INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Brain Tumor Detection Using Transfer Learning in Deep Learning

Sri Sai Meghana Alla[1], **Kavitha Athota**[2]*

[1] Department of Computer Science and Engineering, JNTUH College of Engineering, Hyderabad 500085, India
[2] Department of Computer Science and Engineering, JNTUH College of Engineering, Hyderabad 500085, India

OPEN ACCESS

Received: 21-06-2022

Accepted: 16-09-2022

Published: 29-10-2022

Citation: Alla SSM, Athota K (2022) Brain Tumor Detection Using Transfer Learning in Deep Learning. Indian Journal of Science and Technology 15(40): 2093-2102. https://doi.org/10.17485/IJST/v15i40.1307

* **Corresponding author**.

athotakavitha@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Background/Objectives: Magnetic resonance imaging (MRI) is widely used for tumor evaluation. However, MRI generates enormous data, making manual segmentation difficult in a reasonable amount of time, which limits the use of accurate measurements in clinical practice. Therefore, this study focuses on the automatic and reliable segmentation methods which are needed for early diagnosis of brain tumors. **Methods:** In this study, we used deep learning-based convolutional neural networks (CNNs) to extract features and automatically classify brain tumors based on MRI images. In addition to conventional CNNs, the application of transfer learning was investigated by using three types of CNNs (Inception-V3, VGG-16, and VGG-19) to achieve reasonable accuracy, with fine tuning of the final layers to improve the accuracy of the models. **Findings:** The results show that applying transfer learning to a CNN achieves high accuracy in less time and with a smaller dataset. VGG-19 achieves 97 % accuracy, and VGG-16 achieves 96 % accuracy, which is better than the accuracy of Inception-V3 (89 %). Our proposed model using CNNs with transfer learning provides more robust automatic and reliable segmentation methods. **Novelty:** The novelty of this work is the use of Transfer Learning in conjunction with deep learning-based CNNs, as Transfer Learning provides a novel technique to analyse data with few annotations by transferring information from the source domain to the target domain.

Keywords: Brain tumor detection; Convolutional Neural Networks; Malignant; Benign; Transfer Learning VGGNET; InceptionV3; MRI

1 Introduction

Brain tumor treatments include surgery, chemotherapy, radiotherapy, and a combination of these treatments. Even with critical medical monitoring, patients usually do not survive more than 14 months [1]. Early detection can increase patients' chances of survival; therefore, it becomes one of the most important phases of careful treatment planning.

Magnetic resonance imaging (MRI) can produce high-quality images, and is therefore considered efficient for tracking brain tumors [2]. Convolutional neural networks (CNNs)[3] are mainly used for image classification tasks because they are based on the biological neurons in the human visual cortex. Therefore, they learn the features responsible for accurate classification of images[4].

Two CNN architectures, Inception-V3 and VGG-Net, with features and weights extracted during training on an Image-net dataset[3] with more than one million images from 1000 different classes, are used to classify the given MRI images into images with tumors and those without. The limitation of using CNNs without transfer learning is that large datasets are needed to train the models. When a large dataset is not available, these neural networks do not provide satisfactory results for classification problems. To overcome this limitation, i.e., to achieve acceptable accuracy even with a smaller dataset, the transfer learning technique is useful. Because the dataset used in this study contained only 3000 MRI images[5], it was insufficient to train a neural network from scratch, and transfer learning was used. The parameters were updated based on the features present in the dataset.

Abdalla and Esmail[5] proposed a method for brain tumor detection using artificial neural networks (ANNs). This method uses ANNs to classify normal and tumorous MRI images of the brain. The method uses statistical feature analysis to extract features from the images and detect tumors.

Another automatic brain tumor detection method was proposed using CNNs with 3 × 3 kernels[6]. This architecture was used to classify brain MRI images from the BRaT dataset. Because the dataset was very large (almost 7 GB), training the architecture on multi-core GPU systems took 30 h.

A two-phase multi-model automatic diagnosis of brain tumors was proposed[7], in which the CNNs were used to classify MRI images into normal and abnormal images. In this method, the CNN models Alex-Net, VGG-16, and VGG-19 are used and the method is tested on images from the RIDER neuro MRI database. This database contains 349 images, including 240 healthy and 109 unhealthy images. In this study, the CNN was trained with transfer learning because the dataset used was insufficient to train a model from scratch.

The model in[8] describes a method for extracting and concatenating multi-level characteristics for early detection of brain tumors. This model is validated by two pre-trained deep learning models: Inception-v3 and DensNet201. Two alternative scenarios for brain tumor detection and classification were examined using these two models.

A model for automatic brain tumor detection was proposed[9] using pre-trained VGG-16 and Inception-V3, using a dataset with 253 images. This dataset contains 155 tumor images and 98 healthy images. The dataset was too small to fine-tune the CNNs and the test dataset was too small to determine the accuracy of the model.

