizes strong magnets to produce detailed images of the body's interior, including the brain. Compared to traditional imaging methods, MRI provides clearer visualization of the brain, making it a common approach for identifying brain tumors..

Although human experts can manually detect brain tumors, this approach is prone to human error and time-consuming. Alternatively, artificial neural networks, specifically algorithms based on Convolutional Neural Networks (CNN), can analyze MRI scans and accurately identify brain tumors. The objective of this project is to develop a "Brain Tumor Detection Model" using CNN. This model can automatically detect brain cancers by leveraging CNN classification techniques, as demonstrated in a study referenced.

1.Introduction

The brain is a highly intricate organ responsible for numerous vital functions such as cognition, memory, emotions, sensory perception, motor skills, vision, and bodily regulation. It plays a crucial role in governing all processes within our body [1].

Brain tumors are abnormal cell growths that form in the brain, posing a serious threat to a person's quality of life. They can significantly impact both physical and mental well-being, and the effects can vary depending on the type of tumor, with some being life-threatening. Recent studies have indicated a notable increase in the incidence of brain tumors over the past two decades across all age groups, particularly in adults where the increase has exceeded 40% [2]. The incidence of nervous system tumors varies widely globally, with reported cases ranging from 0.01 to 12.7 per 100,000 individuals in males and from 0.01 to 10.7 in females. Africa exhibits the least prevalence rates. Hence, early detection of brain tumors holds paramount importance [2].

Medical imaging technology has gained immense significance in both daily medical diagnosis and research. Consequently, the study of medical diagnostic image data, particularly related to brain tumors, has become a crucial area of focus in the medical field [3].

Timely detection of brain tumors can greatly enhance treatment options and improve the chances of survival. However, due to the substantial volume of MRI images generated during medical practice, manually segmenting tumors or lesions from these images is a time-consuming, challenging, and stressful task. MRI, or magnetic resonance imaging, is primarily employed to locate brain lesions or tumors. The segmentation of brain tumors from MRI scans presents one of the key challenges in medical image processing, often requiring extensive data and advanced

Accurately segmenting brain tumors from surrounding soft tissues can be particularly challenging, as the boundaries of tumors may be poorly defined. This task requires significant effort and expertise. To address this challenge, a dataset available on Kaggle consists of a total of 3,264 image files specifically related to brain tumors. This dataset serves as the foundation for training and evaluating models to achieve high accuracy in tumor segmentation. The dataset is organized into two main folders: Testing and Training. Within each folder, there are subfolders dedicated to different types of tumors, including pituitary tumors, meningioma tumors, glioma tumors, and images without tumors.

The training dataset comprises 827 files of pituitary tumors, 822 files of meningioma tumors, 826 files of glioma tumors, and 395 files without tumors. Similarly, the testing dataset includes 74 pituitary tumor files, 115 meningioma tumor files, 100 glioma tumor files, and 105 files without tumors. This diverse dataset allows for comprehensive training and evaluation of models aimed at accurately segmenting brain tumors from medical images.

A study introduced an efficient method that uses convolutional neural networks (CNNs) to automatically identify and classify brain tumors.

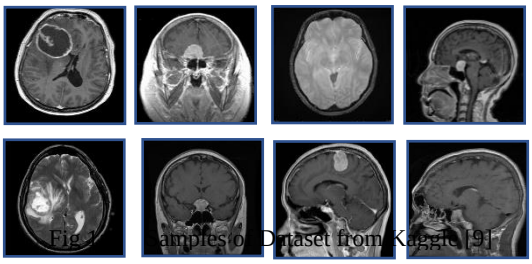

Fig 1 Samples of Dataset from Kaggle [9]

CNNs are deep learning algorithms designed for image analysis. This approach aims to eliminate the need for human involvement in the process. By leveraging CNNs, the proposed method can analyze medical images like MRI scans and accurately detect brain tumors and categorize them. This automation reduces the reliance on manual intervention, saving time and minimizing the risk of human error

a formidable obstacle in this project. This involves the complex task of accurately outlining and identifying the tumor within the provided dataset. This unique and arduous undertaking sets this project apart from others.

