Journal of Computer Science

Original Research Paper

Deep Learning Approaches for Brain Tumor Diagnosis using Fused Layer Accelerator

[1]Mitrabinda Khuntia, [2]Prabhat Kumar Sahu and [1]Swagatika Devi

[1]*Department of Computer Science and Engineering, Institute of Technical Education and Research, India*
[2]*Department of Computer Science and Information Technology, Institute of Technical Education and Research, India*

Article history
Received: 22-10-2022
Revised: 16-12-2022
Accepted: 27-12-2022

Corresponding Author:
Mitrabinda Khuntia
Department of Computer Science and Engineering,
Institute of Technical Education and Research, India
Email: mitrabindakhuntia@soa.ac.in

Abstract: Deep Convolutional Neural Networks (DCNNs) are an emerging field in biomedical processing. Tumor classification is a key stage in the pathology analysis process, and deep learning algorithms for brain tumor categorization have recently shown promising results. However, these approaches often require more storage and more expensive training procedures to input a large number of training variables. To address this issue, light-weight deep learning models should be investigated without reducing classification accuracy. The aim of this study was to compare the classification rate of three pre-trained Transfer Learning classifiers, namely InceptionResNetV2, EfficientNetV1, and MobileNetV2, in categorizing brain tumors into four classes such as glioma tumors, meningioma tumors, pituitary tumors, and normal patients. In this article, attention modules based on pretrained deep learning models such as MobileNetV2, EfficientNetV1, and InceptionResNetV2 were highlighted. Following the fully connected layers and the ReLU6 layer, attention and convolution modules were integrated to obtain high-level object-based and critical semantic information. The effectiveness of this strategy was demonstrated by building a CNN accelerator based on the fusion of the top five convolutional layers of MobileNetV2, EfficientNetV1, and InceptionResNetV2 networks and comparing it to a Python accelerator. The EfficientNetV1 model showed the best results compared to the InceptionResNetV2 model and MobileNetV2 model.

Keywords: MobileNetV2, EfficientNetV1, InceptionResNetV2, Fused-Layer Accelerator, Attention Layer

Introduction

Deep convolutional neural networks acquire changed recognition computer vision precision. Massive advancements in processing underpin CNN accuracy gains. The accuracy of recognition improves with each successive network constructed, as does the amount of calculations necessary to evaluate the network. Even though the CNN algorithm is conceptually transparent, the absolute complexity of operations precludes even a unique layer of dataflow implementation. Every convolution layer necessitates the utilization of feasible computation units iteratively. Automated brain tumor classification has been researched using both traditional computer vision-based approaches as well as deep learning-based methods. Traditional machine learning methods are often accustomed to extracting low-level features before performing image classification using computer vision-based approaches, whereas deep learning-based methods derive features efficiently. Image processing along with traditional computer vision algorithms has employed imaging properties such as texture, color, and form as inputs for brain tumor categorization.

By the means of the emergence of deep learning based approaches in numerous fields, researchers have been focusing on creating deep learning-based solutions for the brain tumor classification challenge, particularly utilizing the transfer learning methodology. A Convolutional Neural Network (CNN) containing four convolutional layers, a max-pooling layer, a fully connected layer, and a ReLU layer was used to detect brain tumors. Existing DL-based techniques, however, have two major shortcomings. First of all, despite their claims of lightweight designs, such approaches often crave a significant amount of trainable variables. Second, the

Science Publications

quantity of classes and datasets available affects the performance of these models, hence we provide a unique lightweight deep-learning model based on the MobileNetV2 dataset (Sandler *et al*., 2018). Although our proposed model incorporates two compelling ideas i.e., convolution together with attention, we presume that the integration of these two principles for lightweight design is, to our knowledge, the first research in brain tumor classifications. The convolution along with attention modules are supposed to work collectively to increase performance in brain tumor categorization. The convolution module gathers the image's tangled features, while the attention module captures the image's conspicuous parts. We conducted tests on three distinct publicly available datasets to assess the efficacy of our suggested technique and the results demonstrate that our method provides reliable performance.

To prevent overfitting with DSC, (Pereira *et al*., 2016) employed small 3 × 3 kernels for convolution, with 78% total tumor, 65% core tumor, and 75% boosting regions. One of Arunachalam's significant breakthroughs is a revolutionary segmentation method that combines SIST with NSCT transformation to convert an image into a multi-resolution image (Arunachalam and Royappan Savarimuthu, 2017). The extraction of common features happens. The proposed approach has an accuracy of 99.8%. According to Havaei *et al.* (2017), the two path CNN (focused on local together with global routes) yielded a DSC of 0.85 for the entire segmentation, 0.78 for core, and 0.73 for augmenting. According to AlBadawy *et al.* (2018), training data from various universities might give significantly diverse outcomes. Hasan and Linte (2018) contended that a typical U-net 'deconvolves' rather than convolved the voxels. The deconvolution layer is replaced in this study with an upsampling layer that goes through two convolution layers, an up-sampling layer, and elastic transformation augmentation, where DSC is raised from 0.86 to 0.87. Chang *et al*. (2019) claimed that skull stripping outperformed other methods described in the literature. FLAIR hyperintensities were identified in a multi-institutional setting, both preoperatively as well as postoperatively, with a DSC for the FLAIR volume of 0.917.

Materials and Methods

Deep Learning (DL) approaches arose in tandem with the progress of artificial neural networks, whatever are activated by the human brain. The DL-based approach is essentially a larger neural network with more layers, nodes, as well as activation functions (Vickers, 2017). Furthermore, (Xiang *et al*., 2019) suggested a lightweight model contingent on a transfer-learning technique utilizing MobileNetV2. Their approach was created with devices with minimal power and limited resources in mind. On top of the core MobileNetV2 design, they added a convolution and dropout layer. Deep learning approaches based on transformers, which are commonly used in NLP, have recently been examined for computer vision applications just as image classification (Dosovitskiy *et al*., 2020). Furthermore, a few studies which used transformers for image classification discovered that when there was adequate training data, transformers outperformed CNN (Dosovitskiy *et al*., 2020). Furthermore, unlike vision transformers, the opportunity of pre-trained models for multiple CNNs makes image categorization applications more accessible.

A. MobileNetV2

53-layer-deep TL image recognition classifier MobileNetV2 (Sandler *et al*., 2018) was developed. The MobileNetV2 picture input size is 224 × 224. The MobileNetV2 model is appropriate for real-time as well as mobile applications since it is more computationally efficient. The model's great speed is attributable to the application of pointwise and depth-wise convolution methods. To connect bottleneck levels, the network uses unused connections. In the MobileNetV2 network, as shown in Fig. 1, a convolutional layer (with 32 filters) is followed by 19 residual bottleneck layers, dropout, dense, as well as softmax activation functions.

B. EfficientNetV1

EfficientNet is a design and scaling strategy for convolutional neural networks that employ a compound coefficient to consistently scale all depth/width/resolution dimensions. It attempts to design a sensible heuristic for scaling a CNN by linking the resolution, breadth, and depth of a CNN. The foundational EfficientNetB0 network is built on the inverted bottleneck residual blocks of MobileNetV2, along with squeeze-and-excite blocks. EfficientNet uses a compound coefficient to scale network width, depth, and resolution evenly. Figure 2 shows the architecture of the EfficientNetV1 model. Depthwise convolutions contain fewer parameters and FLOPs than normal convolutions. MBConv layers with depthwise convolutions were used in the EfficientNetV1 design. EfficientNet employs 7 MBConv blocks, with the following specs for each of those blocks:

- Kernel size is the kernel size for convolution
- Num repeat defines the number of times a specific block must be duplicated i.e., larger than 0
- Input filters and output filters are the amounts of itemized filters
- Expand ratio is the ratio of input filter expansion
- Id skip indicates whether or not to use the skip connection
- Se ratio specifies the squeezing ratio for the squeeze and excitation blocks

C. InceptionResNetV2

Inception ResNet-V2 is a Convolutional Neural Network trained over numerous images from the ImageNet dataset. Residual connections are used for inception blocks, to avoid vanishing gradients. It employs three distinct types of inception ResNet blocks, as shown in Fig. 3. The Beginnings of ResNet-A block is made up of three routes including convolutions of 1 × 1 and 3 × 3. There are two paths in the Inception ResNet-B as well as Inception ResNet-C blocks. It is a deep network composed of 1 stem block, 5 Inception ResNet-A, 10 Inception ResNet-B, 5 Inception ResNet-C, 1 Reduction-A, and 1 Reduction-B block. InceptionResNetV2 boosts performance by replacing the original Inception with cheaper Inception blocks and adding a filter expansion layer after each Inception block that employs 1 × 1 convolution without activation. To enhance the number of inception blocks, Batch Normalisation (BN) is only used on top of conventional layers. This network examines a 299 × 299 pixel image as input. To get high-performance outcomes at smaller epochs, the two deep learning models, Inception and ResNet, were integrated.