A model for automatic brain tumor detection was proposed[10] using VGG-16 with the BRaTs dataset. The model achieved 84% accuracy using transfer learning and fine-tuning for 50 epochs.

N. Srivastava et.al. in[11] proposed a dropout technique for addressing overfitting in neural networks by randomly dropping units and their connections.

P. Dvorák et.al. in[12] chose convolutional neural network as the learning algorithm because it is well-suited to dealing with feature correlation. They tested the technique on the public BRATS2014 data set with three multimodal segmentation tasks, obtaining cutting-edge results for this brain tumor segmentation data set of 254 multimodal volumes in only 13 seconds per volume.

S. Irsheidat et.al. in[13] built a model based on Artificial Convolutional Neural Networks that takes magnetic resonance pictures and analyses them using mathematical formulas and matrices operations. This neural network predicts the likelihood of the presence of a tumor in the brain, and it was trained using magnetic resonance images from 155 healthy brains and 98 tumors. The collection contains a total of 253 magnetic resonance pictures.

Sravya et al.[14] investigated brain tumor detection and presented some important challenges and techniques.

An automated brain tumor detection system[15] was proposed and studied using the YOLO model and the deep learning library FastAi with the BRATS 2018 dataset, which contained 1,992 MRI scans of the brain. The authors achieved 85.95 % accuracy for YOLO and 95.78 % for the FastAi classification model.

A brain tumor detection application[16] was proposed to classify MRI images as tumorous and non-tumorous using the VGG-16 model. The authors used the Kaggle dataset for training and showed an improvement in accuracy. However, the authors trained the entire model.

From a literature review I is learnt that the CNN is a promising approach for accurate image classification problems. CNNs use kernels and feature maps to extract features, reducing the dimensions of the input. This helps to increase the efficiency of the model with respect to time and memory constraints.

Insufficient training data leads to lower accuracy. For any classification problem with insufficient training data, transfer learning is an effective way to achieve higher accuracy because it requires fewer computations and fewer trainable parameters than training from scratch. Therefore, it provides higher accuracy, even if the amount of data is smaller. Fine-tuning is another technique that can be applied to increase the accuracy of the model when the classification problem is different from the problem used for transfer learning.

In this study, an automatic brain tumor detection model, using Inception-V3, VGG-16, and VGG-19 architectures, is developed based on transfer learning. In addition, fine tuning is performed to improve the classification accuracy, because the dataset used to train the model with transfer learning is different from the dataset used for the problem for which the model was trained.

In this study, automatic classification of brain MRI images into healthy images and images with tumors is proposed.

2 Methodology

In this section, we give a brief description of the dataset used and an introduction to CNNs, transfer learning, and fine-tuning.

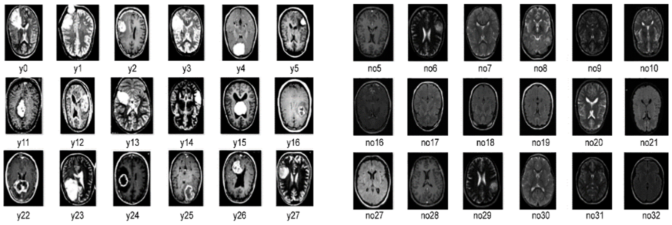

Fig 1. MRI images of two categories: a) with tumor b) without tumor

Table 1. Description of the dataset

Training Set		Test Set	
Yes	No	Yes	No
1050	1050	450	450

2.1 Brain Tumor Dataset

The experiments described in this study were performed using a publicly available dataset acquired from a Kaggle warehouse. This dataset consisted of 1500 brain MRI images with tumors and 1500 brain MRI images without tumors. All images were two-dimensional and had a height and width of 256 × 256 pixels. All images were skull-stripped and labeled yes if they contained a tumor and no if they did not. Figure 1 shows the dataset of images with and without tumors labeled yes and no, respectively[5]. The descriptions of the training and testing datasets are listed in Table 1.

2.2 Preprocessing of the Dataset

The dataset contains 3000 images with and without a brain tumor. All images need to be resized to 224 × 224 pixels and then converted into vectors, which serve as input to the neural network. All images were then converted to grayscale images. This preprocessed data, along with the label of the image, is provided as input to the neural network. Label 0 represents an image without a tumor and label 1 represents an image with a tumor.