In their study, Seetha and Raja (referenced as [4]) employed a conventional approach for brain tumor classification. They utilized Fuzzy C Means (FCM) for segmenting the tumors, extracted texture and shape features, and employed SVM and DNN algorithms for classification. Despite working with a limited dataset that was relatively simple, J. Seetha and S. Selvakumar Raja successfully detected the presence of brain tumors, achieving a training accuracy of 97.6%. It should be emphasized that their research specifically focused on detecting the presence of brain tumors.

Ali et al. (referenced as [5]) utilized transfer learning methods in their research to classify the type of brain tumor. They leveraged pre-trained models such as VGG-15, ResNet-50, and Inception-v3. Their results showed accuracy rates ranging from 75% to 96%. However, it is crucial to highlight that their dataset was relatively small and only allowed for binary predictions regarding the presence of brain tumors.

In their study, Pashaei et al. (referenced as [6]) employed a range of convolutional neural network (CNN) techniques to classify diverse types of brain tumors. Their CNN model consisted of 4 convolutional layers, 4 pooling layers, 1 fully connected layer, and additional intermediate layers for data normalization. Their methodology resulted in an accuracy of 81.09% for classifying brain tumors.

In their research, Parveen et al. (referenced as [7]) utilized a hybrid approach combining Fuzzy C-Means and SVM methods. They applied Fuzzy C-Means for segmenting different regions of the brain and accurately identifying tumor areas, achieving positive outcomes. After the segmentation stage, they employed the Gray Level Run-Length Matrix (GLRLM) to extract meaningful features from the images. The purpose of feature extraction was to identify relevant image characteristics that would aid in the classification process. The classification technique employed in their methodology was SVM. Their approach achieved an accuracy rate of 83.33%.

In their research, Wu et al. (referenced as [8]) introduced a innovative approach for tracking tumor objects in magnetic resonanc (MR) brain images using the K-means clustering technique. The fir step of their method involved converting the original gray-level M image into a color space image. They achieved this by utilizing th CIELab color model, which is specifically designed to represe colors based on human vision.

In order to capture both the intensity and color details, the researche converted the MR image into the CIELab color space. Th conversion allowed them to differentiate tumor objects from oth elements present in the MR image. They then utilized the K-mea clustering algorithm to group the pixels within the color space imag

Through color-based clustering, the algorithm effectively groupe pixels, allowing for the distinction of tumor objects from the imag background and other structures. This process greatly facilitated th identification and tracking of the tumor objects across the entire MI image.
The methodology introduced by Ming-Ni Wu et al. successfully achieved the separation and tracking of tumor objects in MR brain images. This was accomplished through the utilization of the CIELab color model and the K-means clustering technique.

3. Proposed Methodology Using CNN

CNNs are widely utilized for image processing and fall under the category of deep neural networks. They are composed of node layers, which typically include an input layer, one or more hidden layers, and an output layer. The objective of our study was to create an exemplary model using a CNN architecture that could effectively classify the type of tumor from 2D Brain MRI images.

A CNN architecture with 22 layers has been specifically designed for brain tumor detection. The comprehensive model includes a total of 23 stages, incorporating multiple hidden layers. Notably, this model has exhibited impressive abilities in accurately detecting and localizing tumors within the brain.

