



## OPTIMISED RESNET50 FOR MULTI-CLASS CLASSIFICATION OF BRAIN TUMORS

A SRAVANTHI PEDDINTI \*AND SUMAN MALOJI †

**Abstract.** Categorizing brain cancers, including glioma, meningioma, and pituitary tumors, based on magnetic resonance imaging (MRI) images presents a significant challenge. Deep learning and machine learning techniques have shown promise in enhancing image categorization. To address this challenge, we leverage the power of the optimized ResNet50 model. Our approach involves classifying medical images using Convolutional Neural Network (CNN) features, which are then compared with the ResNet50 model. The primary goal is to detect brain tumors at an early stage using an advanced deep-learning model. We utilize an accessible dataset from Figshare, containing MRI images of the three distinct categories of brain tumors. Existing brain tumor models face limitations in handling multi-class problems and early-stage diagnosis. Therefore, we propose a fully automated approach employing Convolutional Neural Networks (CNN) to extract diverse properties from brain MRI scans. This method aims to provide accurate tumor diagnosis, even with a high number of classes and limited information in MRI data. Our proposed model involves the creation of identification blocks within a four-layered primary architecture, followed by testing and assessment of the interconnected layers. The results demonstrate that our model outperforms existing methods, achieving an impressive overall classification accuracy of 99.03

**Key words:** Brain Tumor, Convolutional Neural Network, health care, tumor segmentation, data augmentation

**1. Introduction.** Digital medical images have gained increasing importance in the identification of various ailments, playing a vital role in education and research. The use of electronic medical photos has become essential, exemplified by a study conducted by the Department of Radiology at the University Hospital of Geneva in 2002, where 12,000 to 15,000 images were analyzed daily[1]. Medical report writing and image analysis necessitate an accurate and effective computer-aided diagnostic system. The traditional method of physically assessing medical imaging is laborious, imprecise, and prone to mistakes which lead brain tumors into a severe problem throughout the years, coming in at number 10 among the leading causes of mortality worldwide. According to reports, in 700,000 people, Brain tumors are a medical condition characterized by abnormal growths within the brain. Approximately 80% of these tumors are harmless, while the remaining 20% are malignant. [2].

As seen in Figure 1.1, the brain tumor is the most commonly observed kind of brain disease characterized by the uncontrolled development of brain cells. There are two distinct forms of brain cancer, namely primary and secondary brain tumors. Primary brain tumors originate within the brain and typically remain localized in that region. On the other hand, secondary brain tumors form as cancerous cells elsewhere in the body and then spread to the brain [3]. There are two distinct types of tumors: malignant and benign.

In contrast, a malignant tumor is characterized by its highly aggressive nature, capable of metastasizing to distant sites. In contrast, a benign tumor has a relatively sluggish growth pattern and cannot invade adjacent organs. The World Health Organisation (WHO) classifies brain tumors into four grades, from grades I to IV. Tumors classified as groups III and IV typically exhibit malignancy and are associated with a less favorable prognosis, whereas malignancies falling under categories I and II are generally characterized by a slower growth rate. [4].

The development of numerous imaging methods over the past few decades, such as "X-ray, Magneto Encephalo Graphy (MEG), Computed Tomography (CT), ultrasound, Electronic Ephalo Graphy (EEG), single-photon emission computed tomography (SPECT), positron emission tomography (PET), and magnetic resonance imaging (MRI)," has allowed for the precise diagnosis of brain tumors and the selection of the most

---

\*Research Scholar, Department of ECE, KLEF (Deemed to be University), Vaddeswaram, AP, India ([swarna26258@gmail.com](mailto:swarna26258@gmail.com)),

†Professor & HoD, Department of ECE, KL University, KLEF (Deemed to be University), Vaddeswaram, AP, India ([saratkumarphd4@gmail.com](mailto:saratkumarphd4@gmail.com)).

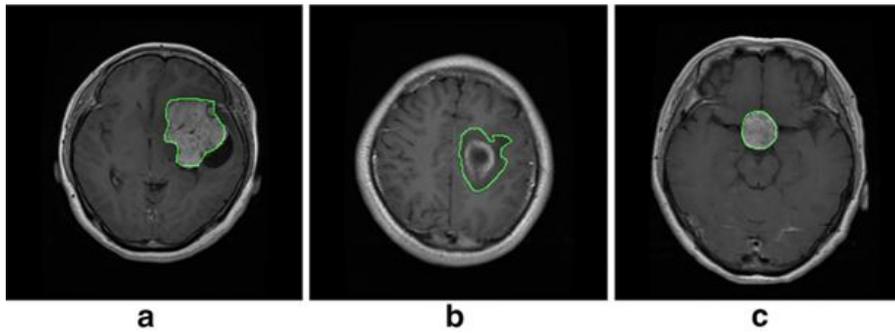


Fig. 1.1: Tumor types a) Meningioma b) Glioma c) Pituitary [1]

effective treatment options. Magnetic resonance imaging (MRI) is among the most regularly utilized imaging techniques for detecting brain tumors [5]. Brain tumor size, location, and form can all be accurately determined with MRI's soft tissue contrast imaging since patients are not subjected to unnecessary ionizing radiation.

It takes a longer time and a lot of skill and knowledge on the part of the radiologist to diagnose a brain tumor. As a result of the rise, Patients now have an unprecedented amount of data that must be analyzed, rendering traditional techniques ineffective, both costly and erroneous [6]. The difficulties are caused by significant variations in the size, shape, and seriousness of an identical kind of brain tumor and by the similarity of many other disease types' presentations. A brain tumor's incorrect classification can have severe repercussions and lower the patient's survival chance. Building automated image processing technologies is becoming increasingly popular to get around the drawbacks of manual diagnosis and its potential uses [7,8]. Several CAD [9] systems have been created recently to detect brain tumors automatically.

Researchers looked at various techniques for swiftly and reliably identifying and classifying brain tumors. Building automated The use of Deep Learning (DL) techniques to rapidly develop systems for correctly classifying brain tumors has become commonplace. In the case of brain tumor classification, DL enables the use of a pre-trained "Convolutional Neural Network (CNN)" architecture [10], such as GoogLeNet [11], "AlexNet," and ResNet-34, which has been designed for a wide range of applications. DL [12] is a backpropagation-based, multi-layered deep NN that minimizes the difference between the desired and observed values. However, as the number of layers in an artificial neural network grows, the complexity of the model development process also increases.

The subsequent sections of the paper are structured in the following manner: The methodologies employed for detecting brain tumors are discussed in Section 2, published research in this field. The proposed method, including the proposed model and algorithm, is fully described in Section 3. The outcomes of the experiments are displayed in Section 4. The description of the approach and generated effects are included. The study effort is concluded in Section 5 and Section 6.