2.3 Data Augmentation

Although we have a dataset of 3000 MRI images, a CNN model with one million parameters cannot be trained because it is considered insufficient. As stated in the objectives, the solution to this problem is data augmentation, which is more effective than manual segmentation because it takes more time. Similarly, as more data is used, this approach minimizes the occurrence of errors, which are seen as major setbacks in these types of applications. This is a technique for artificially increasing the size of existing data by rotating, scaling, and adding noise. Data can be enlarged by flipping the images horizontally or vertically by a certain angle, zooming the image, and increasing or decreasing the brightness range. All of this was used to magnify our data. Each image was augmented by approximately 16 times the original data. This ensures that the model is not overfitted to the data[11]. Another method for reducing overfitting on models is data augmentation, which involves increasing the amount of training data by using only information from the training data. The purpose of augmentation is to not only prevent overfitting but also to supplement data in such a way that it best improves the classifier.

2.4 Convolutional Neural Network

A CNN[3] is a type of feed-forward ANN in which the connectivity pattern between neurons is inspired by the neurons of an animal's visual cortex.

It mainly consists of a sequence of four steps for classification: convolution, activation, pooling, and fully connected layers, as shown in Figure 2. The convolutional layer, activation layer, and pooling layer were applied multiple times.

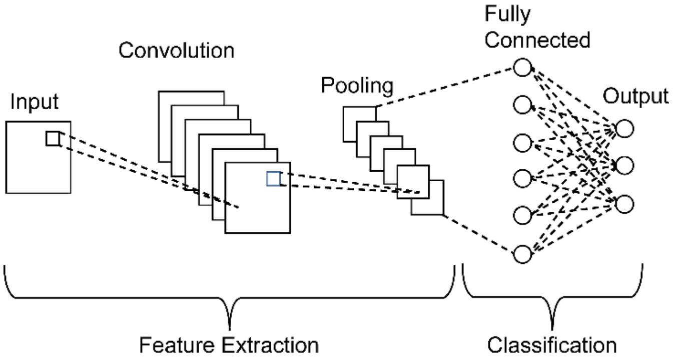

Fig 2. CNN Architecture

Convolutional layer: This layer involves a mathematical operation that requires two inputs: the input image matrix and a filter. The input image was multiplied by the filter, and a feature map was generated as an output.

Activation layer: This layer includes an activation function that gives nonlinearity to the neural network. Rectifier linear units (ReLUs) are used because they increase the training speed. Equation (1) shows the mathematical equation for the ReLU activation.

$$\mathrm{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases}$$

Pooling layer: The main limitation of the convolutional layer is that it captures features that depend on their location. Thus, if the location of the feature in the image changes slightly, the classification becomes inaccurate. Pooling allows the network to overcome this limitation by making the representation more compact so that it is invariant to minor changes and insignificant details. Max pooling and average pooling were used to connect the features.

Fully connected layer: The features learned from the convolutional layers are finally fed into the fully connected layer. The term "fully connected" means that every node in this layer is connected to every other node in the next layer. The main purpose of this layer is to associate a class with a particular input image. This layer uses softmax activation.

Loss function: This function (H) must be minimized during training. The output is calculated after the image has passed through all the previous layers. It is compared to the desired output using the loss function, and the error rate is calculated. This process is repeated for several iterations until the loss function is minimized. The loss function we used was the categorical cross-entropy (CCE). Equation (2) shows the mathematical equation for CCE.

$$\mathrm{H} = -H = -\sum_{m=1}^{M} \quad y_m \cdot \log \hat{y}_m \tag{2}$$

Where $\widehat{y}_m$ represents predicted label and y_m represents target label of sample m among M number of samples.

2.5 Transfer Learning

Transfer learning[10] is a machine-learning technique that applies previously acquired knowledge to a different but related problem.

Transfer learning is mainly used when there is not enough training data available. Therefore, the features and weights obtained when training the architectures with the Image-net dataset were initialized. The Inception-V3[17], VGG-16[18], and

VGG-19[18] architectures were trained with an Image-net dataset containing 14 million images belonging to 20000 different classes.

These models then receive the magnetic resonance images of the brain as input. The input, output, and fully connected layers with soft-max activation are trained, and the features are learned. These features were used to classify the images into two classes: healthy and tumor.

The Inception-V3[17] model, a deeper architecture with 311 convolutional layers, using transfer learning, trained 4098 parameters to achieve 87 % training accuracy and 86 % test accuracy. VGG-16, which is a shallow architecture with 16 convolutional layers, trained 1026 parameters to achieve 87 % training accuracy and 86 % test accuracy. VGG-19, a shallow architecture with 19 convolutional layers, trained 4096 parameters, achieving a training accuracy of 84 % and a test accuracy of 83 %.

2.6 Fine Tuning

The features of the Inception-V3[17], VGG-16[18], and VGG-19[18] architectures trained on Image-net datasets are used for transfer learning for brain tumor detection problems. In transfer learning, the weights and features of the input, output, and fully connected layers are trained by freezing the weights of all the other layers. However, to use the features of a model trained on a non-medical dataset to classify biomedical images, we need to obtain relevant features to classify these images. This was achieved by fine-tuning the last few layers of the model rather than just the fully connected layer. The fine-tuning is achieved by re-training the last few layers on the new dataset and updating all the weights of these layers[10,11].