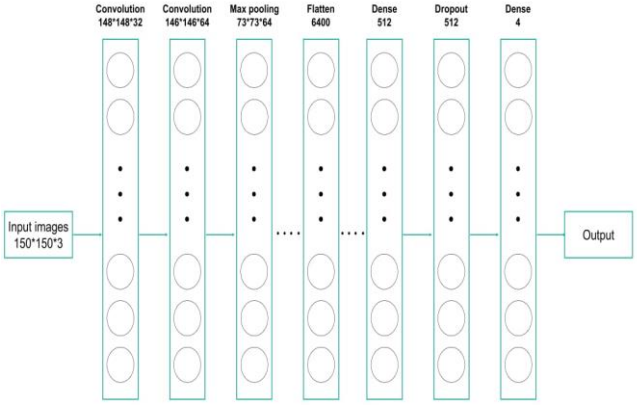

Fig 3a. Proposed Methodology for tumor detection using 22-Layer

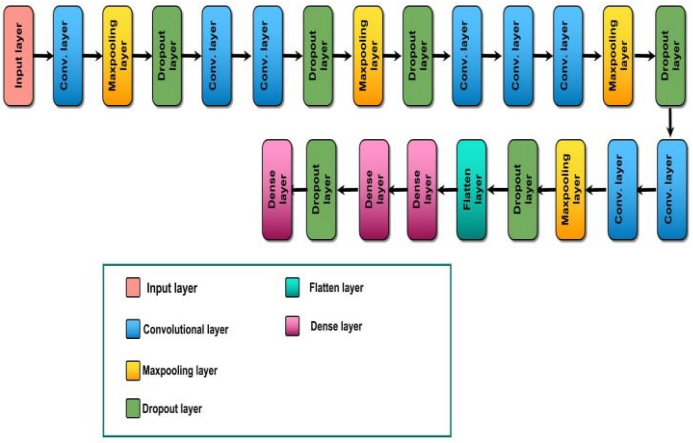

Fig 3b. Working Flow Devised for Proposed CNN

The model's first convolutional layer applies 32 filters with a 3x3 field of view to the input pictures. It makes use of the ReLU activation feature. (150, 150, 3) is the input shape for this layer, where 3 stands for the RGB colour channels.

The Rectified Linear Unit (ReLU) activation function was used in all convolutional layers to streamline the model design and alleviate the vanishing gradient issue. This choice of activation function enables effective computing and aids in resolving the vanishing gradient issue, which might impede the deep neural network training process. The output of the first convolutional layer (148*148*32) is convolutioned using 64 filters of size 3x3 in the second convolutional layer.The output shape of this layer is 146*146*64.

The output from the preceding layer is down sampled by the third layer max pooling layer in the network with a pool size of 2x2. By choosing the highest value inside each 2x2 frame, the feature maps' spatial dimensions are reduced. This pooling process helps to extract important information from the data while lowering computational cost. The output shape of this layer is 73*73*64.
Figure 3c, which was produced from the model, may be used to establish the output shapes for all layers.function called summary. The model architecture, including the input and output shapes of each layer, is briefly described in this synopsis. One can quickly determine the dimensions of the output tensors at each layer of the model by looking at Figure 1.
The Dropout layer, the fourth layer, is intended to prevent neural networks from being overfit. This is accomplished by randomly deactivating a certain number of input units—in this case, 30%—during the training phase. It prevents overfitting by reducing the dependency and interdependence between neurons in this way.

Using 64 3x3 filters, the following layer convolutionally processes the output of the preceding layer.

Using 64 filters of size 3x3, Convolutional Layer 4 convolutionally transforms the output of the preceding layer.
Following this, a Dropout Layer with a Dropout Rate of 0.3 applies once more to Dropout Regularisation, just like the Dropout Layer before it.

Max pooling is done on the output of the preceding layer by the following layer, the Max Pooling Layer.

Similar to the previous Dropout Layers, this one also applies dropout regularisation at a rate of 0.3.

Utilising 128 filters of size 3x3, Convolutional Layer 5 convolutionally transforms the output of the preceding layer.

Using 128 filters of size 3x3, Convolutional Layer 6 convolutionally transforms the output of the preceding layer.

Using 128 filters of size 3x3, Convolutional Layer 7 conducts convolution on the output of the preceding layer.

The output of the preceding layer is subjected to maximum pooling in the following layer.
Similar to the previous Dropout Layers, this one also applies dropout regularisation at a rate of 0.3.
Utilising 128 filters of size 3x3, Convolutional Layer 8 convolutionally transforms the output of the preceding layer.

Using 256 3x3-pixel filters, Convolutional Layer 9 convolutionally transforms the output of the preceding layer.

The output of the preceding layer is subjected to maximum pooling in the following layer. Similar to the previous Dropout Layers, this one also applies dropout regularisation at a rate of 0.3.