## 2. Related Work. Some of the notable research published in this area include:

This work endeavors to employ machine learning techniques and feature selection methodologies. In the dataset, images from several categories exhibit inconsistencies in detecting diseases in their early stages. These images need to undergo further pre-processing and segmentation to enhance the effectiveness of feature extraction. This system combines DL techniques and "image processing" technologies to identify potential disorders ("X-ray, MRI, and CT scan images"). CNN is often utilized to divide brain tumors into normal and abnormal, with an accuracy rate of 94.81% [1].

The structure of CNN is employed to identify sparse depictions in the nonlinear space. The generated coding vectors of distinct classes can then be used to approximate the discrimination REMBRANDT dataset for the classification of meningiomas, gliomas, and pituitary tumor types to achieve optimum performance in (accuracy, precision, recall, F1-score, and balance loss) with an accuracy rate of 96.39%[2].

The suggested fully automated method is evaluated using MRI scans of the three most common types of

brain tumors from an open collection on Figshare. CNN uses Brain MRI images to extract various features, and for better performance, a multiclass SVM and CNN features are used in a fivefold cross-validation technique. 95.82% of collected classifications were accurate overall. When there is an absence of training data, It has been found that the SVM classifier works much better than the softmax classification for things like CNN. [3].

The overfitting and vanishing gradient issues are fixed Using ResNet-50 and global average pooling in a deep network system. The three-tumor brain magnetic resonance image dataset, consisting of 3064 images, was used to determine how well the model program worked. Key performance metrics were utilized to measure how well the proposed model and its competitors worked. With and without data addition, the mean accuracy was 97.08% and 97.48%, respectively. [4].

At first, a 3D CNN architecture is built to find different brain tumors. Then, tumors are located on top of a learned CNN model to see more tumors using a DL-based method for detecting microscopic brain tumors along with their tumor type classification. The multiple BraTS datasets from (2015, 2017, and 2018) are deployed to run experiments along with validation, and the obtained features are then incorporated into the correlation-based selection procedure, where they are finally classified after being confirmed by a feed-forward neural network. The accuracy of these datasets is 98.32, 96.97, and 92.67%, respectively [5].

By scrutinizing each pixel in the image, the suggested methodology seeks to recognize and categorize the various types of tumors. An alternative approach to enhance semantic segmentation performance involves utilizing the "patch-wise classification method". This study employs a composite system, utilizing deep multimodal convolutions based on the "U-NET" architecture. The convolutions are applied using convolutional neural networks (CNNs) to divide the input into three scale patches based on pixel-level analysis. The LSTM network integrates the three pathways to ascertain the categorizations of tumors. Using the MRI BRATS'15 dataset, a fivefold cross-validation approach is used to authenticate the suggested methodology. The experiment results demonstrate that the MSMCNN model outperforms CNN-based models with an accuracy of 96.36 over the Dice coefficient [6].

Using contrast calculations to analyze the pixel, The differential deep convolutional neural network (CNN) model can accurately and seamlessly categorize an extensive repository of images, owing to its ability to process visual sequences. A dataset consisting of 25,000 brain magnetic resonance imaging (MRI) images is employed to assess and refine the efficacy of this particular model. This dataset encompasses aberrant and normal photos, enabling comprehensive evaluation and training of the model's performance. The experimental findings demonstrated that the model suggested had a 97.33% accuracy rate [7].

According to a hypothesized attention mechanism, the type of tumor depicted in the images can be identified by increasing focus on tumor parts while lowering stress on non-tumor sections. Based on the benchmark datasets Figshare and BraTs2018, our strategy is much more efficient in terms of generalization and simplicity about the number of layers compared to the current advanced models that follow the fine-tuning of deep CNN models. Two sets of tumor image projections can be jointly trained using the advised two-channel architecture to obtain good generalization. Moreover, 97.8% of the proposed model was accurate [8].

Comparing the proposed architecture to the 2D CNN variation, quantitative assessments reveal that it produces the optimum The study conducted by Brats-2018 utilized an unsupervised feature map to distinguish between low-grade (LG) and high-grade (HG) gliomas. The approach described in this study demonstrates a high level of accuracy, achieving an overall accuracy of 96.49% when applied to the validation dataset. This performance surpasses previously created supervised and unsupervised state-of-the-art methods, as reported in reference. The experimental results demonstrate that accurate classification may result from appropriate MRI preprocessing and data augmentation when employing CNN-based techniques [9].

This study proposes a novel brain tumor classification system that utilizes convolutional neural networks (CNN) and is designed to accommodate many grades of tumors. The initial stage involves using a deep learning framework to distinguish tumor regions within an MR picture. Substantial data augmentation is implemented to train the proposed system effectively, hence mitigating the issue of insufficient data encountered in multi-grade brain tumor classification utilizing MRI. The utilization of a Convolutional Neural Network (CNN) model is ultimately being considered. That was previously trained to classify the grade of brain tumors is strengthened using new data. The developed model's accuracy rate is 94.58% when the system was experimentally tested using both augmented and original data [10].

There are two essential steps in the proposed methodology where the images are first subjected to several image processing algorithms before being subjected to CNN classification [11]. The study's 3064 image gallery includes three discrete types of brain tumors: glioma, meningioma, and pituitary tumors. The CNN approach that has been suggested enables us to achieve a testing accuracy of 94.39% [12].

As a result, manual brain tumor recognition is complicated, prolonged, and prone to error. Thus, a highly accurate, automated computer-assisted diagnostic is currently needed [13]. This study, which used the Figshare data set, introduced the principles of preprocessing and data. The augmentation techniques employed significantly improved, with the achieved level of "Intersection over Union (IoU)" reaching 95.04 for enhancing the classification rate [14]. It provides segmentation utilizing Unet architecture utilizing ResNet50 as a backbone. Brain tumors are divided into various categories using optimization approaches and reinforcement learning with transfer learning [15].

The Figshare Brain Image (FBID) dataset is an assortment of pictures of the human brain gathered from different sources. MRI scans, CT scans, as well as other imaging methods are included in the collection. More than 1,000 images of the brain in a variety of settings and circumstances are included [16]. The pictures are divided into groups like healthy, dysfunctional, and diseased brains. Each image in the dataset has annotations describing the sort of scan used to obtain it and any pertinent clinical data [17]. This dataset is helpful for scientists researching the anatomy and pathology of the brain and for doctors who need to identify and treat neurological illnesses [18]. A total of 233 patients diagnosed with three distinct types of brain tumors, namely meningiomas (708 slices), gliomas (1426 slices), and pituitary tumors, were included in the study [19]. The patients underwent 3064 T1-enhanced weighted contrast brain imaging scans, with 930 pieces dedicated explicitly to this imaging modality [20].