For Inception-V3, 29,630,466 parameters were trained while fine-tuning the model for the last 156 layers, and an accuracy of 89 % was achieved on the training set and 89 % on the test set. In VGG-16, 11,800,066 parameters were trained while fine-tuning the model for the last nine layers, and an accuracy of 98 % was achieved on the training set, and 96 % was achieved on the test set. In VGG-16, 18,289,922 parameters were trained while fine-tuning the model for the last 11 layers. An accuracy of 98 % was achieved for the training set, and 97 % for the test set.

2.7 Evaluation Measures

The model was evaluated by using an accuracy metric. The accuracy of a model can be defined as the ratio between the number of correctly classified images/samples and the total number of images in the dataset. The accuracy of the training set and the test set was calculated using Equation (3).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{3}$$

Where TP represents True Postitives, TN represents True Negatives, FP represents False Positives, and FN represents False Negatives.

3 Method

1. Initialize the CNN and load its weights and features
2. Replace the input and output layers and the fully connected layer with new layers using the ReLU activation function.
3. Freeze all other layers and train the model
4. Finally, re-train the last layers to update all higher-level parameters responsible for accurate tumor classification, using a very low learning rate and the Adam optimizer.

3.1 Dataset and Parameters

The dataset used was publicly available on the Kaggle website, where a custom dataset containing 3000 brain MRI images was published for research purposes.

This dataset contains 1500 images with tumors and 1500 images without tumors, with each image having a size of 256 × 256. All these images were converted into 224 × 224 vectors and fed into the neural network as input along with their labels[18]. Because the data was insufficient to train these parameters, the dataset was enlarged using data augmentation. This also reduces the probability of overfitting[17].

Data augmentation includes many methods, including rotating images, flipping images, increasing/decreasing image brightness, and increasing/decreasing image size[19]. In our model, we rotated the image by 40°, enlarged it by 20 %, increased the brightness by 30 %, shifted the image to the left and right by 20 %, and flipped it horizontally. Figures 3 and 4 show the images before and after the data augmentation, respectively.

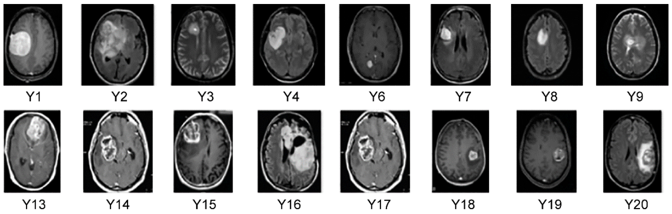

Fig 3. Images before augmentation

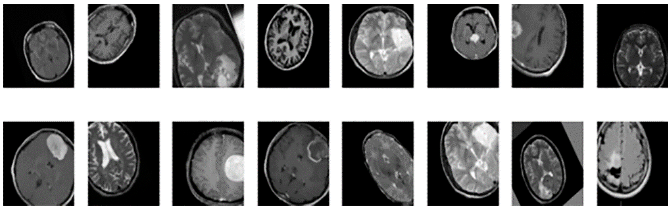

Fig 4. Images after augmentation

The proposed method was implemented in Python and Keras, using TensorFlow as the backend. 30 epochs were selected and the batch size was 128. The selected learning rate was set to 0.0001. The learning rate of the model was carefully chosen as a higher learning rate will cause the optimizer to diverge the loss function of the model instead of converging it.

3.2 Architectures

Because we aimed for an accurate and reliable tumor detection method, we used three CNN architectures, a deep architecture Inception-V3 and the others being shallow architectures VGG-16[20] and VGG-19[21].

Inception-V3: Inception-V3 is a deep architecture with 48 inception layers, each of which consists of four convolutional layers with activation functions and two max-pooling layers[22,23]. Thus, the architecture consists of 311 convolutional layers. Figure 5 shows a single inception layer.

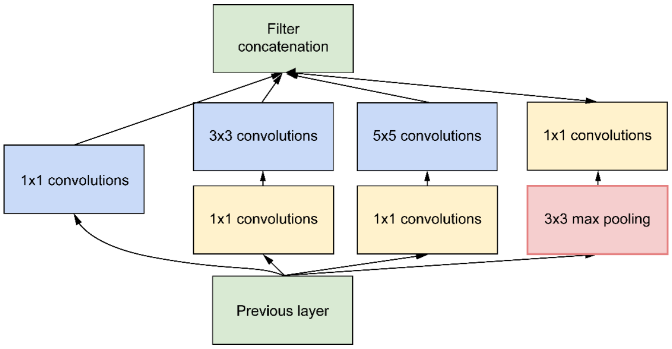

Fig 5. Inception Layer

VGG-16: VGG-16 has a shallow architecture with only 16 layers , as shown in Figure 6.