The previous layer's output is converted into a 1-dimensional vector by the flatten layer. The multidimensional feature maps are transformed into a format that can be fed into the thick layers.
Dense Layer 1 is a fully connected layer with 512 units. Dense Layer 2 is another fully connected layer with 512 units. In both layers the ReLU activation function is used.
Similar to the previous Dropout Layers, this one also applies dropout regularisation at a rate of 0.3.

The final layer of the model, Dense Layer 3 (Output Layer), has 4 units, which represent the number of classes. The probability that each class in the input exists are provided by the softmax activation function.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 148, 148, 32)      896

 conv2d_10 (Conv2D)          (None, 146, 146, 64)      18496

 max_pooling2d_4 (MaxPooling  (None, 73, 73, 64)       0
 2D)

 dropout_6 (Dropout)         (None, 73, 73, 64)        0

 conv2d_11 (Conv2D)          (None, 71, 71, 64)        36928

 conv2d_12 (Conv2D)          (None, 69, 69, 64)        36928

 dropout_7 (Dropout)         (None, 69, 69, 64)        0

 max_pooling2d_5 (MaxPooling  (None, 34, 34, 64)       0
 2D)

 dropout_8 (Dropout)         (None, 34, 34, 64)        0

 conv2d_13 (Conv2D)          (None, 32, 32, 128)       73856

 conv2d_14 (Conv2D)          (None, 30, 30, 128)       147584

 conv2d_15 (Conv2D)          (None, 28, 28, 128)       147584

 max_pooling2d_6 (MaxPooling  (None, 14, 14, 128)      0
 2D)

 dropout_9 (Dropout)         (None, 14, 14, 128)       0

 conv2d_16 (Conv2D)          (None, 12, 12, 128)       147584

 conv2d_17 (Conv2D)          (None, 10, 10, 256)       295168

 max_pooling2d_7 (MaxPooling  (None, 5, 5, 256)        0
 2D)