Proposed Model

The six components of our proposed approach are preprocessing, the convolution module, the attention module, the fusion of convolution as well as attention modules and fully-connected layers. Figure 4 depicts the entire flow of our strategy.

A. Preprocessing

The input images must be resized because our suggested solution is based on the MobileNetV2 (Sandler *et al*., 2018) framework. Using the same input dimension as the pre-trained MobileNetV2 model aids in the generation of highly discriminating features from photos. As a result, all input images are shrunk to 224 by 224 pixels within the range [-1, 1].

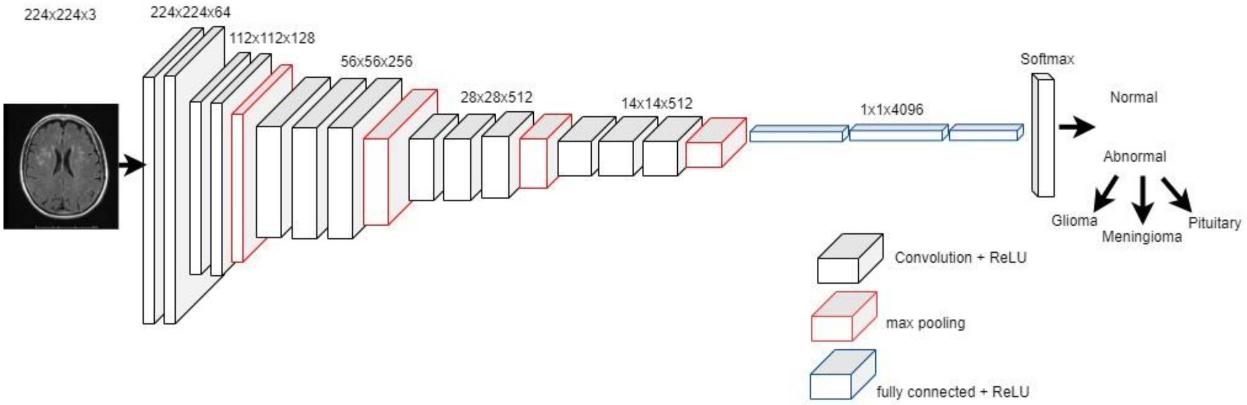


Fig. 1: MobileNetV2 model

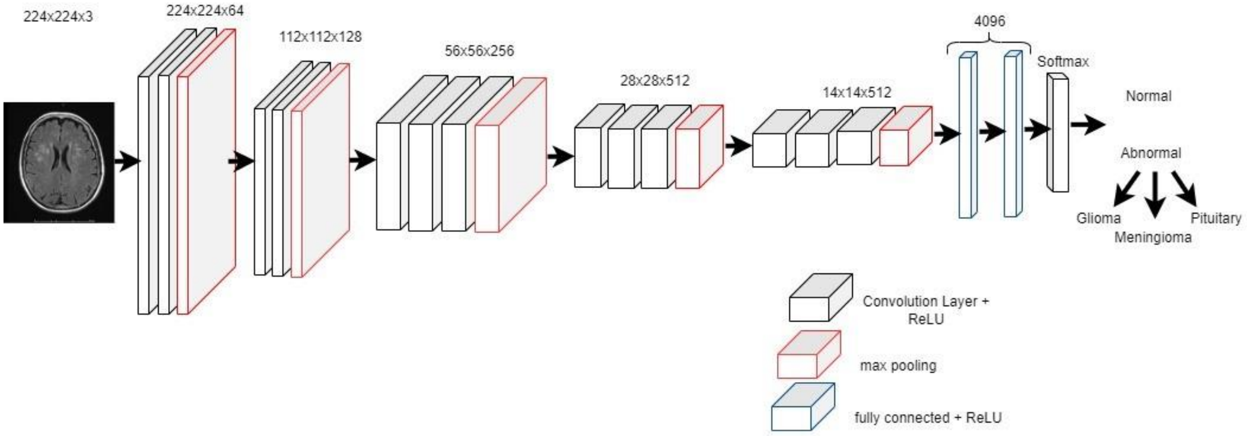


Fig. 2: EfficientNetV1 model

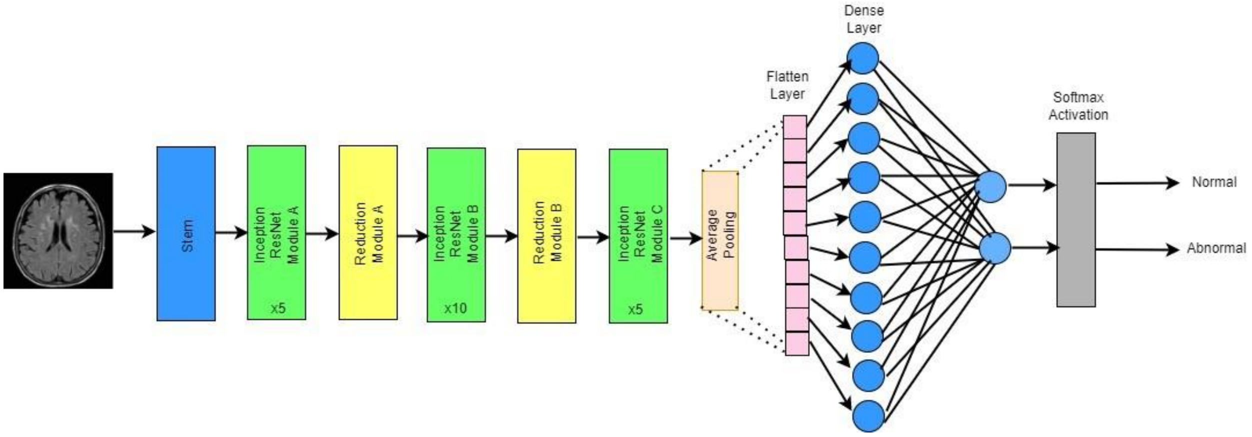


Fig. 3: InceptionResNetV2 model

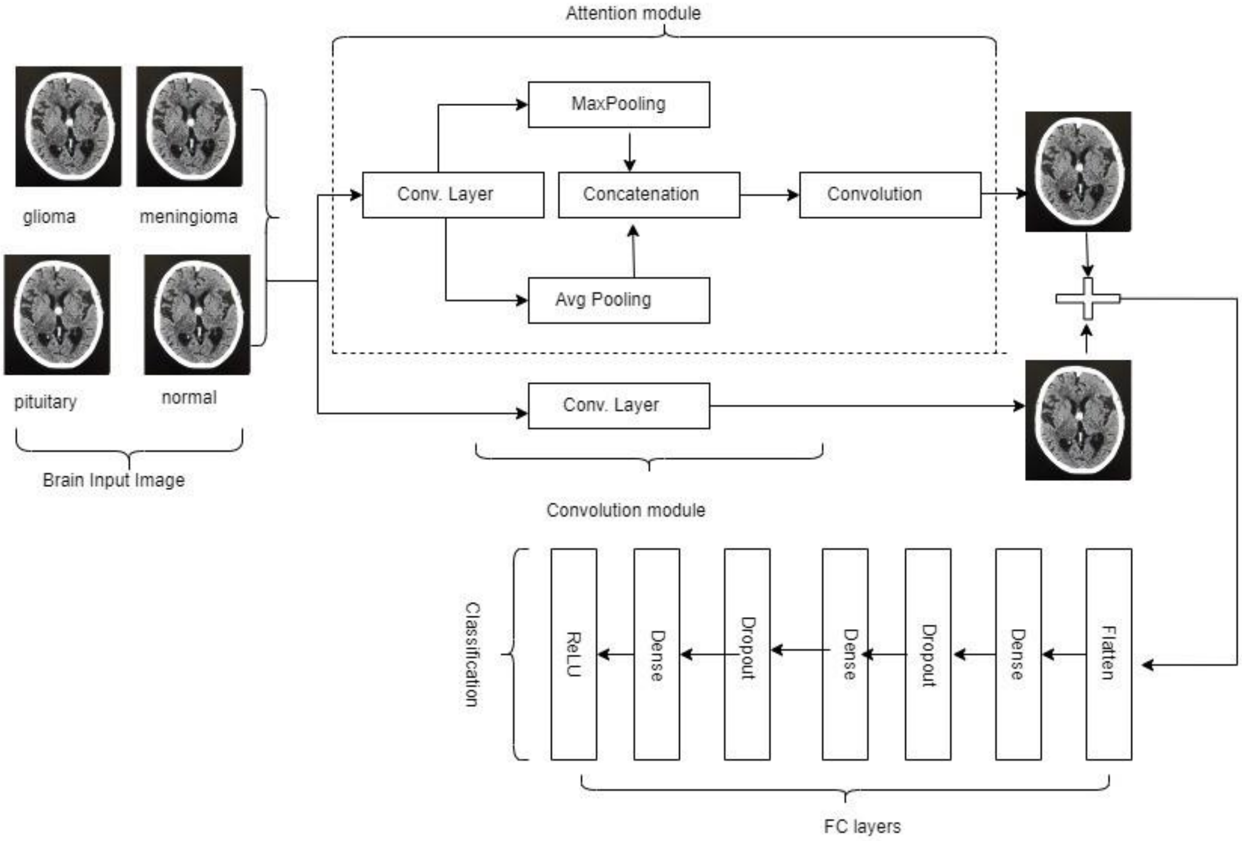


Fig. 4: Block diagram of our suggested brain tumor classification model

B. The Convolution Module

Convolutional Neural Networks (CNN) have enhanced image recognition accomplishment and have emerged as the dominant network topology in deep learning technologies (Hossain *et al*., 2018; Sitaula *et al*., 2021; Mishra and Shahi, 2021). Convolutional algorithms are critical in computer vision applications, but as the network structure deepens and gets larger, they become computationally costly. The MobileNetV1 (Howard *et al*., 2017) model introduces the depth-wise separable convolution concept, which separates the convolution into two sub-tasks: A depth-wise convolution that filters the input along with a point-wise convolution that combines the filtered values to generate new features. MobileNetV1’s core architecture consists of a typical (3 × 3) convolution layer followed by 13 depth-wise separable convolution blocks (Howard *et al*., 2017). The expansion layer, residual connections, as well as projection layers, as well as depth-wise convolution layers known as a bottleneck residual block, are now included in the MobileNetV2 (Sandler *et al*., 2018) model. The expansion convolution layer (1 × 1) adds channels based on the expansion factors, while the projection layer subtracts channels to produce a tensor of lower channels. The remaining connection in the network promotes gradient flow. There is a batch normalization and ReLU6 activation layer after each convolution layer, where ReLU6 is a ReLU activation function version with a maximum size of 6. The entire MobileNetV2 architecture is composed of 17 such bottleneck residual blocks, followed by a conventional (1 × 1) convolution, an average pooling layer, along with a classification layer (Sandler *et al*., 2018).

As a backbone network, we exploit the MobileNetV2 pre-trained using the 'ImageNet' dataset, where the convolution feature map is stemming from the final residual block, which is then followed by MobileNetV2's convolution layer. The smaller feature maps are produced by the lower-level residual blocks. These blocks are unhelpful to our inquiry since they do not collect high-level information for overall image identification. At the time of model construction and training, all layers are frozen up to the convolution layers, which generate a 7 × 7 3D feature map. In our investigation, this backbone network represents a convolutional feature extractor. Equation (1) shows it mathematically as follows:

$$F1(I) = Conv(I) \qquad (1)$$

where, $F1(I)$ is the convolution feature for the input image, I.

C. The Attention Module

The attention network is based on the human brain's attention mechanism, which states that while seeing visuals, individuals focus their attention on crucial cues rather than inspecting every element of the item. As more scholars worked on attention mechanisms, numerous attention models emerged (Woo *et al*., 2018). We are particularly interested in the attention mechanism which illustrates the spatial connectivity of visual information in brain tumor representations. The channel attention module seeks to catch the tumor region which is essential in the visuals provided because the spatial attention module seeks to capture the area or which portion of an image is more prominent. The structural attention technique is used in the development of our attention module to spotlight where there is an informative component in the brain tumor image. First, we apply max-pooling along with average pooling to the convolution module's input. Second, as recommended by Woo *et al*. (2018), the max-pooled as well as average-pooled tensors are concatenated. Ultimately, convolution with a filter size of (7 × 7) is employed in conjunction with the sigmoid activation function to activate visual hints in pictures. Equations (2) and (3) outline the total stages for the attention module (3):