VGG-19: VGG-19 is a CNN architecture with 19 layers, as shown in Figure 7.

4 Results and Discussion

All architectures, Inception-V3, VGG-16, and VGG-19, were trained for 30 epochs with a batch size of 128 using categorical cross-entropy as the loss function. The Adam optimizer with a very low learning rate of 0.0001 was used to optimize the training.

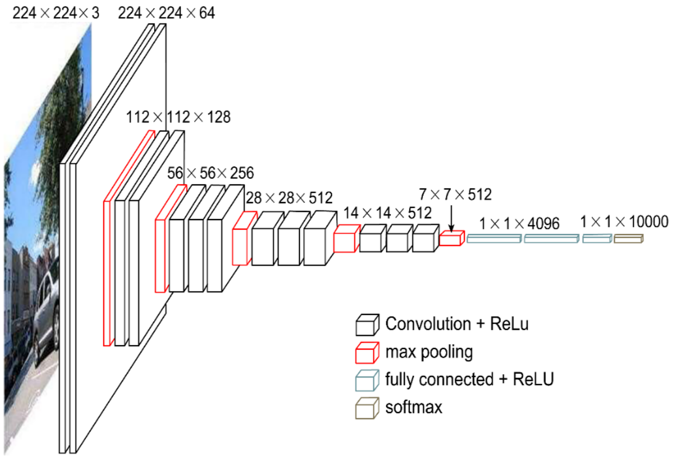

Fig 6. VGG-16 architecture

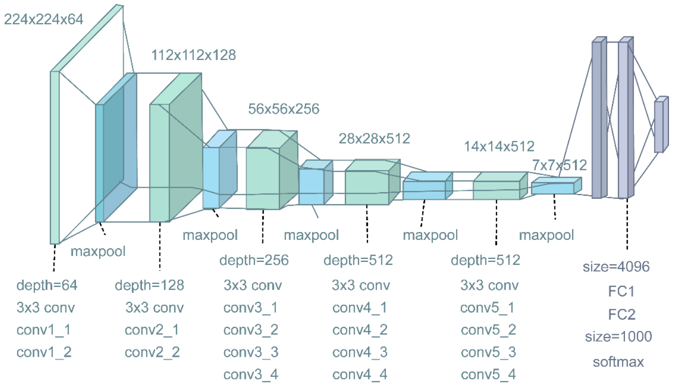

Fig 7. VGG-19 architecture

The step size is considered too small because larger values lead to divergence in the loss function instead of convergence. The plots of accuracy and loss for Inception-V3 are shown in Figure 8, those for VGG-16 in Figure 9, and those for VGG-19 in Figure 10.

Inception-V3 was trained with 4098 parameters, VGG-16 with 1029 parameters, and VGG-19 with 4096 parameters using transfer learning. During fine-tuning, Inception-V3 was trained for 29,630,466 parameters, VGG-16 for 11,800,066 parameters, and VGG-16 for 18,289,922 parameters for 20 epochs. The results obtained after fine-tuning were better than those obtained with transfer learning only because the weights of the model were re-trained with the current dataset and updated according to the current problem we are dealing with.

Table 2. Comparison of Proposed Classification Accuracies

Model	Fine-Tuning	Training Accuracy	Test Accuracy
Inception-V3	Not applied	87%	86%
Inception-V3	Applied	89%	**89%**
VGG-16	Not Applied	87%	86%
VGG-16	Applied	98%	**96%**
VGG-19	Not Applied	84%	83%
VGG-19	Applied	98%	**97%**