Datasets are a priceless resource for researchers using medical image analysis to examine brain tumors [21]. It provides an extensive selection of pictures that are captioned with crucial information regarding each patient's diagnosis and treatment plan [22]. This enables academics to develop refined machine-learning algorithms and models that improve the precision of clinical diagnosis and patient outcomes [23].

The lack of publicly accessible datasets is the main issue facing the field of MRI [24]. Although there are various datasets, mainly on the Internet, few images are tailored to our problem. Thus, to cover the data gap and to make the system transformative and noise invariant, we extensively enrich the data using a variety of factors and methodologies [25].

### 3. Proposed Methodology.

**3.1. Data Segmentation.** The concept of the suggested models involves partitioning data into smaller segments or sections to facilitate more convenient examination. The purpose of this action is to mitigate the intricacy of the data and reduce the training process of the model. We have collected all the datasets from the internet and extracted the images from four folders (bt\_set1, bt\_set2, bt\_set3, and bt\_set4) which comprised of .mat files are merged to generate image data and further converted to new\_dataset which contains (images, labels, marks, and borders). Then all the images are generated using the h5py python library is used to provide all the pictures with HDF5 data format used to create the dataset as (labels. pickle, training\_data.pickle) files.

**3.2. Data Preprocessing and Augmentation.** Since the suggested CNNs model requires all images to be the same size, we first transform all of the photos into 512x512 forms with a boundary and a mask as a component of the data preprocessing procedure. Then, all the images must be normalized to guarantee that they have all been kept with the same range of values to improve accuracy. In this research, we use the batch normalization technique to do this. Then, change every image from grayscale to RGB, which may result in more data to analyze but less noise in the final product.

Data augmentation is a method for making modified versions of already-existing The act of artificially increasing the size of a database by including irrelevant or misleading data. This is performed by applying random alterations to the existing data, such as ("rotation, scaling, cropping, flipping, and introducing noise"). Machine learning models can perform better using data augmentation by receiving additional training data and experiencing less overfitting. In our proposed approach, we have augmented the image in seven ways.

Data augmentation is a strategy that solves the overfitting issue by artificially expanding any dataset during the training phase. In addition to the data augmentation method outlined above, the following techniques are

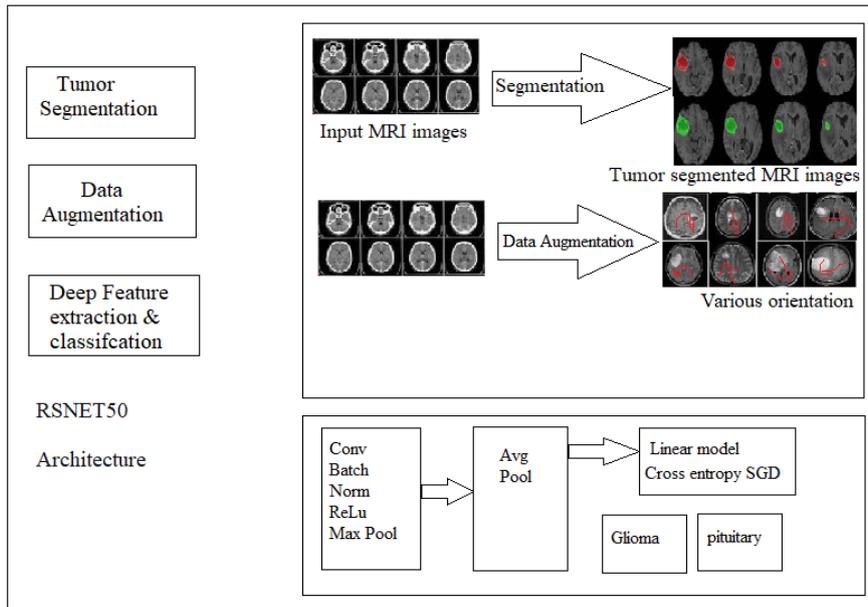


Fig. 3.1: Proposed optimized ResNet50 architecture

employed over every .mat file to strengthen our model:

- **PID:** Property Identifier and a unique identifier utilized by HDF5 datasets to guarantee that data is accurately described and fetched from the dataset. It has the shape of 6,1, and its type is u2. Where u2 is a user-defined identifier used to identify and access data stored, and it may be in the form of any string of characters or numbers.
- **Image:** in HDF5, datasets represent the type of file format used, which will be most often used to store large amounts of data which will be either in the form of raw data or compressed data. We have modified the images to a shape of 512,512 by using i2 as an image compression algorithm. The i2 technique is used to compress an image while maintaining its quality.
- **Label:** is a name given to a dataset with the shape of 1,1 to access all its features. We chose the f8 data format type to store less data, which divides image data into 8-bit integer values
- **TumorBorder:** is a feature in the HDF5 dataset that provides information on tumor borders of shape 1,38. This characteristic, which determines whether a pixel is inside the tumor boundary (1) or not (0), is represented as an 8-bit integer (f8)
- **TumorMask:** contains details about the position and size of tumors in any 512,512-shaped medical image. It aids in tumor analysis, and we utilized the ul(upper left) type, which provides the x-coordinate of the tumor mask's upper left corner.

**3.3. Model architecture.** The model we have proposed is an optimized RSNET50 architecture, which is the best one per the literature survey. To the architecture, the images are given as input for performing tumor segmentation through (labels. pickle, training\_data.pickle) files which are further augmented into seven different augmentations that are (The angles mentioned are 45°, 90°, 120°, 180°, 270°, 300°, and 330°) are grouped to generate a new batch using the utility function from the PyTorch library that allows you to chain together multiple transformations.

**3.4. Train test splitting.** In the process of assessing the efficacy of a machine learning model, it is imperative to partition the data into distinct subsets, namely the "training set," "validation set," and "test set." The test sets are utilized to quantify the model's ability to generalize, encompassing 15% of the available data. The validation set comprising 15% of the data is employed to assess the model's performance on previously

untested data. The remaining 70% of the data is allocated to the training set. This practice ensures that any alterations made to the model are grounded on impartial evidence and that any inferences from the evidence are trustworthy. The collected samples consist of 2144 training samples, 460 validation samples, and 460 testing samples. An augmented dataset was also created, resulting in 17152 training samples, 3680 validation samples, and 3680 testing samples.

We have established a Data Loader for each by enabling the shuffle feature. When conducting training, a batch size of 4 was utilized and validation and a batch size of 10 for testing. The design was then implemented using Cuda. The detailed architecture and its layer's functions were explained. This model can train efficiently and provide accurate weight files which is shown in Fig 3.1.