$$F2(I) = \left[AvgPool(F); MaxPool(F)\right] \qquad (2)$$

where, $AvgPool(F) \epsilon R^{H \times W \times 1}$ illustrate the average pooling operation as well as $MaxPool(F) \epsilon R^{H \times W \times 1}$ describe max pooling operation. The concatenation of these two feature maps in 3D results in the feature map, $F2(I) = \epsilon R^{H \times W \times 1}$:

$$F3(I) = \sigma\left(f7\times7\left(F2(I)\right)\right) \qquad (3)$$

where, σ is a sigmoid activation function. f 7×7 describes the convolution operation of filter size (7 × 7), along with $F3(I)$ illustrating the attention features, which is H × W × C sized tensor with Height (H), Width (W), as well as depth (C).

D. Fusion of Convolution and Attention Module

Zhou *et al*. (2020), a multi-modality fusion network, can gather more detailed as well as effective information for several modalities than a single-encoder-based network. The purpose of fusion block is to emphasize the most important features from many modalities to point out the critical regions for brain tumor segmentation. While this method may lose some essential information in the latent representation, it is a straightforward technique to merge the independent latent representations.

To create a combined feature map, the feature maps obtained from the convolution as well as attention modules are merged by applying the basic concatenation feature fusion technique proposed by Afchar *et al*. (2018). We chose a basic concatenation fusion method over other approaches such as the max, min as well as sum since 2 feature maps contain distinct image attributes. It is also less computationally expensive than other ways that do the tensor product operation, such as the bilinear approach (Chu *et al*., 2021). The combination of these two characteristics yields a single feature tensor. The mathematical definition of the concatenated resultant feature tensor $T(I)$ for an image I is given in Eq. (4):

$$T(I) = \left[F1(I), F3(I)\right] \qquad (4)$$

where, $F1(I) \in R^{H \times Wx1}$ along with $F2(I) \in R^{H \times Wx1280}$ are 3D tensors including identical Width (W) as well as Height (H), it enables them to concatenate in 3D to generate 3D tensors $T(I) \in R^{H \times Wx1280}$.

E. The Fully Connected (FC) Layers

Following feature fusion, we employ multiple Fully Connected (FC) layers to turn a three-dimensional tensor into a 1-D feature vector. Figure 4, the FC layers include the pooling layer with the highest global average and the dropout layer. We set the density tier to 128 units, use the ReLU6 activation function, as well as set the dropout rate at 0.5.

F. Overview

The principal fused layer to the fusion technique is to take advantage of locality in the dataflow of a convolution. The output value of each convolutional layer is solely reliant on a tiny subset of the layer's inputs. We take use of this phenomenon by developing an assessment approach in which the first fused layer estimates its outputs in the order that the second fused layer demands them. This enables data to be transferred directly from one layer to the next without being moved off and back on the chip; the intermediate data is deleted once the subsequent layer consumes it. We begin with a single point in the output and work our way back to determine which region of the input feature maps it is dependent on. This approach generates a computation pyramid over numerous levels of feature maps when the layers are displayed spatially. It is unnecessary to load a whole fresh input pyramid base after completing a pyramid computation. Instead, each time the pyramid's base moves one space ahead, just one new column of the input space is loaded.

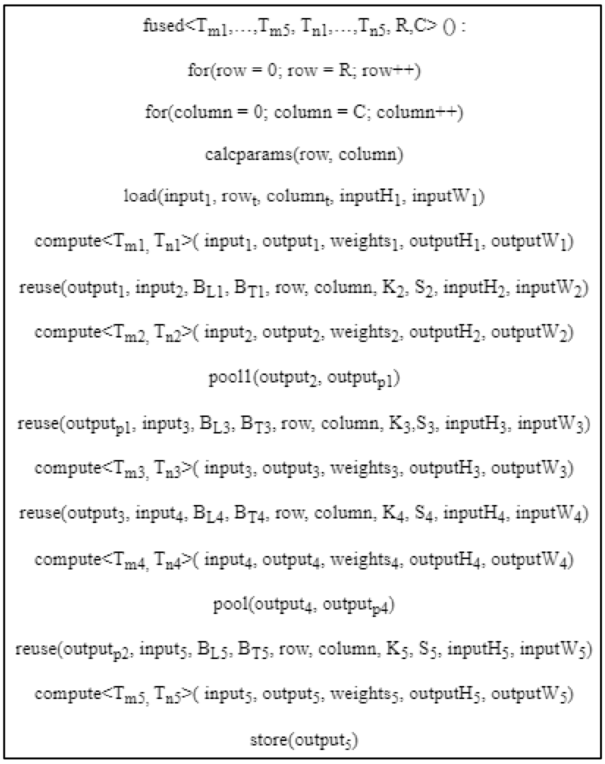
Some intermediate feature map values are required for both the black as well as red outputs. Both sets of output values are created using several intermediate values since their pyramids overlap. There are two options for dealing with this scenario. We may either compute the values every time they are required, or we can cache as well as repeat the in-between results while constructing the next pyramid. Although recalculating the integers requires more arithmetic operations, it has the benefit of simplicity because the core dataflow of each pyramid is the same. Caching the intermediate results removes this needless work, although it demands on-chip buffering moreover it makes each pyramid's processing random since a few pyramids require extra computing than others.