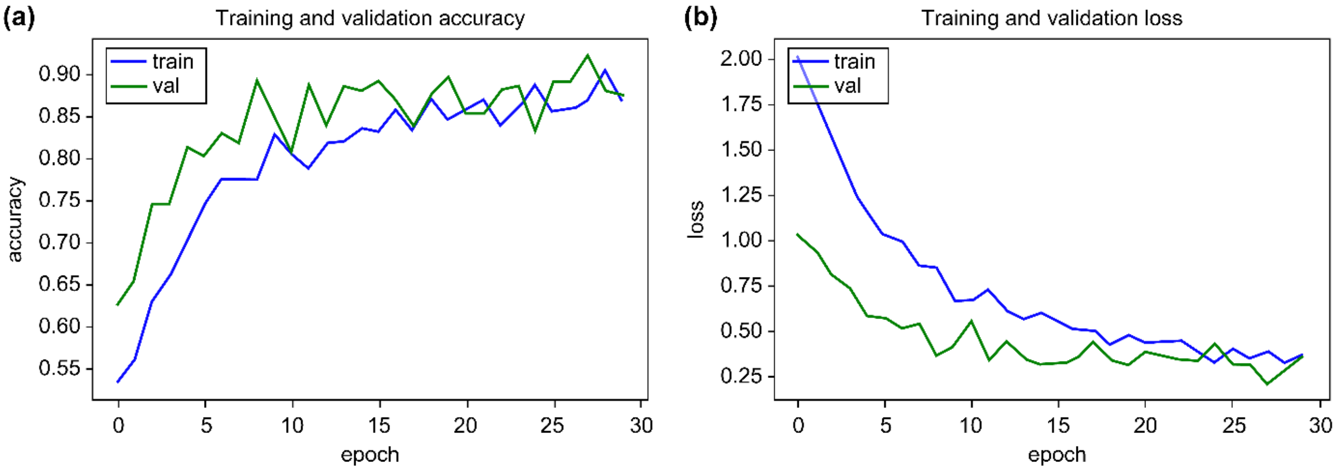

Fig 8. a) Accuracy plot of Inception-V3 b) Loss plot of Inception-V3

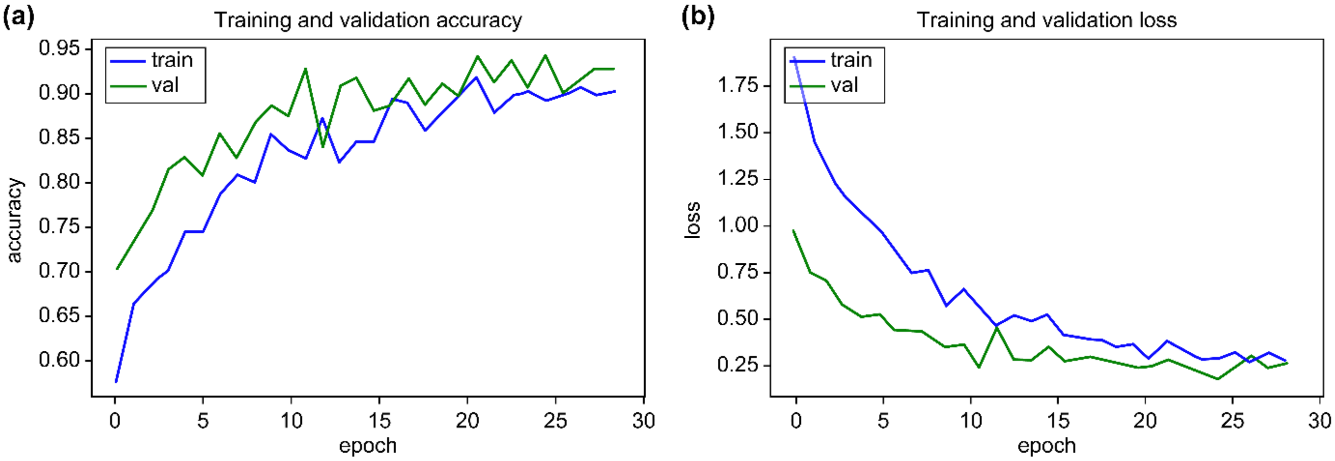

Fig 9. a) Accuracy plot of VGG-16 b) Loss plot of VGG-16

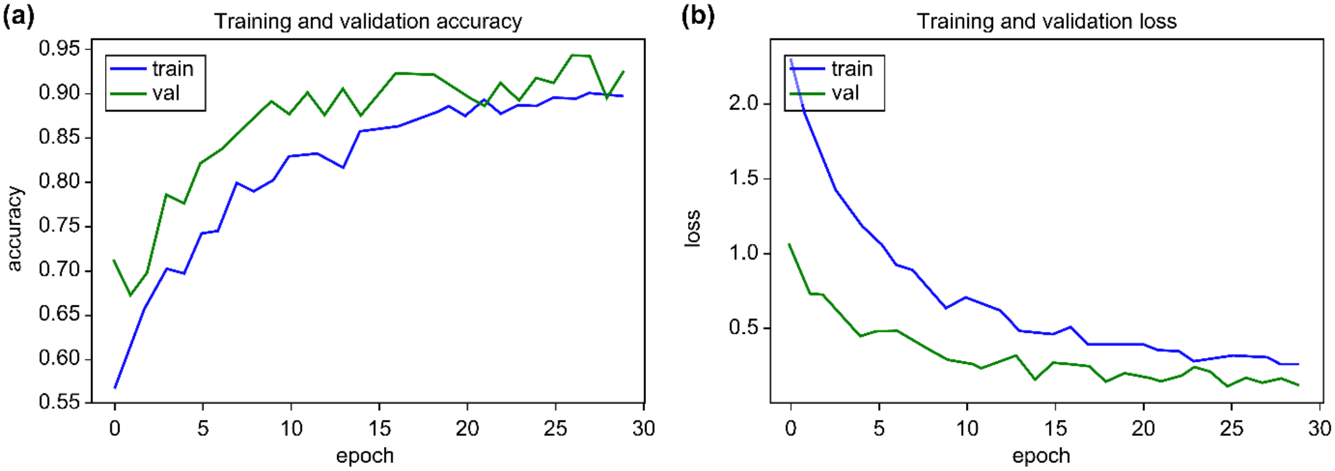

Fig 10. a) Accuracy plot of VGG-19 b) Loss plot of VGG-19

The results obtained with transfer learning and fine-tuning are presented in Table 2. Higher accuracy was observed in the experiments performed with fine-tuning than when transfer learning was used alone.