### 3.5. Optimized RESNET50 architecture for performing deep feature extraction and classification.

- conv1: This convolutional layer consists of (3,64,256,512,1024,2048), input channels, (64,128, 256,512) output channels. The convolutional layer is configured with a kernel size of (7, 7), a stride of (2, 2), and padding of (3, 3). The bias parameter is configured to False, indicating that the layer does not incorporate a bias vector. The mentioned layer executes a two-dimensional convolution operation on the input data utilizing the provided parameters. The output will be an array of size  $64 \times (\text{input\_height} - 7 + 6) / 2 \times (\text{input\_width} - 7 + 6) / 2$ .
- bn1: is a 2-dimensional Batch Normalization layer. By removing the batch mean & dividing by the batch standard deviation, the batch normalization layer normalizes the input from the layer below it. This layer's parameters include 64 input channels, a small value added to the denominator for numerical stability, and a scalar value used as momentum to compute the running mean and variance. Affine is set to True, indicating that this layer has learnable affine parameters, and track\_running\_stats denote that the layer keeps track of the running mean and variance, whereby it may be utilized for assessment during training.
- ReLU: activation function utilized in artificial neural networks is the corrected linear unit. It's a non-linear function that accepts an input and depends on whether the information is positive or negative. When the in-place option is set to True, the procedure is carried out in position, which prevents the input tensor from being replaced with a new tensor and instead modifies it directly. Both performance, as well as memory, can be improved.

$$f(x) = \max(0, x) \quad (3.1)$$

In eq.3.1 monotonic ReLU function evaluates either negative input, giving back 0. If the code gets a specific positive value, x, it returns that value. So, the range of the result is from 0 to infinity.

- MaxPool2d: is a type of pooling layer in CNN that cuts down on the number of dimensions of the input divided into a set of non-overlapping rectangles and then takes the maximum value from each rectangle. kernel\_size represents the pooling window size which will be either square or rectangular region, stride represents the number of pixels to move between each pooling window and two will reduce the size of the output by half, padding will add extra pixels around the edges of the input to ensure that all regions are included in the pooling operation, dilation determines how much space should be between each pooling window, ceil\_mode determines that the fractional values resulting from the pooling operation will not be rounded with the nearest integer value.
- Sequential Bottleneck (SB): is a type of ResNet architecture comprised of layers with progressively fewer filters to make the feature maps smaller, producing deeper networks with fewer parameters and faster training times. Due to a sequence of convolutional layers preceded by a bottleneck layer, the SB architecture features fewer filters on each layer. Once the necessary depth has been obtained, The data from the bottleneck layer is sent on to the next convolutional layer. Using fewer parameters than standard architectures, this form of architecture can be utilized to build intense networks with dependable accuracy.
- ownsampling: is a method for lowering the number of data points in a dataset by choosing only a portion of them. Typically, this is done to shrink the dataset and simplify processing. By lowering data noise, it can also be utilized to enhance model accuracy, and due to this step, there is no scope for overfitting or underfitting. Moreover, this procedure includes BatchNorm2d and conv2d.

Table 3.1: Proposed optimized RESNET50 algorithm layers

Layer	Bottleneck	Conv2d	BatchNorm2d	down-Sampling
0	64x64,64x64	64,64,256	64x256	256
(1), (2)	256x64,64x64	64,64,256	64x256	256
(1), (2), (3)	512x128	128,128,512	512x1024	1024
(1),(2),(3),(4),(5)	1024x256	256,256,1024	512x1024	1024
(0)	1024x512	512,512,2048	1024x2048	2048

- Identity Blocks: In ResNet architectures, identity blocks are a particular CNN layer. They consist of two or more convolutional layers that add the input of the block to its output. This enables the network to pick up identity functions, which lessens the vanishing gradient issue and enhances performance as a whole. In ResNet topologies, identity blocks are generally utilized to extend the depth of the network without raising its complexity.
- AdaptiveAvgPool2d: is a 2D adaptive average pooling layer that adapts its output size to meet the input size by considering the input size. It is used to shrink a 3D tensor's spatial dimensions while retaining most of its essential characteristics. This layer's output size is always (1, 1).
- Linear: is a neural network layer that is linear and is a fully linked layer that receives 2048 features as input and generates 2048, 2048,4 features as output also, we have connected bias parameter layer attached to a bias vector which is used to modify the layer's output.
- SELU: Scaled Exponential Linear Unit() is a kind of activation function utilized in ResNet in which the activation function is non-linear, which aids in enhancing deep learning models' accuracy. It is based on the neural network's self-normalizing characteristic and aids in vanishing gradient reduction, which occurs when the loss function gradients become very small, making it difficult for the network to learn further.
- Dropout: is a regularisation method utilized in deep learning to avoid overfitting. It arbitrarily alters a fraction of the input units from 0 to 0.4 to streamline the model and prevent overfitting. This method modifies the current tensor in place or returns a brand-new tensor with the investigation's results implemented to regulate the proportion of input units set to 0.

LogSigmoid: The output of this activation function, which ranges from 0 to 1, can be used to convey the probability that a given input belongs to a particular group. It uses the weighted sum of information and then undergoes a sigmoid transformation. Since it enables more precise predictions than other activation functions, it is frequently employed in classification problems. In eq.3.2 to 3.4,  $Wx$  represents the dimensions of the image, where  $W$  represents several parameters and  $b$  represents the hidden layers.

$$a = \sigma(Wx + b) \quad (3.2)$$

$$W^{[L]} : (n^{[L]}, n^{[L-1]}) \quad (3.3)$$

$$B^{[L]} = (n^{[L]}, 1) \quad (3.4)$$

The proposed optimized RSNET50 architecture comprises four layers is shown in figure 3.1, which is illustrated in Table 4.1.

- In eq.3.5, The number of parameters for the L layer is where L is the L layer, and  $n[L]$  is the number of units in the L layer:

$$param = n^{[L]} * n^{[L-1]} + n^{[L]} (LayerL) \quad (3.5)$$

- SGD optimizer: The resnet model's parameters were optimized using the stochastic gradient descent (SGD) optimizer. We have applied the momentum of 0.9 and the learning rate of 3e-4. The SGD

optimizer helps to gradually minimize the loss by updating the model's parameters using gradients calculated from the loss function.

Table 3.1 illustrates that Layer 1 has three conv2d and Three conv2d & BatchNorm2d layers, each with 3 Identity blocks, which make up Layer 2. 5 Identity blocks spread throughout Layer 3's three conv2d and BatchNorm2d layers make up Layer 3. Layer 4 has three conv2d and BatchNorm2d layers with two Identity blocks. And all the layers have downsampling performed for nonidentity block with single conv2d and batchNorm2d. And in all layers, bottleneck(0) is a nonidentity layer, and the rest of the bottlenecks are identity layers.