 dropout_10 (Dropout)        (None, 5, 5, 256)         0

 flatten_1 (Flatten)         (None, 6400)              0

 dense_3 (Dense)             (None, 512)               3277312

 dense_4 (Dense)             (None, 512)               262656

 dropout_11 (Dropout)        (None, 512)               0

 dense_5 (Dense)             (None, 4)                 2052

=================================================================
Total params: 4,447,044
Trainable params: 4,447,044
Non-trainable params: 0
```

Fig 3c. Model synopsis

Stage	Model-parameter	
	Number of Epochs	20
	Number of steps in each Epoch	83

4. Mathematical Model of Activation Function Used

4.1 ReLU Function

ReLU is an abbreviation for rectified linear unit, a type of non-linear activation function frequently employed in neural networks. One of its key benefits is the ability to selectively activate neurons, as they are only deactivated when the output of the linear transformation is zero. In mathematical terms, ReLU can be expressed as the maximum value between zero and the input value:

$$f(x) = \max(0, x).$$

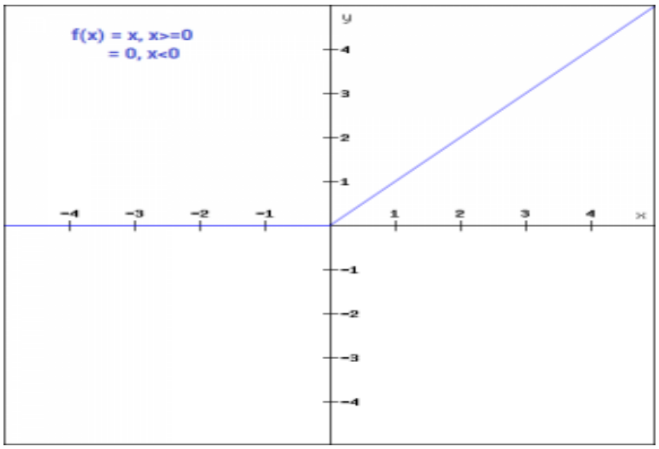

Fig 4a. ReLU Activation Plot [10]

Compared to other activation functions, ReLU is known for its efficiency due to the selective activation of neurons. Unlike other activation functions where all neurons may be activated simultaneously, ReLU activates only a subset of neurons at any given time. This selective activation helps address the problem of gradient values becoming zero, which can occur with other activation functions.

When the gradient becomes zero, the weights and biases of the neurons are not updated during the backpropagation step in neural network training, leading to ineffective learning. By using ReLU, this issue is mitigated as the non-zero gradient allows for proper weight and bias updates, facilitating effective training of the neural network.

4.2 Softmax Function

The softmax function is created by combining multiple sigmoid functions. While a sigmoid function produces values between 0 and 1 that can be interpreted as probabilities for a single class, the softmax function is designed for multiclass classification tasks. It calculates the probabilities across all individual classes for each data point.

In a multiclass classification model, the output layer consists of neurons, with each neuron representing a specific class in the target variable. The softmax function is applied to the outputs of these neurons, normalizing them into a probability distribution across all classes.

The probability of a class can be expressed as the output of the softmax function. It ensures that the probabilities of all classes sum up to 1, allowing for a meaningful interpretation of the model's predictions. The softmax function helps determine the likelihood of each class given the input data, enabling effective multiclass classification.

Probability can be expressed as:-

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$ [10]

In a neural network or model specifically created for multiclass classification, the output layer is structured to have a neuron dedicated to each class present in the target variable.

4.3 Sigmoid Function