Accelerator Architecture

A Convolutional Neural Network (CNN) harvests features by the use of a set of convolutional layers, which are generally followed by one or more dense ("fully connected") neural network layers that do classification. Each layer gets N feature maps as input (N channels of R × C values) as well as convolute them with M sets of N × K × K filters (whose weights had earlier been calculated using a learning procedure as backpropagation). Convolution is performed for each of the *M* sets by sliding the filter with a stride of *S* across the input feature map (where *S* sites are moved by a filter at each step). At each point, the values of a filter are multiplied by the overlapping values of the input feature maps. In an output feature map, the resulting products are joined together to create a single value. This method is done for each of the *M* filter sets, each time yielding one output feature map. Convolutional networks include abundant layers, including the output of the previous layer acting as input feature maps for the following layer. Increased network intensity (layer count) improves recognition accuracy (Simonyan and Zisserman, 2014). As a result, the number of layers utilized in cutting-edge CNNs has increased significantly in recent years. A hardware prototype is used to evaluate fused-layer CNN accelerators.

A. Baseline CNN Accelerator

Loop modifications, like loop reordering, tiling, and unrolling, are used in the design to reorganize calculations and memory accesses, enhancing performance and minimizing data transmission (Zhang *et al*., 2015). For each computed invocation, this accelerator is constantly active, loading the input from off-chip memory and storing the output to off-chip memory. By pipelining the loop iterations and overlaying the compute operation of each tile with the load operation of the neighboring tile, the usage of double-buffering allows this design to execute at high efficiency.



```
fused<$T_{m1}$,…,$T_{m5}$, $T_{n1}$,…,$T_{n5}$, R,C> () :
for(row = 0; row = R; row++)
for(column = 0; column = C; column++)
calcparams(row, column)
load(input$_1$, row$_t$, column$_t$, inputH$_1$, inputW$_1$)
compute<$T_{m1}$, $T_{n1}$>( input$_1$, output$_1$, weights$_1$, outputH$_1$, outputW$_1$)
reuse(output$_1$, input$_2$, $B_{L1}$, $B_{T1}$, row, column, $K_2$, $S_2$, inputH$_2$, inputW$_2$)
compute<$T_{m2}$, $T_{n2}$>( input$_2$, output$_2$, weights$_2$, outputH$_2$, outputW$_2$)
pool1(output$_2$, output$_{p1}$)
reuse(output$_{p1}$, input$_3$, $B_{L3}$, $B_{T3}$, row, column, $K_3$,$S_3$, inputH$_3$, inputW$_3$)
compute<$T_{m3}$, $T_{n3}$>( input$_3$, output$_3$, weights$_3$, outputH$_3$, outputW$_3$)
reuse(output$_3$, input$_4$, $B_{L4}$, $B_{T4}$, row, column, $K_4$, $S_4$, inputH$_4$, inputW$_4$)
compute<$T_{m4}$, $T_{n4}$>( input$_4$, output$_4$, weights$_4$, outputH$_4$, outputW$_4$)
pool(output$_4$, output$_{p4}$)
reuse(output$_{p2}$, input$_5$, $B_{L5}$, $B_{T5}$, row, column, $K_5$, $S_5$, inputH$_5$, inputW$_5$)
compute<$T_{m5}$, $T_{n5}$>( input$_5$, output$_5$, weights$_5$, outputH$_5$, outputW$_5$)
store(output$_5$)
```

Fig. 5: Pseudo-code for fused layer accelerator

B. Fused Layer Implementation

The fused layer accelerator was built on top of the base CNN accelerator. The proposed fused-layer CNN accelerator computes for many layers, avoiding all off-chip data transmission of intermediate data along with writing the output feature maps only after all fused layers have been computed. Figure 5 shows the pseudo-code for this fused layer, which instantiates a distinct compute module for each layer being fused. The reuse model fused-layer accelerator and the baseline accelerator do the same amount of computation; our contribution as well as the key difference among the designs is that we minimize bandwidth utilization by minimizing needless data movement on as well as off-chip between processing layers.

The calcparams module is called first to obtain the R as well as C dimensions for each fused layer, after which the compute module is invoked. The reuse module may prepare the input between compute operations by combining data from the previous convolution output and data from the load operation, which transfers data from off-chip memory. We include pooling and padding layers, which are essential for the CNN computation to be completed successfully. Because pooling along with padding is not computationally expensive, earlier work has mostly ignored them in favor of building convolutional layer accelerators. We combine these actions when fusing layers, however, to ensure that the entire system is complete.

At design time, the calcparams module is configured with the values *X*, *Y*, *Sx*, as well as *Sy* (pyramid base width, height, along with stride between neighboring pyramids) described above. The row, as well as column values, indicate the input data to transfer from off-chip Memory (row_t, $column_t$, *inH*1, *inW*1) along with the dimensions of each layer's computation at each iteration using the following formulas:

$$row_t = \begin{cases} Y + (row-1)Sy - (K-S), & if\ row > 0 \\ 0, & if\ row = 0 \end{cases} \quad (5)$$

$$columnt_t = \begin{cases} X + (column-1)Sx - (K-S), & if\ column > 0 \\ 0, & if\ column = 0 \end{cases} \quad (6)$$

$$inputW_n = \begin{cases} X, & if\ n=1 \,\&\, col = 0 \\ Sx + K - S, & if\ n=1 \,\&\, col > 0 \\ OutputW_n - 1, & if\ n > 1 \end{cases} \quad (7)$$

$$inputH_n = \begin{cases} Y, & if\ n=1 \,\&\, row = 0 \\ Sy + K - S, & if\ n=1 \,\&\, row > 0 \\ OutputH_n - 1, & if\ n > 1 \end{cases} \quad (8)$$

Finally, the fused accelerator is pipelined to overlap the compute levels so that level 2 may begin processing as soon as level 1 completes its initial step. The pipeline steps that comprise the compute modules must be balanced to make good use of the FPGA resources assigned to the accelerator. This is accomplished by determining an acceptable set of T_m as well as T_n unroll factors for each layer; greater values improve parallelism inside the module, allowing the associated layer to complete faster.

Experiments Setup

A. Dataset used

The suggested machine learning system is evaluated on an MRI brain tumor dataset as shown in Fig. 6. The MRI brain tumor dataset includes 826 MRI pictures of patients with glioma tumors, 822 MRI images of patients with meningioma tumors, 827 MRI images of patients with pituitary tumors, as well as 835 MRI images of normal people.

B. Implementation

Our suggested solution is written in Python and implemented with Keras. Table 1 shows the hyper-parameters we employed in our research. We divided each dataset into train and test sets in a 70:30 split. To display the final averaged performance for each dataset, 5 various arbitrary train/test divisions are used. To minimize over-fitting at the time of training, we fixed 20% of the train for verification along with changing the rate of learning value for the sake of individual epochs as given in Eq. (9):

$$\alpha_n = \alpha_0 * 0.4 * (1 + epoch(n)) / 4 \quad (9)$$

where, α_n, $\alpha_{0,}$ as well as *epoch*(*n*), describe the learning rate at nth epoch, initial learning rate, along with a subsequent number of the current epoch.

Table 1: Hyper-parameters used in our study https://doi.org/10.1371/journal.pone.0264586.t002

Parameters	Values
Size of image	224 × 224
Size of batch	64
Number of epochs	60
Range of rotation	90
Range of shear	0.2
Vertical flip	True
Optimizer	Adam
Learning rate α	0.0001
Loss	Categorical cross-entropy
Horizontal flip	True

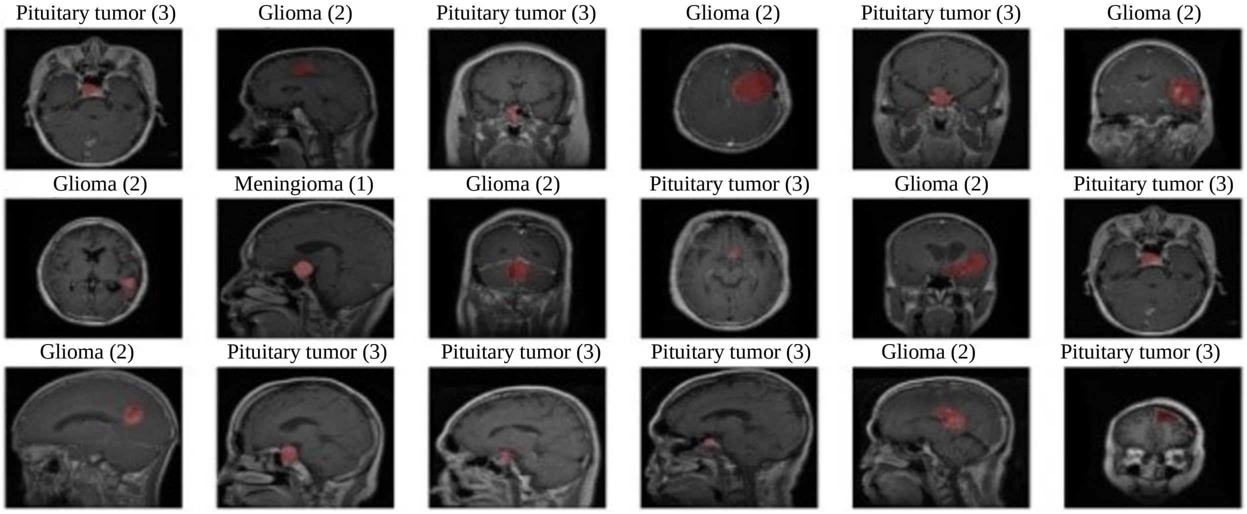


Fig. 6: Dataset of brain tumor

C. Evaluation Metrics

To assess the effectiveness of our proposed model, we employ 7 assessment criteria (Eqs. 10 to 16). These are computed adopting a confusion matrix derived in distinction to classifications. The confusion matrix compares actual classes to expected classes. During classifications, the diagonal of the confusion matrix indicates the properly identified occurrences:

$$Precision = TP/(TP+FP) \quad (10)$$

$$Recall = TP/(TP+FN) \quad (11)$$

$$F1 = 2*(P*R)/(P+R) \quad (12)$$

$$Acc = (TP+TN)/(TP+TN+FP+FN) \quad (13)$$

where, *TP*, *TN*, *FP*, as well as *FN*, describes true positive, true negative, false positive as well as false negative for a particular class. Equivalently, *Precision*, *Recall*, and *F*1 illustrate precision, recall as well as *F*1-score for a particular class:

$$kappa-score = (p_0 - p_e)/(1-p_e) \quad (14)$$

where, p_0 is the observed agreement ratio and p_e is the hypothetical probability of chance agreement:

$$MAF1 = (1/|L|)\sum_{a \in L} F1_a \quad (15)$$

$$WAF1 = \left(1/\sum_{a \in L} Supp(a)\right)\sum\left(F1_a Supp(a)\right) \quad (16)$$

where, *L* illustrates the set of classes, and supp (a) Denotes the number of samples in a particular class. *MAF*1 and *WAF*1 illustrate macro-average *F*1-score as well as weighted-average *F*1-score accordingly.

Results and Analysis

Three distinct deep-learning models were utilized in this study to detect brain tumor illness patients. On MRI scans, popular pretrained deep learning models such as MobileNetV2, EfficientNetV1, and InceptionResNetV2 were trained and assessed.

A. MobileNet-V2

The MobileNet-V2 model has a total of 2,263,108 parameters. The training parameters are 2,228,996, whereas the non-training parameters are 34,112.

The classification metrics of MobileNetV2 for characteristics such as precision, recall, as well as *F*1-score are shown in Fig. 7, along with a macro-average accuracy of 98% along with weighted-average accuracy of 98% for precision, recall, as well as *F*1-score.

The confusion matrix for MobileNetV2 is shown in the output Fig. 8, with exponential values as 4.2e+02, which means 420 ($4.2*10^2$) accurate values in decimal for no tumor, with exponential values as 3.5e+02, which means 350 ($3.5*10^2$) accurate values in decimal for glioma tumor, with exponential values as 3.4e+02, that means 340 ($3.4*10^2$) accurate values in decimal for meningioma tumor as well as with exponential values as 3.8e+02, that means 380 ($3.8*10^2$) accurate values in decimal for a pituitary tumor.

The graphical representation of the results in terms of accuracy or loss for different epochs for MobileNetV2 is shown in Fig. 9. The testing accuracy, as well as loss, is practically constant after a specific number of epochs, although the validation accuracy, as well as loss, vary considerably at first for several epochs, then settle to a constant value after a certain number of epochs.

B. InceptionResNetV2

The total amount of parameters employed in the InceptionResNetV2 model is 54,342,884. 54,282,340 are utilized as training parameters, whereas 60,544 are used as non-training parameters.

The classification metrics of InceptionResNetV2 for parameters such as precision, recall, as well as *F*1-score are shown in Fig. 10, along with macro-average accuracy of 100% as well as weighted-average accuracy of 98% for precision, recall, and *F*1-score.

The confusion matrix for InceptionResNetV2 is shown in the output Fig. 11, with exponential values as 4.2e+02, which means 420 (4.2*102) accurate values in decimal for no tumor, with exponential values as 3.5e+02, which means 350 (3.5*102) accurate values in decimal for glioma tumor, with exponential values as 3.4e+02, that means 340 (3.4*102) accurate values in decimal for meningioma tumor as well as with exponential values as 3.8e+02, that means 380 (3.8*102) accurate values in decimal for a pituitary tumor.

The graphical representation of the results in terms of accuracy or loss for different epochs for InceptionResNetV2 is shown in Fig. 12. The testing accuracy, as well as loss, is practically constant after a specific amount of epochs, but the validation accuracy, as well as loss, vary substantially at first for several epochs, then settle to a constant value after a certain number of epochs.

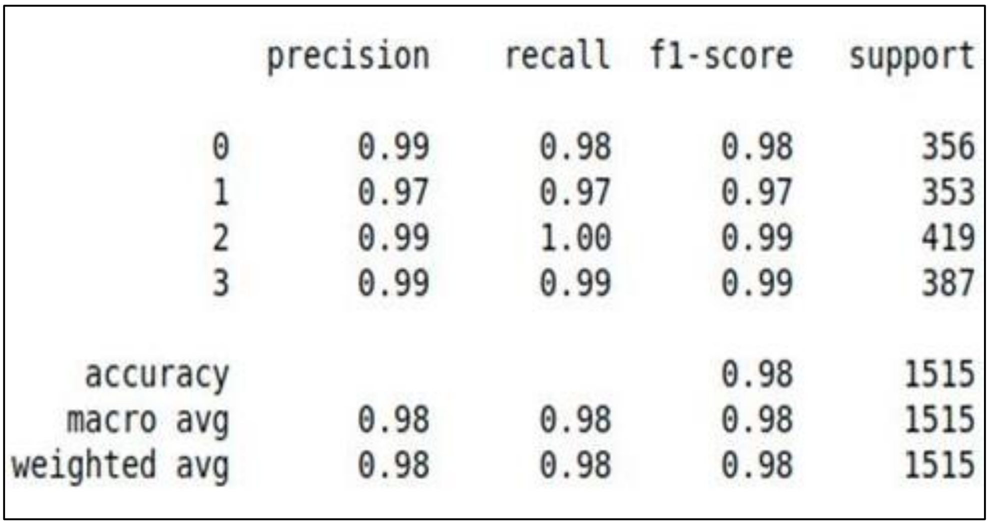
C. EfficientNet

The total number of parameters employed in the EfficientNet model is 6,580,363. 6,518,308 are utilized as training parameters, whereas 62,055 are used as non-training parameters.

The classification metrics of EfficientNet for parameters such as precision, recall, as well as *F*1-score are shown in Fig. 13, as well as macro-average accuracy of 100% as well as weighted-average accuracy of 100% for precision, recall, and *F*1-score.

The confusion matrix for EfficientNet is shown in the output Fig. 14, with exponential values as 4.2e+02, which means 420 ($4.2*10^2$) accurate values in decimal for no tumor, with exponential values as 3.5e+02, which means 350 ($3.5*10^2$) accurate values in decimal for glioma tumor, with exponential values as 3.5e+02, that means 350 ($3.5*10^2$) accurate values in decimal for meningioma tumor as well as with exponential values as 3.9e+02, that means 390 ($3.9*10^2$) accurate values in decimal for a pituitary tumor.