The layer's output is sent to the next one. avgpool and then further to the FullyConnected layer, which comprises ("Linear, SELU, LogSigmoid, Dropout") will produce the output and provide data for the Cross-Entropy Loss SGD Optimizer to determine whether the image belongs to one or more classes ("Meningioma, Glioma, Pituitary").

**3.6. ResNet 50 optimized Algorithm .** The following is the proposed optimized RESNET50 algorithm

**4. Algorithm: Optimized RESNET50. Input:** labels.pickle, training\_data.pickle

**Output:** Accuracy, multiclass classification results

**Step 1:**

- Set training start time.
- Initialize loss value checkpoint threshold.
- Empty batch variables.

**Step 2:**

- Start training based on epochs.
- Initialize training and testing counters.
- Set the epoch's starting time.
- Train the batches.

**Step 3:**

- Calculate the loss for each sample image using argmax of the predicted tensor.
- Optimize through backpropagation with loss.

**Step 4:**

- Calculate training metrics.
- Evaluate validation accuracy and loss.

**Step 5:**

- Evaluate test accuracy.
- Plot the confusion matrix.
- Generate a classification report.

**Step 6:**

- Perform multiclass classification.

The proposed algorithm begins by setting the training start time and initializing the loss value checkpoint threshold and batch variables (Step 1). The training process based on epochs is initiated in Step 2, with counters and timings appropriately set. Step 3 involves calculating the loss for each sample image and optimizing through backpropagation. Step 4 calculates training metrics and evaluates validation accuracy and loss. Step 5 assesses test accuracy, plots the confusion matrix, and generates a classification report. Finally, Step 6 performs multiclass classification.

The proposed algorithm initiates in step 1 by setting the training start time; then, we have to set the loss value checkpoint threshold as the initial step by initializing all the batch variables to default values. Then in step 2, the initialization of the training step is performed, which merely relies on epochs that are set to 30 means there exist 30 iterations. In step 3, the development of the proposed model will be modified through hyperparameters for evaluating loss after each epoch batch with Step 1: set training start time, loss value checkpoint threshold by emptying empty batch variables. Step 2: start training based on epochs by emptying training correctly and testing the correct counter during every iteration, then set epoch's starting time then train the batches.

Step 3: Calculate the loss for each of the sample images using argmax of predicted tensor and further performing optimization through backpropagate with loss

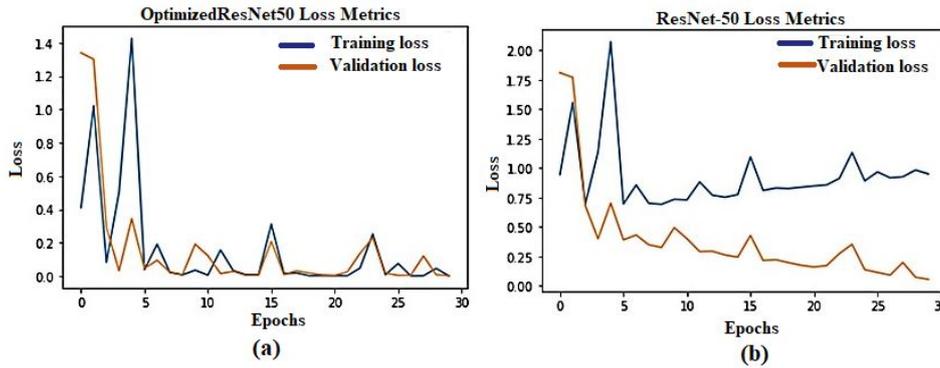


Fig. 5.1: Loss metrics of a) optimizedResNet50 and b) ResNet50

Step 4: Calculate training metrics, validation accuracy, and loss.

Step 5: Evaluate test accuracy and plot the confusion matrix and generate a classification report

Step 6: perform multiclass classification

The loss of the previous epoch. The torch. arg max () function can be used to determine the arg max of a predicted and accepted tensor as an input parameter, generating an index of the tensor's highest value as output through backpropagation. In step 4, the number of correctly identified instances in each batch is added to the overall number of correctly classified images, leading to the model's accuracy over time, and nonclassified photos are to be evaluated as loss via backpropagation. To reduce loss and raise model accuracy, we can use this to adjust the weights and biases. Each epoch batch's backpropagation calculation helps us ensure that our model continuously learns from errors and improves accuracy. In step 5, validation accuracy is evaluated, then accuracy metrics are plotted, and test loss is evaluated to identify the correlation coefficient between the last item and the total number of items in each epoch. In step 6, multi-class classification is performed by defining ground truth labels: "Meningioma, Glioma, Pituitary."

**4.1. Loss Function.** We'll use the cross-entropy loss [15] to measure how much the actual output differs from the predicted one., it serves as the Loss Function in ResNet 50. The model's performance is measured using this loss function, which is also used to quantify how much model adjustment is required to enhance performance [16].

**4.2. Jaccard Similarity score.** We'll use the cross-entropy loss [15] to measure how much the actual output differs from the predicted one., it serves as the Loss Function in ResNet 50. The model's performance is measured using this loss function, which is also used to quantify how much model adjustment is required to enhance performance [17].

**5. Results.** We have implemented the proposed architecture as illustrated in Table 5.1 and as per the algorithm presented in section 3.3.3 by collecting the FBID dataset.

**5.1. After training the batches using cuda through forward pass image samples and after performing a training matrix for every epoch (accuracy, loss, validation accuracy, validation loss) generated in Table 5.1 which classifies the difference between OptimizedResNet50 and ResNet50 [18]..** In OptimizedResNet50 epoch 27,28 and 30 accuracy has exceeded 100, hence we must not consider these results for the study but we are considering the validation accuracy as the measure. To acquire it we have implemented the downsampling and dropout functions hence we can consider these epochs.