The sigmoid function is a widely used activation function in neural networks because of its non-linear characteristics. It transforms input values into a range between 0 and 1. Mathematically, the sigmoid function can be represented as f(x) = 1 / (1 + e^-x). It has a smooth S-shaped curve and is continuously differentiable.

The derivative of the sigmoid function is given by

$$f'(x) = 1 - \text{sigmoid}(x).$$

A drawback of the sigmoid function is its lack of symmetry around zero, which causes all output values of neurons to have the same sign. However, this limitation can be addressed by scaling the sigmoid function, thereby enhancing its performance.

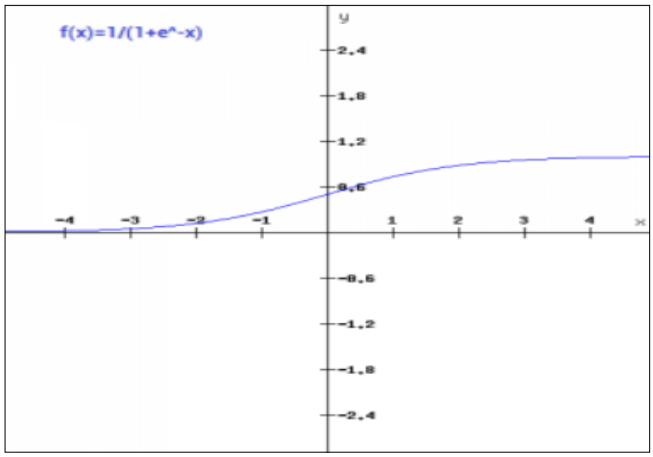

Fig 4b. Sigmoid Activation Function Plot [10]

Once we integrated the Adam optimizer as the loss function, we proceeded to compile the model and evaluate its accuracy in detecting tumors. The Adam optimizer, also referred to as Adaptive Moment Estimation, is highly acclaimed for its efficiency and effectiveness in training neural networks.

Fig 5b. Relationship between Epochs andValidation Accuracy

5. Experimental Result

Classification Using CNN

A proposed approach using a 22-layer convolutional neural network (CNN) and a large dataset has yielded impressive results in the detection of tumors. The CNN architecture includes 8 convolutional layers, 4 max pooling layers, 6 dropout layers, 1 flatten layer, and 3 dense layers. To account for translation invariance in CNNs, data augmentation techniques were employed prior to training the model. The model's performance was assessed on four types of brain tumors: pituitary, meningioma, glioma, and no tumor. For training purposes, 90% of the dataset was used, while the remaining images were reserved for testing. The model achieved a validation accuracy of 89.8% and a training accuracy of 96.67%. These results indicate the model's efficacy in accurately identifying brain tumors.

Following table is obtained after training the model:-

Epoch	Training Accuracy
1	0.342
2	0.535
3	0.606
4	0.645
5	0.701
6	0.730
7	0.768
8	0.807
9	0.852
10	0.864
11	0.885
12	0.896
13	0.909
14	0.922
15	0.912
16	0.945
17	0.939
18	0.943
19	0.958
20	0.966

Fig 5a. Relationship between Epochs and Training Accuracy

Epoch	Validation Accuracy
1	0.510
2	0.568
3	0.598
4	0.636
5	0.687
6	0.775
7	0.751
8	0.792
9	0.758
10	0.789
11	0.826
12	0.853
13	0.812
14	0.850
15	0.880
16	0.840
17	0.833
18	0.860
19	0.897
20	0.874

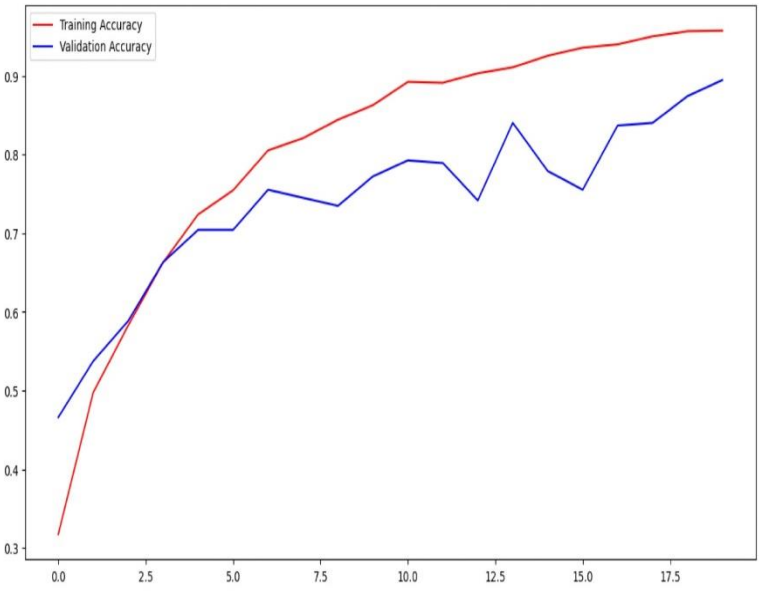

Fig 5c. Training and Validation Accuracy

During the training of the model, the following values were observed for training loss and validation loss:

Training Loss: The training loss refers to the average loss value computed during the training process. It indicates how well the model is fitting the training data.

Validation Loss: The validation loss is the loss value calculated on a separate validation set that consists of data not seen during training. It is used to assess the model's performance on unseen data and determine if it is overfitting or generalizing well.

Epoch	Training Loss
1	1.724
2	1.066
3	0.920
4	0.820
5	0.723
6	0.665
7	0.585
8	0.494
9	0.401
10	0.353
11	0.297
12	0.277
13	0.244
14	0.208
15	0.239
16	0.156
17	0.181
18	0.168
19	0.126
20	0.091

Fig 5d. Relationship between Epochs and Training loss

Epoch	Validation Loss
1	1.250
2	0.958
3	0.918
4	0.802
5	0.741
6	0.622
7	0.643
8	0.533
9	0.532
10	0.587
11	0.535
12	0.430
13	0.508
14	0.565
15	0.423
16	0.485
17	0.583
18	0.461
19	0.374
20	0.376