The graphical representation of the results in terms of accuracy or loss for different epochs for EfficientNet is Fig. 15. The testing accuracy, as well as loss, is practically constant after a specific amount of epochs, but the validation accuracy, as well as loss, vary substantially at first for several epochs, then settle to a constant value after a certain number of epochs.



```
              precision    recall  f1-score   support

           0       0.99      0.98      0.98       356
           1       0.97      0.97      0.97       353
           2       0.99      1.00      0.99       419
           3       0.99      0.99      0.99       387

    accuracy                           0.98      1515
   macro avg       0.98      0.98      0.98      1515
weighted avg       0.98      0.98      0.98      1515
```

Fig. 7: Classification metrics for the MobileNetV2 model

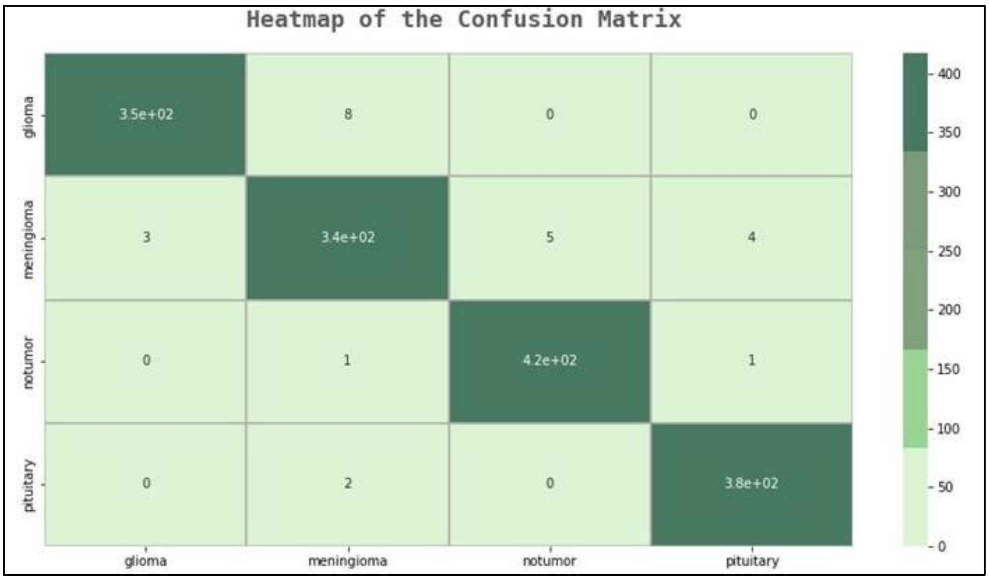


Fig. 8: Confusion matrix for MobileNetV2

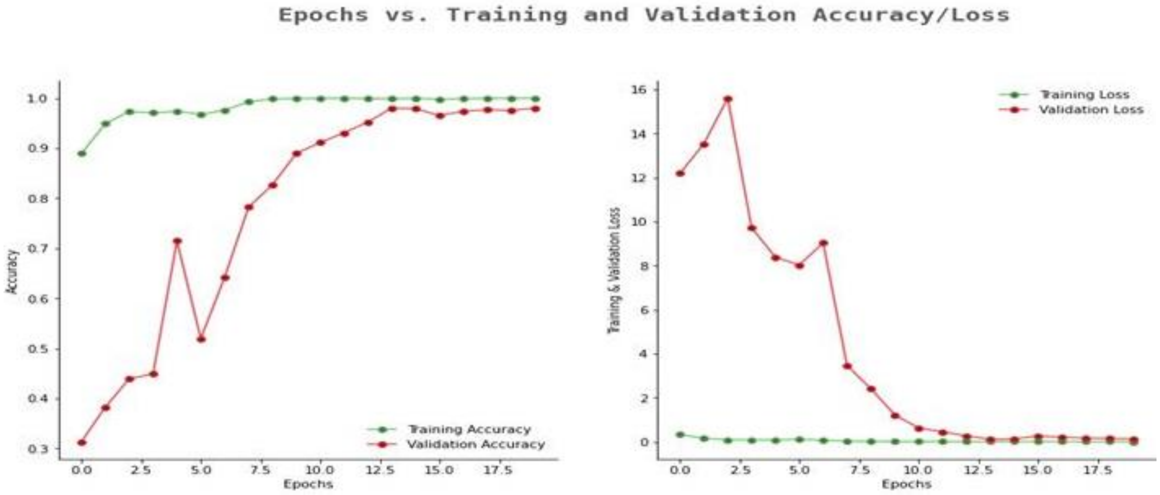
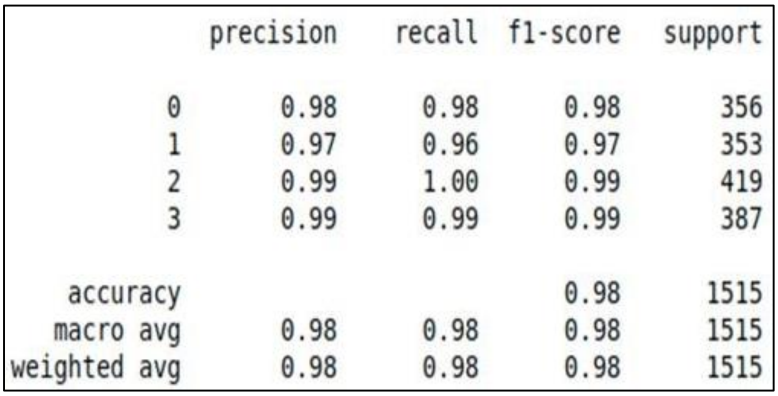


Fig. 9: Comparison of accuracy and loss graph for MobileNetV2


```
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       356
           1       0.97      0.96      0.97       353
           2       0.99      1.00      0.99       419
           3       0.99      0.99      0.99       387

    accuracy                           0.98      1515
   macro avg       0.98      0.98      0.98      1515
weighted avg       0.98      0.98      0.98      1515
```

Fig. 10: Classification metric for the inception of the ResNetV2 model

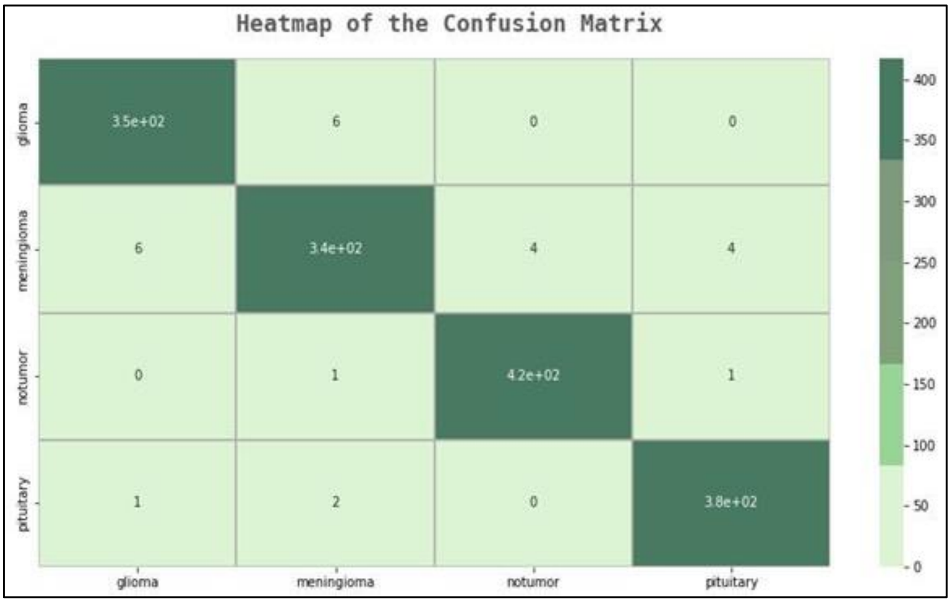


Fig. 11: Confusion matrix of InceptionResNetV2 model

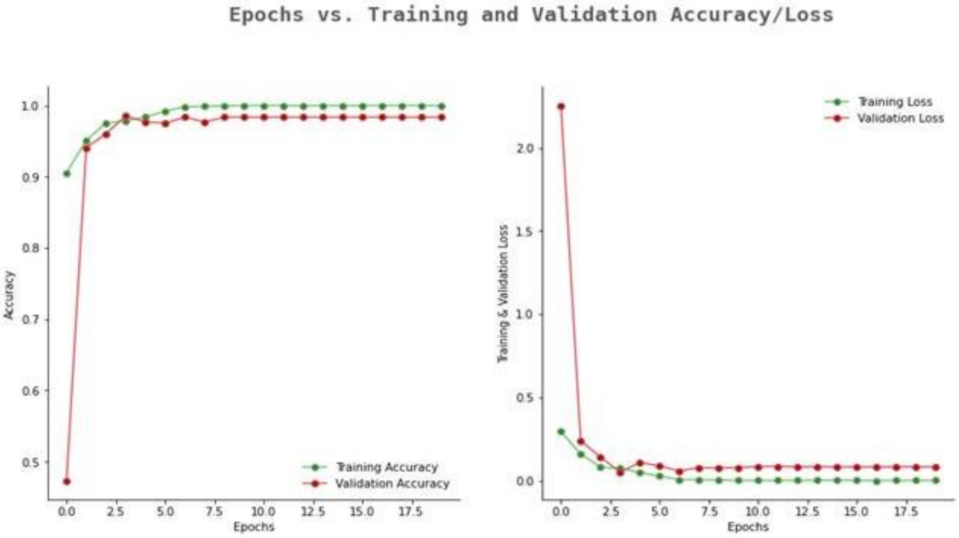
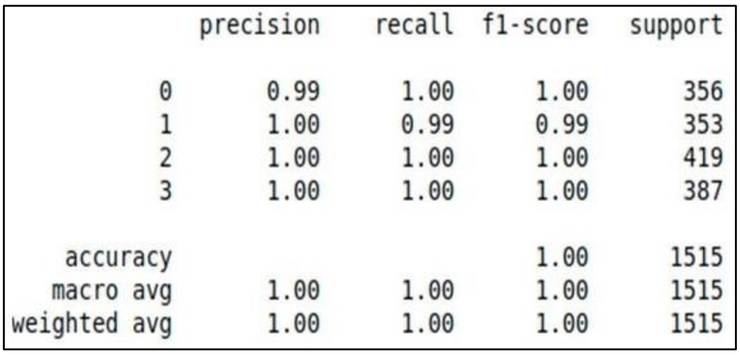