We plotted the results generated through all the 30 epochs through which training and validation loss acquired is plotted as illustrated in Figure 5.1 and Figure 5.2. Training and validation accuracy is plotted where training accuracy is more than the precision of the validation. The training loss will measure the

Table 5.1: Loss and Validation Metrics for Optimized RESNET50 and ResNet50

Epoch	OptimizedResNet50 Loss	ResNet50 Loss	Validation Accuracy	Validation Loss
1	0.4112	1.3395	51.4892	0.9433
2	1.0226	1.3008	74.1297	1.5546
3	0.082	0.2911	79.5896	0.6950
4	0.5032	0.0314	82.1292	1.1363
5	1.4266	0.3442	83.8491	2.0707
6	0.0373	0.0449	85.0799	0.6933
7	0.1904	0.0942	86.5993	0.8545
8	0.0215	0.023	88.4996	0.6985
9	0.0065	0.008	89.2396	0.6905
10	0.0347	0.1907	90.2293	0.7328
11	0.005	0.1216	89.1095	0.7281
12	0.1557	0.0142	89.7797	0.8807
13	0.0323	0.028	90.4296	0.7683
14	0.0075	0.0036	90.6395	0.7506
15	0.008	0.0067	90.6395	0.7731
16	0.3111	0.2079	90.5097	1.0931
17	0.0145	0.0079	90.5198	0.8085
18	0.0176	0.0313	90.9791	0.8287
19	0.0012	0.0181	90.6892	0.8243
20	0.0017	0.0065	91.0791	0.8358
21	0.0013	0.0034	91.0697	0.8463
22	0.0014	0.0242	91.3096	0.8554
23	0.0474	0.1353	90.7596	0.9104
24	0.2521	0.2287	91.0295	1.1302
25	0.0075	0.0178	90.8292	0.8896
26	0.0745	0.0034	90.9398	0.9655
27	0.0002	0.0045	90.8992	0.9153
28	0.0014	0.1201	90.7798	0.9244
29	0.0453	0.0074	90.5093	0.9814
30	0.0006	0.0018	90.5995	0.9487

performance of the suggested model. the performance through which model weights and biases are adjusted in the training process, and validation loss evaluates the ability of model generalization shown in table 5.1.

Very minimum loss is achieved for the proposed optimized resnet50 algorithm, which is 0.0006, and validation loss of 0.0018. For the case where training loss exceeds validation loss, it indicates overfitting, which means that the model is underfitting, but it's not as the model is capturing all the underlying patterns on training data and need not require adding some more layers for the architecture. Specifically, we find that the training loss & validation loss are quite near to one another, demonstrating the superiority of our suggested model. It is performing well.

**5.2. Confusion matrix.** Every evolution matrix will evaluate the performance of the model proposed through the confusion matrix by evaluating (accuracy, precision, recall, f1-score, and support) [19].

- Accuracy: In a confusion matrix, accuracy is the proportion of a model's accurate predictions to all of its predictions as illustrated in eq.4.1.

$$Accuracy = \frac{TruePositives + TrueNegatives}{sum(TruePositives, FalsePositives)} \quad (5.1)$$

- Precision: Precision is a metric for how well a model predicts successful outcomes. It's determined by

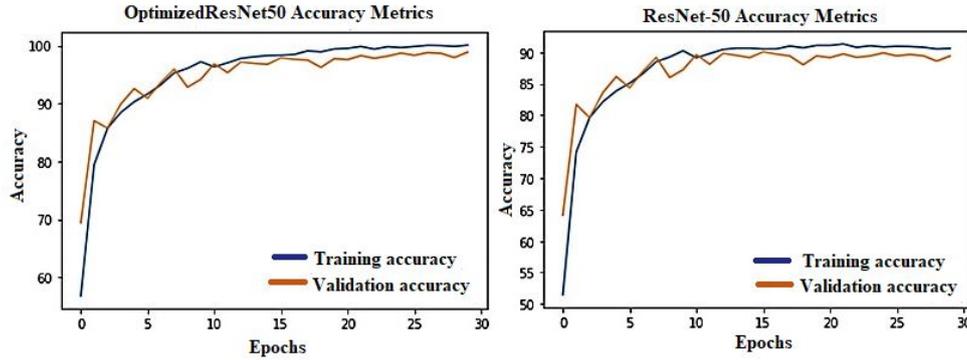


Fig. 5.2: Accuracy metrics of a) OptimizedResNet50 and b) ResNet-50

subtracting the number of true positives from the total number of expected positives eq.4.2.

$$Precision = \frac{numberofTruePositives}{sum(TruePositives, FalsePositives)} \quad (5.2)$$

- Recall: Recall, usually referred to as sensitivity, is a measure of how accurate a classifier is shown in table 5.2. It measures how effectively a classifier can locate all pertinent occurrences through eq.4.3.

$$Recall = \frac{numberofTruePositives}{sum(TruePositives, FalseNegatives)} \quad (5.3)$$

- F1 score: is indeed a metric used to assess a model's performance on a classification issue. It is determined by averaging precision and recall harmonically. It is determined in a confusion matrix by averaging precision and recall to every class.
- Support: The amount of samples of the real response that fall into a certain category or class constitutes the support in a confusion matrix.

$$n\_correct = TP_0 + TP_1 + \dots + TP_{N-1} \quad (5.4)$$

$$Accuracy = \frac{n\_correct}{n\_total} \quad (5.5)$$

$$Recall = \left( \frac{TP_0}{TP_0 + FN_0} + \frac{TP_1}{TP_1 + FN_1} + \dots + \frac{TP_{N-1}}{TP_{N-1} + FN_{N-1}} \right) \times \frac{1}{N} \quad (5.6)$$

$$Precision = \left( \frac{TP_0}{TP_0 + FP_0} + \frac{TP_1}{TP_1 + FP_1} + \dots + \frac{TP_{N-1}}{TP_{N-1} + FP_{N-1}} \right) \times \frac{1}{N} \quad (5.7)$$

$$F1 = \left( \frac{2P_0R_0}{P_0 + R_0} + \frac{2P_1R_1}{P_1 + R_1} + \dots + \frac{2P_{N-1}R_{N-1}}{P_{N-1} + R_{N-1}} \right) \times \frac{1}{N} \quad (5.8)$$

Eqn 4.4. represents  $n\_correct$  which evaluates the summation of true positives. Eq. 4.5  $n\_total$  denotes the total number of test set samples,  $N$  means the total number of defective types, in eqn 4.6,4.7,4.8 has  $TP$ , which represents correct answers,  $FN$  indicates incorrect answers, and  $FP$  indicates erroneous responses.