Inception-V3 achieved 87 % accuracy on the training set and 86 % with the test set on transfer learning. After fine-tuning the architecture, an accuracy of 89 % was achieved on the training set and 89 % on the test set, representing a 2 % increase in training set accuracy and a 3 % increase in test set accuracy. VGG-16 achieved 87 % accuracy on the training set and 86 % on the test set during transfer learning. After fine-tuning the architecture, an accuracy of 98 % was achieved on the training set and 96% on the test set, representing an 11 % increase in the accuracy of the training set and a 10% increase in the accuracy of the test set. VGG-19 achieved 84 % accuracy on the training set and 83% on the test set for transfer learning. However, after fine-tuning the architecture, it achieved 98 % accuracy on the training set and 97 % on the test set, representing a 14 % increase in accuracy in the training set and a 14 % increase in accuracy in the test set.

VGG-19 and Inception-V3 achieved similar accuracy in transfer learning, as they trained almost the same number of parameters. However, when fine-tuning the model, Inception-V3 must train a much larger number of parameters than VGG-19. Therefore, although the accuracies achieved by Inception-V3 and VGG-19 after transfer learning were similar, there was a significant difference in the accuracies they achieved after fine-tuning.

VGG-16 and VGG-19 achieved higher accuracies than Inception-V3 after fine-tuning because the efficiency of the optimizer is affected by the number of layers present in the network. The optimizer works better with networks with fewer layers than those with more layers. Because the optimizer is efficient with VGG-16 and VGG-19, they achieve better accuracy than Inception-V3.

Table 3. Comparison of Proposed Classification Accuracies with Existing Methods

Model	Accuracy (%)	Description
VGG-19	97.00	Trained on automatically extracted MRI features using transfer learning and fine-tuning 12 layers with learning rate=0.0001
VGG-16	96.00	Trained on automatically extracted MRI features using transfer learning and fine-tuning 9 layers with learning rate=0.0001
VGG-16 (2020)	96.00	Trained on a small dataset for 15 epochs using transfer learning
Inception-V3	89.01	Trained on automatically extracted MRI features using transfer learning and fine-tuning 155 layers with learning rate=0.0001
VGGNet -16 (2017)	83.86	Trained on Image-Net Dataset and tested on MRI slices 200×200
ResNet (2017)	84.91	Trained on Image-Net Dataset and tested on MRI slices 200×200
Inception-V3 (2020)	75.00	Trained on a small dataset for 15 epochs using transfer learning

Therefore, it can be stated that whenever the dataset is limited and contains an insufficient number of images to train a neural network, transfer learning can be used to achieve better accuracy in less time than training the model from scratch. Fine-tuning can then be used in conjunction with transfer learning to achieve higher accuracy, as the weights of the model are adjusted to fit the current problem.

5 Conclusion

In this study, we presented a solution to the computer vision problem. That is, we automated the detection of brain tumors in MRI images using CNNs and transfer learning. The deep architecture and shallow architecture of CNNs Inception-V3, VGG-16, and VGG-19 were used to extract features using transfer learning. Then, some weights were updated by fine-tuning. An accuracy of 89 % was achieved with Inception-V3, 96 % with VGG-16, and 97 % with VGG-19 on the experimental dataset. Another technique called data augmentation was used to reduce the likelihood of overfitting the dataset, because the dataset was small, and to solve the problem of biased datasets. Transfer Learning presents a novel method for analyzing data with few annotations by transferring knowledge from the source domain to the target domain. In the future, this technique can be advised for determining the size of a tumor so that its stage can also be determined. In addition, this model can be recommended to apply transfer learning to any other tumor-detection problem for which training data is lacking.

6 Data Availability

The data used to support the findings of this study are available at https://www.kaggle.com/ahmedhamada0/brain-tumour-detection.