Fig. 12: Comparison of accuracy and loss Graph for inception res net Model



```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       356
           1       1.00      0.99      0.99       353
           2       1.00      1.00      1.00       419
           3       1.00      1.00      1.00       387

    accuracy                           1.00      1515
   macro avg       1.00      1.00      1.00      1515
weighted avg       1.00      1.00      1.00      1515
```

Fig. 13: Classification metric for EfficientNet model

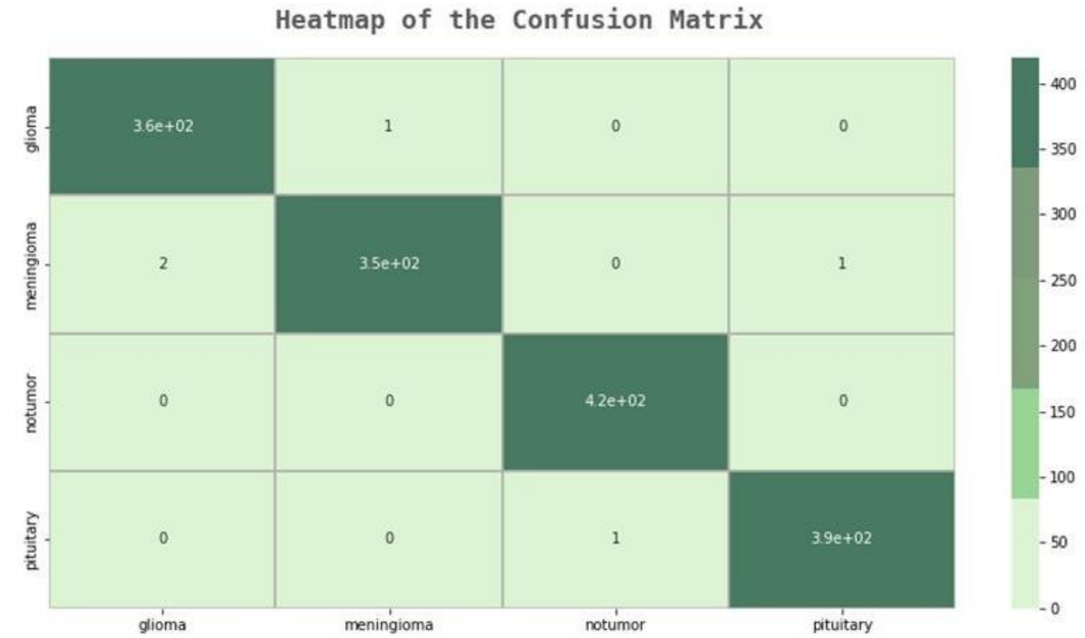


Fig. 14: Confusion matrix for EfficientNet

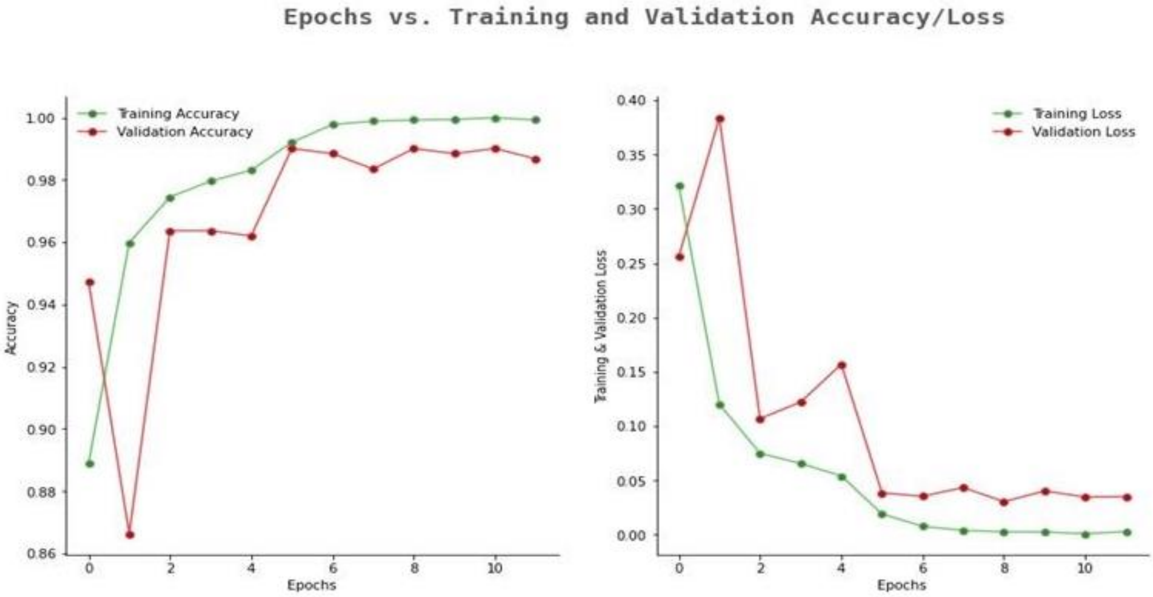


Fig. 15: Comparison of accuracy and loss graph for EfficientNet

Discussion

The prediction results were evaluated using the following parameters: Precision, recall, $F1$-score, as well as accuracy. Table 2 shows the obtained precision, recall, $F1$-score, and support for the classification of ML models. MobileNetV2 models and InceptionResNetV2 models exhibit nearly identical results for image categorization of brain tumors, with both macro-average as well as weighted-average accuracy of 98%. Because of its increased generalization and inbuilt ensemble learning feature, the EfficientNet model produces the greatest results, with 100% accuracy for both macro-average as well as weighted-average. Larger networks with more breadth, depth, or resolution are more accurate. We suggested CNN model outperforms all previous models trained in terms of experimental performance, with an accuracy of 97%.

Table 2: A comparative study of categorization metrics using three distinct deep-learning approaches was conducted

		Precision	Recall	F1-score	Support
MobileNetV2	Glioma tumor	0.99	0.98	0.98	356
	Meningioma tumor	0.97	0.97	0.97	353
	No tumor	0.99	1.00	0.99	419
	Pituitary tumor	0.99	0.99	0.99	387
InceptionResNetV2	Glioma tumor	0.98	0.98	0.98	356
	Meningioma tumor	0.97	0.96	0.97	353
	No tumor	0.99	1.00	0.99	419
	Pituitary tumor	0.99	0.99	0.99	387
EfficientNet	Glioma tumor	0.99	1.00	1.00	356
	Meningioma tumor	1.00	0.99	0.99	353
	No tumor	1.00	1.00	1.00	419
	Pituitary tumor	1.00	1.00	1.00	387

A. Related Problems

There are several unresolved issues regarding brain tumor classification. Brain tissue classification, (known as anatomical brain classification), tries to assign each unit a different brain tissue class. Tumor detection seeks to distinguish aberrant tumors or lesions, and also gives the tissue's expected class. It should be noted that certain brain tumor segmentation research approaches only provide an individual description of the classification mask or the tumor core's center point beyond sub-region classification. The process of extracting predefined features from brain scan images, then classifying feature representation into graded disorders such as High-Grade Gliomas (HGG) and Low-Grade-Gliomas (LGG), Mild Cognitive Impairment (MCI) (Suk *et al*., 2016), Alzheimer's Disease (AD) (Suk and Shen, 2016) and Schizophrenia (Pinaya *et al*., 2016) is referred to as disorder classification. In addition to clinical diagnosis, survival prediction analyses tumor patterns and activity (Yoo *et al*., 2016) to predict survival rates (Rota Bulo *et al*., 2017). Based on the tumor segmentation results, both disease categorization and survival prediction may be considered downstream tasks.