Table 5.2: Results of evolution matrix

Model/Par	Accuracy	Recall	F1-Score	Support
ResNet50	88.7	0.88	0.88	3241
optimizedResNet50	99.03	0.99	0.99	3680

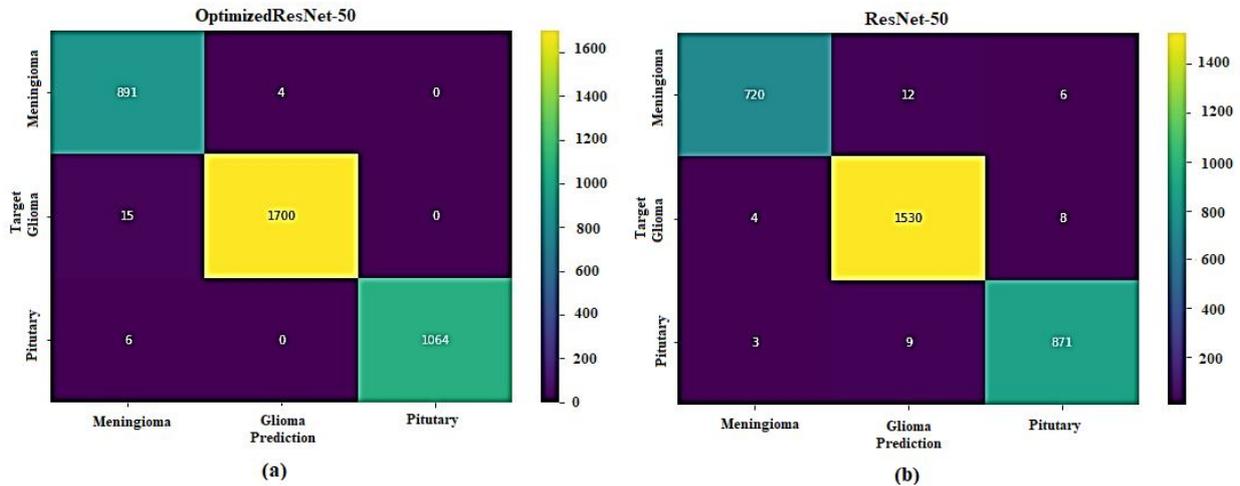


Fig. 5.3: Heatmaps generated to visualize confusion matrices of a) OptimizedResNet50 and b) ResNet-50

As per Table 4.1, we can identify that epochs (22, 24, 25, 26, 27, 28, 30) have acquired more than 98% of validation accuracy with validation loss of (0.0242, 0.2287, 0.0178, 0.0034, 0.0045, 0.1201, 0.0018). We can identify that epoch 30 has acquired maximum validation accuracy with minimum loss.

Table 5.2 denotes that the optimizedResNet50 has acquired more accuracy.

$$J(A, B) = \tag{5.9}$$

A metric called the Jaccard index or Jaccard similarity coefficient is illustrated in eq.4.9, where J denotes Jacarda distance, A and B are two distinct sets, it is used to assess how similar and diverse sample sets are. The magnitude of The ratio of the intersection to the cardinality of the union of the sample sets and it assesses consistency between finite sample sets. And the Jaccard Index obtained for optimizedResNet50 is: 99.03% where, as the ResNet50 is 88.5%.

Heatmaps were generated through Seaborn library [20] to visualize The utilization of a confusion matrix is a common practice in evaluating the performance of a model, as well as in visualizing the results to identify which classes are being misclassified and how well the model is performing overall and the total frequency of misclassifications is identified in figure 5.3 represents that the optimizedResNet50 has fewer misclassifications concerning ResNet50.

**6. Discussion.** We show that it is possible to The objective is to effectively categorize the brain tumor image into many classifications, including "Meningioma, Glioma, Pituitary." [21]. The challenge is identifying the tumor's location more precisely and classifying it so that miss classifications can be reduced [22]. We have modified the RESNET50 CNN model by imposing dropouts and downsampling at each layer to avoid overfitting; we have used four layers where each layer has a primary block and identity blocks through which we have implemented multiple hidden layers internally. By using SELU, we have increased the performance of neural networks, which consists of self-normalizing aspects by which the network can automatically adjust

Table 6.1: Comparison of Brain Tumor Detection Techniques

Reference	No. of Images	Dataset Source	Technique Used	Accuracy (%)
S. Solanki et al. [1]	1074	BraTS 2018	CNN	95.71
Gu et al. [2]	3064	REMBRANDT	CDLLC on CNN	97.49
Deepak et al. [3]	1426	Figshare	SVM with CNN	96.92
Kumar et al. [4]	3064	Figshare	RNGAP model on CNN	98.18
Rehman et al. [5]	1074	BraTS 2018	3DCNN	93.77
Rajasree et al. [6]	374	BraTS 2015	MSMCNN	97.46
Abd El Kader et al. [7]	3064	Figshare	HSANN	98.43
Bodapati et al. [8]	1074	BraTS 2018	ELM	96.9
Mzoughi et al. [9]	1074	BraTS 2018	3DCNN	97.59
Sajjad et al. [10]	121	Radiopaedia	Deep-CNN	96.68
S. Das et al. [11]	3064	Figshare	CNN	95.49
Sadad et al. [12]	3064	Figshare	ResNet50	96.14
Proposed Methodology	3064	Figshare	OptimizedResNet50	98.4

the parameters by which a stable network is obtained. Additionally, SELU can help minimize overfitting and lead to a regularization effect [23]. Further LogSigmoid activation function is used, which allows more complex relations between training images and testing images and can help minimize overfitting by introducing the nonlinearity aspect into the model [24]. For multiclass classification, we use cross-entropy loss, which is more robust to noisy labels. Moreover, its output is distributed over multiple classes while classifying the nonlinear relationships. And in order to minimize the local convergence, SGDM optimizer is utilized for enhancing the computational efficiency over large datasets to train the deep neural network [25].

In this way, we have implemented the proposed optimized Resnet50 architecture for performing multiclass classification to identify which type of tumor belongs to and which medical facilities can achieve proper treatment. Many models exist in the literature, but we have obtained improved accuracy and a promising strategy for identifying the tumor and performing multiclass classification [26-30].

Table 6.1 represents the % accuracy of the suggested optimisedResNet50 model was found to be 99.3%, which is much better than the existing ones in the literature.

**7. Conclusion.** One of the most complex difficulties to solve was categorizing brain malignancies. Magnetic resonance imaging (MRI) scans are being employed to examine and classify glioma, meningioma, and pituitary tumors. To prevent the issue of overfitting, we employed a 4-layer model with dropouts and down-sampling at each layer. Each layer had primary and identity blocks, which we used to create internal hidden layers. By utilizing SELU, we have improved the performance of neural networks. These networks include self-normalizing features that enable them to modify the parameters necessary to produce stable networks autonomously. To do multiclass classification and help medical facilities determine which type of tumor a patient has, we have built the suggested optimized Resnet50 architecture. Although numerous models exist in the literature, we have improved accuracy and an approach that shows promise for locating tumors and conducting multiclass classification with a 99.03% accuracy rate. The proposed method concentrated the performance efficiency compared to earlier models.

As a part of our future work, we will review the overall performance of our differential 2D ResNet50 model by increasing the network coverage by tweaking the differential filter's parameters. We will enhance deep network architectures by using a multi-channel classifier that improves classification performance more significantly than before.