Fig 5e. Relationship between Epochs and validation loss

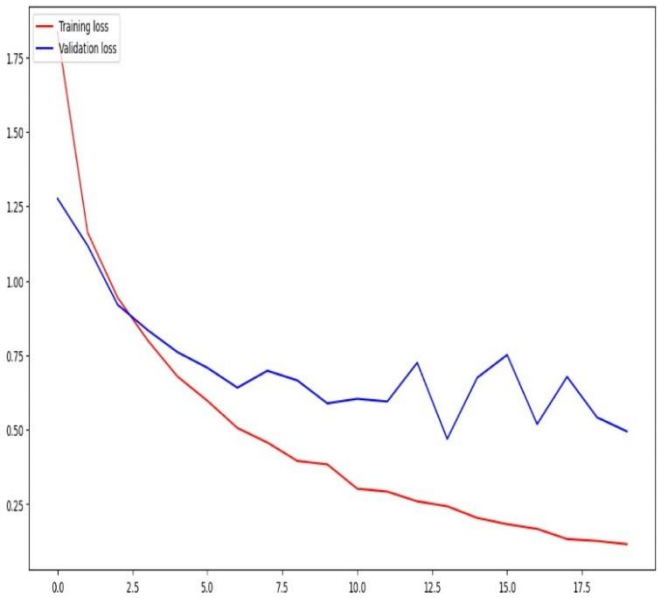

Fig 5f. Training and validation Loss

6. Conclusion

Accurate detection and classification of brain tumors is of utmost importance due to the potential consequences of misdiagnosis. Early detection is critical, making the development of effective methodologies crucial. In our proposed approach, we incorporated a fully connected CNN layer that demonstrated an accuracy of 89.8%, with a training accuracy of 96.67%.

To train our model, we utilized a comprehensive dataset comprising 3264 MRI image samples. We partitioned 90% of the dataset for training the model, while the remaining 10% was dedicated to evaluating the model's performance during testing.

By utilizing this methodology and dataset, we aimed to enhance the accuracy and reliability of brain tumor detection and classification, thereby contributing to improved patient outcomes.

7. Reference

1. Qureshi, S.A.; Raza, S.E.A.; Hussain, L.; Malibari, A.A.; Nour, M.K.; Rehman, A.U.; Al-Wesabi, F.N.; Hilal, A.M. Intelligent Ultra-Light Deep Learning Model for Multi-Class Brain Tumor Detection. Appl. Sci. 2022, 12, 3715

2. Brain cancer in the world an epidemiological review", wcrj, 2019.

3. Kasban, hany & el-bendary, mohsen & salama, dina. (2015). "a comparative study of medical imaging techniques". International journal of information science and intelligent system. 4. 37-58.j. clerk maxwell, a treatise on electricity and magnetism, 3rd ed., vol. 2. Oxford: clarendon, 1892, pp.68–73

4. J. Seetha and S. Selvakumar Raja " Brain Tumor Classification Using Convolutional Neural Networks" Biomedical & Pharmacology Journal, September 2018. Vol. 11(3), p. 1457-1461

5. Hassan Ali Khan , Wu Jue, Muhammad Mushtaq and Muhammad Umer Mushtaq "Brain tumor classification in MRI image using convolutional neural network" Mathematical Biosciences and Engineering Volume 17, Issue 5, 6203–6216 September 2020

6. Ali pashaei , Hedieh sajedi and Niloofar jazayeri "Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines" 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2018, pp. 314-319, doi: 10.1109/ICCKE.2018.8566571

7. Parveen and A. Singh, "Detection of brain tumor in MRI images, using combination of fuzzy cmeans and SVM," 2nd Int. Conf. Signal Process. Integr. Networks, SPIN 2015, pp. 98–102,2015

8. Ming-Ni Wu Chia-Chen Lin Chin-Chen Chang "Brain Tumor Detection Using Color-Based K-Means Clustering Segmentation" IEEE Southwest Symposium on Image Analysis and Interpretation, 2006, pp. 61-65 2007

9. Kaggle Dataset:- https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri?select=Testing

10. Siddharth Sharma, Simone Sharma and Anidhya Athaiya " Activation Functions In Neural Networks" International Journal of Engineering Applied Sciences and Technology,

2020 Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316
Published Online April 2020

11. T. Hossain, F. S. Shishir, M. Ashraf, M. A. Al Nasim and F. Muhammad Shah, "Brain Tumor Detection Using Convolutional Neural Network," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ICASERT.2019.8934561.

12. Hossam H. Sultan , Nancy M. Salem , And Walid Al-Atabany ”Multi-Classification of Brain Tumor Images Using Deep Neural Network” in IEEE Access, vol. 7, pp. 69215-69225, 2019, doi: 10.1109/ACCESS.2019.2919122.