B. Research Challenges

Despite tremendous advances in brain tumor segmentation, updating deep learning methods continues to face problems, with various obstacles to overcome. The difficulties connected with brain tumor segmentation are classified as follows:

1) Uncertainty of Location-Glioma develops from the gluey cells that surround nerve cells. Because of the extensive geographical dispersion of gluey cells, HGG (High-Grade Glioma) or LGG (Low-Grade Glioma) can form anywhere in the brain
2) Morphological Uncertainty-Unlike a hard item, the morphology, such as the size along with the shape of distinct brain tumors, changes including great concern. Being the outer layer of a brain tumor, edoema tissues exhibit a variety of fluid structures that give little prior information for characterizing tumor forms. Tumor subregions can also differ in form as well as size
3) Low contrast-images with high resolution as well as contrast are likely to include a wide range of visual information (Liu *et al*., 2016). As a result of the image projection as well as tomography procedures, MRI images may be of low quality along with contrast. A distinction between biological tissues is sometimes hazy and difficult to discern. Cells along the boundary are difficult to classify, making exact segmentation more complex and difficult to perform
4) Annotation bias-because manual annotation is mostly based on human experience; annotation bias might be introduced during data labeling. During the learning process, annotation biases have a significant influence on the segmentation algorithm (Chen and Joo, 2021)
5) Imbalanced issues-this has an impact on the data-driven learning process since the retrieved characteristics may be heavily impacted by big tumor areas (Rota Bulo *et al*., 2017)

Conclusion

We created a novel convolution-attention module-based MobileNetV2 classifier for brain tumor images in this study. This study provides a comparative analysis of three CNN architecture-based models for brain tumor classification. The categorization was carried out utilizing a brain tumor MRI image database that has four classes, three of which are brain tumor kinds and the remaining category is the normal class.

Deep Convolutional Neural Networks (DCNNs) are gaining popularity in a variety of applications, generating a lot of interest in the development of accelerators for them. The conventional method for constructing CNN accelerators has been to process layers repeatedly until they are complete, necessitating off-chip storage of intermediate data. The fundamental advantage of the fused-layer assessment technique is the ability to preserve inter-layer intermediate data on the chip, reducing off-chip transmission. A suggested

Machine Learning system is evaluated on an MRI brain tumor dataset. The MRI brain tumor dataset includes 826 MRI pictures of patients with glioma tumors, 822 MRI images of patients with meningioma tumors, 827 MRI images of patients with pituitary tumors, as well as 835 MRI images of normal people.

Our technique has certain drawbacks. First, we solely employ online data augmentation for our trials. Second, it would be worthwhile to investigate the integration of characteristics derived from different layers of MobileNetV2 to improve the performance of brain tumor classifications. We must also look at how our technology can be used effectively in a mobile environment or on a cutting-edge computer platform, especially in an Internet of Things (IoT) scenario.

Acknowledgment

I am grateful to all of those with whom I have had the pleasure to work during this research paper. Each of the members has provided me extensive personal and professional guidance and taught me a great deal about both scientific research and life in general.

Funding Information

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Author’s Contributions

Mitrabinda Khuntia: Study conception and designed, data collection, analysis and interpretation of results, draft manuscript preparation.

Prabhat Kumar Sahu: Study conception and designed analysis and interpretation of results, draft manuscript preparation.

Swagatika Devi: Draft manuscript preparation.

Ethics

Ethical considerations in research are a set of principles that guide your research designs and practices. These principles include voluntary participation, in- formed consent, anonymity, confidentiality, potential for harm, and results communication. Data, results, methods and procedures, and publication status are reported honestly. They are not fabricate, falsify, or misrepresent data. We have tried to avoid bias in experimental design, data analysis, data interpretation, peer review, personnel decisions, grant writing, expert testimony, and other aspects of research. We have carefully and critically examined your own work and the work of your peers. Keep good records of research activities.

References

Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018, December). Mesonet: A compact facial video forgery detection network. In *2018 IEEE international Workshop on Information Forensics and Security (WIFS)* (pp: 1-7). https://ieeexplore.ieee.org/abstract/document/8630761

AlBadawy, E. A., Saha, A., & Mazurowski, M. A. (2018). Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical Physics*, *45*(3), 1150-1158. https://doi.org/10.1002/mp.12752

Arunachalam, M., & Royappan Savarimuthu, S. (2017). An efficient and automatic glioblastoma brain tumor detection using shift-invariant shearlet transform and neural networks. *International Journal of Imaging Systems and Technology*, *27*(3), 216-226. https://doi.org/10.1002/ima.22227

Chang, K., Beers, A. L., Bai, H. X., Brown, J. M., Ly, K. I., Li, X., ... & Kalpathy-Cramer, J. (2019). Automatic assessment of glioma burden: A deep learning algorithm for fully automated volumetric and bidimensional measurement. *Neuro-Oncology*, *21*(11), 1412-1422. https://doi.org/10.1093/neuonc/noz106

Chen, Y., & Joo, J. (2021). Understanding and mitigating annotation bias in facial expression recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp: 14980-14991).

Chu, J., Cai, J., Li, L., Fan, Y., & Su, B. (2021). Bilinear Feature Fusion Convolutional Neural Network for Distributed Tactile Pressure Recognition and Understanding via Visualization. *IEEE Transactions on Industrial Electronics*, *69*(6), 6391-6400. https://ieeexplore.ieee.org/abstract/document/9454268

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXivpreprint arXiv:2010.11929*. https://arxiv.org/abs/2010.11929

Hasan, S. K., & Linte, C. A. (2018, October). A modified U-Net convolutional network featuring a nearest-neighbor re-sampling-based elastic-transformation for brain tissue characterization and segmentation. In *2018 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)* (pp: 1-5). https://ieeexplore.ieee.org/abstract/document/8576421

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., ... & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, *35*, 18-31. https://doi.org/10.1016/j.media.2016.05.004

Hossain, M. S., Al-Hammadi, M., & Muhammad, G. (2018). Automatic fruit classification using deep learning for industrial applications. *IEEE Transactions on Industrial Informatics*, *15*(2), 1027-1034. https://ieeexplore.ieee.org/abstract/document/8488544

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXivpreprint arXiv:1704.04861*. https://arxiv.org/abs/1704.04861

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision, Springer, Cham*. (pp: 21-37). https://doi.org/10.1007/978-3-319-46448-0_2

Mishra, B., & Shahi, T. B. (2021). Deep learning-based framework for spatiotemporal data fusion: An instance of landsat 8 and sentinel 2 NDVI. *Journal of Applied Remote Sensing*, *15*(3), 034520. https://doi.org/10.1117/1.JRS.15.034520

Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Transactions on Medical Imaging*, *35*(5), 1240-1251. https://ieeexplore.ieee.org/abstract/document/7426413

Pinaya, W. H., Gadelha, A., Doyle, O. M., Noto, C., Zugman, A., Cordeiro, Q., ... & Sato, J. R. (2016). Using deep belief network modelling to characterize differences in brain morphometry in schizophrenia. *Scientific Reports*, *6*(1), 1-9. https://doi.org/10.1038/srep38897

Rota Bulo, S., Neuhold, G., & Kontschieder, P. (2017). Loss max-pooling for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp: 2126-2135). https://openaccess.thecvf.com/content_cvpr_2017/html/Bulo_Loss_Max-Pooling_for_CVPR_2017_paper.html

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp: 4510-4520). https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXivpreprintarXiv:1409.1556*. https://arxiv.org/abs/1409.1556

Sitaula, C., Xiang, Y., Aryal, S., & Lu, X. (2021). Scene image representation by foreground, background and hybrid features. *Expert Systems with Applications*, *182*, 115285. https://doi.org/10.1016/j.eswa.2021.115285

Suk, H. I., & Shen, D. (2016, October). Deep ensemble sparse regression network for Alzheimer's disease diagnosis. In *International Workshop on Machine Learning in Medical Imaging, Springer, Cham.* (pp: 113-121). https://doi.org/10.1007/978-3-319-47157-0_14

Suk, H. I., Wee, C. Y., Lee, S. W., & Shen, D. (2016). State-space model with deep learning for functional dynamics estimation in resting-state fMRI. *Neuro Image*, *129*, 292-307. https://doi.org/10.1016/j.neuroimage.2016.01.005

Vickers, N. J. (2017). Animal communication: When i'm calling you, will you answer too? *Current Biology*, *27*(14), R713-R715. https://doi.org/10.1016/j.cub.2017.05.064

Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp: 3-19). https://openaccess.thecvf.com/content_ECCV_2018/html/Sanghyun_Woo_Convolutional_Block_Attention_ECCV_2018_paper.html

Xiang, Q., Wang, X., Li, R., Zhang, G., Lai, J., & Hu, Q. (2019, October). Fruit image classification based on Mobilenetv2 with transfer learning technique. In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering* (pp: 1-7). https://doi.org/10.1145/3331453.3361658

Yoo, Y., Tang, L. W., Brosch, T., Li, D. K., Metz, L., Traboulsee, A., & Tam, R. (2016). Deep learning of brain lesion patterns for predicting future disease activity in patients with early symptoms of multiple sclerosis. In *Deep learning and data labeling for medical applications*, *Springer, Cham.* (pp: 86-94). https://doi.org/10.1007/978-3-319-46976-8_10

Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015, February). Optimizing FPGA-based accelerator design for deep convolutional neural networks. In *Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays* (pp: 161-170). https://doi.org/10.1145/2684746.2689060

Zhou, T., Ruan, S., Guo, Y., & Canu, S. (2020, April). A multi-modality fusion network based on attention mechanism for brain tumor segmentation. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)* (pp: 377-380). https://ieeexplore.ieee.org/abstract/document/9098392