## REFERENCES

- [1] GU, X., SHEN, Z., XUE, J., FAN, Y., & NI, T. (2021) , *Brain tumor MR image classification using convolutional dictionary learning with local constraint*. *Frontiers in Neuroscience*, 15, 679847.

- [2] DEEPAK, S., & AMEER, P. M. (2021), *Automated categorization of brain tumor from mri using cnn features and svm*. Journal of Ambient Intelligence and Humanized Computing, 12, 8357-8369.
- [3] KUMAR, R. L., KAKARLA, J., ISUNURI, B. V., & SINGH, M. (2021), *Multi-class brain tumor classification using residual network and global average pooling*. Multimedia Tools and Applications, 80, 13429-13438.
- [4] REHMAN, A., KHAN, M. A., SABA, T., MEHMOOD, Z., TARIQ, U., & AYESHA, N. (2021), *Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture*. iMicroscopy Research and Technique, 84(1), 133-149.
- [5] RAJASREE, R., COLUMBUS, C. C., & SHILAJA, C. (2021), *Multiscale-based multimodal image classification of brain tumor using deep learning method*. Neural Computing and Applications, 33, 5543-5553.
- [6] ABD EL KADER, I., XU, G., SHUAI, Z., SAMINU, S., JAVOID, I., & SALIM AHMAD, I. (2021), *Differential deep convolutional neural network model for brain tumor classification*. Brain Sciences, 11(3), 352.
- [7] BODAPATI, J. D., SHAIK, N. S., NARALASETTI, V., & MUNDUKUR, N. B. (2021), *Joint training of two-channel deep neural network for brain tumor classification*. Signal, Image and Video Processing, 15(4), 753-760.
- [8] MZOUGH, H., NJEH, I., WALI, A., SLIMA, M. B., BENHAMIDA, A., MHIRI, C., & MAHFUDHE, K. B. (2020), *Deep multi-scale 3D convolutional neural network (CNN) for MRI gliomas brain tumor classification*. Journal of Digital Imaging, 33, 903-915.
- [9] SAJJAD, M., KHAN, S., MUHAMMAD, K., WU, W., ULLAH, A., & BAIK, S. W. (2019), *Multi-grade brain tumor classification using deep CNN with extensive data augmentation*. Journal of computational science, 30, 174-182.
- [10] DAS, S., ARANYA, O. R. R., & LABIBA, N. N. (2019, MAY), *Brain tumor classification using convolutional neural network*. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5). IEEE.
- [11] SADAD, T., REHMAN, A., MUNIR, A., SABA, T., TARIQ, U., AYESHA, N., & ABBASI, R. (2021), *Brain tumor detection and multi-classification using advanced deep learning techniques*. Microscopy Research and Technique, 84(6), 1296-1308.
- [12] SADAD, T., REHMAN, A., MUNIR, A., SABA, T., TARIQ, U., AYESHA, N., & ABBASI, R. (2021), *Brain tumor detection and multi-classification using advanced deep learning techniques*. Microscopy Research and Technique, 84(6), 1296-1308.
- [13] SOLANKI, S., SINGH, U. P., CHOUHAN, S. S., & JAIN, S. (2023), *Brain Tumor Detection and Classification using Intelligence Techniques: An Overview*. IEEE Access.
- [14] LAKSHMI, M. J., & NAGARAJA RAO, S. (2022), *Brain tumor magnetic resonance image classification: a deep learning approach*. Soft Computing, 26(13), 6245-6253.
- [15] JUN, W., & LIYUAN, Z. (2022), *Brain Tumor Classification Based on Attention Guided Deep Learning Model*. International Journal of Computational Intelligence Systems, 15(1), 35.
- [16] HS, S. K., & KARIBASAPPA, K. (2022), *An effective hybrid deep learning with adaptive search and rescue for brain tumor detection*. Multimedia Tools and Applications, 81(13), 17669-17701.
- [17] LAKSHMI, M. J., & NAGARAJA RAO, S. (2022), *Brain tumor magnetic resonance image classification: a deep learning approach*. Soft Computing, 26(13), 6245-6253.
- [18] ASTHANA, P., HANMANDLU, M., & VASHISTH, S. (2022), *The proposition of Possibilistic sigmoid features and the Shannon-Hanman transform classifier along with the pervasive learning model for the classification of brain tumor using MRI*. Multimedia Tools and Applications, 81(17), 23913-23939.
- [19] EUNICE, JENNIFER AND POPESCU, DANIELA ELENA AND CHOWDARY, M KALPANA AND HEMANTH, JUDE (2022), *Deep learning-based leaf disease detection in crops using images for agricultural applications*. Agronomy, 10,12, 2395, MDPI.
- [20] LUNDERVOLD, A. S., & LUNDERVOLD, A. (2019), *An overview of deep learning in medical imaging focusing on MRI*. Zeitschrift für Medizinische Physik, 29(2), 102-127.
- [21] BONTE, S., GOETHALS, I., & VAN HOLEN, R. (2018), *Machine learning based brain tumour segmentation on limited data using local texture and abnormality*. Computers in biology and medicine, 98, 39-47.
- [22] ISLAM, A. T., APU, S. M., SARKER, S., SHUVO, S. A., HASAN, I. M., ALAM, A., & DIPTO, S. M. (2022, DECEMBER), *An Efficient deep learning approach to detect Brain Tumor using MRI images*. In 2022 25th International Conference on Computer and Information Technology (ICCIT) (pp. 143-147). IEEE.
- [23] MURTHY, A AND RANI, PARITALA JHANSI AND ALMAKASSEES, SARAH MAJEED AND SAIKUMAR, K AND SALEH, MOHAMMED AND ETTYEM, SAJJAD ALI (2023), *A novel classification model for high accuracy detection of Indian currency using image feature extraction process*. AIP Conference Proceedings,2845,1,050028.
- [24] SRINIVASARAO, GAJULA AND RAJESH, VULLANKI AND SAIKUMAR, KAYAM AND BAZA, MOHAMED AND SRIVASTAVA, GAUTAM AND ALSABAAN, MAAZEN (2023), *Cloud-Based LeNet-5 CNN for MRI Brain Tumor Diagnosis and Recognition*. Traitement du Signal, 40,4,pp. 1581–1592.
- [25] SAJJAD, M., KHAN, S., MUHAMMAD, K., WU, W., ULLAH, A., & BAIK, S. W. (2019), *Multi-grade brain tumor classification using deep CNN with extensive data augmentation*. Journal of computational science, 30, 174-182.

*Edited by:* Polinpapilinho Katina

*Special issue on:* Scalable Dew Computing for Future Generation IoT Systems

*Received:* Oct 17, 2023

*Accepted:* Feb 1, 2024