References

1) Van Meir EG, Hadjipanayis CG, Norden AD, Shu HK, Wen PY, Olson JJ. Exciting New Advances in Neuro-Oncology: The Avenue to a Cure for Malignant Glioma. *CA: A Cancer Journal for Clinicians*. 2010;60(3):166–193. Available from: https://doi.org/10.3322/caac.20069.
2) Deangelis LM. Brain tumours. *New England Journal of Medicine*. 2001;344(2):114–123. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.3127&rep=rep1&type=pdf.
3) Scegedy C, Liu W, Jia Y. Going Deeper With Convolutions in Proceedings of the IEEE Conference on Computer Vision and Pattern. Available from: https://doi.org/10.48550/arXiv.1409.4842.
4) Prakash RM, Kumari RSS. Classification of MR Brain Images for Detection of Tumor with Transfer Learning from Pre-trained CNN Models. In: 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET). IEEE. 2019;p. 508–511. Available from: https://doi.org/10.1109/WiSPNET45539.2019.9032811.
5) Abdalla HEM, Esmail MY. Brain Tumor Detection by using Artificial Neural Network. In: 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). IEEE. 2018;p. 1–6. Available from: https://dx.doi.org/10.1109/ICCCEEE.2018.8515763.
6) Pereira S, Pinto A, Alves V, Silva CA. Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images. *IEEE Transactions on Medical Imaging*. 2016;35(5):1240–1251. Available from: https://doi.org/10.1109/TMI.2016.2538465.
7) Abd-Ellah MK, Awad AI, Khalaf AA, Hamed HF. Two phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks. *EURASIP Journal on Image and Video Processing*. 2018;2018(1). Available from: https://doi.org/10.1186/s13640-018-0332-4.
8) Noreen N, Palaniappan S, Qayyum A, Ahmad I, Imran M, Shoaib M. A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor. *IEEE Access*. 2020;8:55135–55144. Available from: https://doi.org/10.1109/ACCESS.2020.2978629.
9) Khan HA, Jue W, Mushtaq M, Mushtaq MU. Brain tumour classification in MRI image using convolutional neural network. *Mathematical Biosciences and Engineering*. 2020;17(5):6203–6216. Available from: https://doi.org/10.3934/mbe.2020328.
10) Amin J, Sharif M, Haldorai A, Yasmin M, Nayak RS. Brain tumor detection and classification using machine learning: a comprehensive survey. *Complex & Intelligent Systems*. 2022;8(4):3161–3183. Available from: https://doi.org/10.1007/s40747-021-00563-y.
11) Srivastava N, Hinton GE, Sutskever I. A simple way to prevent neural networks from over fitting. *Journal of Machine Learning Research*. Available from: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.
12) Dvorák P, Menze B. Structured prediction with convolutional neural networks for multimodal brain tumour segmentation, MICCAI Multimodal Brain Tumour Segmentation Challenge (BraTS). 2015;p. 13–24. Available from: http://people.csail.mit.edu/menze/papers/dvorak_15_cnnTumor.pdf.
13) Irsheidat S, Duwairi R. Brain Tumor Detection Using Artificial Convolutional Neural Networks. In: 2020 11th International Conference on Information and Communication Systems (ICICS). IEEE. 2020;p. 197–203. Available from: https://dx.doi.org/10.1109/ICICS49469.2020.239522.
14) Sravya V, Malathi S. Survey on Brain Tumor Detection using Machine Learning and Deep Learning. *2021 International Conference on Computer Communication and Informatics (ICCCI)*. 2021;p. 1–3. Available from: https://doi.org/10.1109/ICCCI50826.2021.9457019.
15) Dipu NM, Shohan SA, Salam KMA. Deep Learning Based Brain Tumor Detection and Classification. *2021 International Conference on Intelligent Technologies (CONIT)*. 2021;p. 1–6. Available from: https://doi.org/10.1109/CONIT51480.2021.9498384.
16) Gaikwad S, Patel S, Shetty A. Brain Tumor Detection: An Application Based on Machine Learning. *2021 2nd International Conference for Emerging Technology (INCET)*. 2021;p. 1–4. Available from: https://dx.doi.org/10.1109/INCET51464.2021.9456347.
17) Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. *Computer Vision and Pattern Recognition*. Available from: https://doi.org/10.48550/arXiv.1512.00567.
18) Varshney P. VGGNet-16 Architecture: A Complete Guide. *Computer Vision and Pattern Recognition*. Available from: https://www.kaggle.com/blurredmachine/vggnet-16-architecture-a-complete-guide.
19) Ullah N, Khan JA, Khan MS, Khan WA, Hassan I, Obayya M, et al. An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning. *Applied Sciences*. 2022;12(11):5645–5645. Available from: https://doi.org/10.3390/app12115645.
20) Khalid El Fazazy, Mohamed Adnane Mahraz, Hamid Tairi. *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML 2021)*;p. 566–571.
21) Amin J, Anjum MA, Sharif M, Jabeen S, Kadry S, Ger PM. A New Model for Brain Tumor Detection Using Ensemble Transfer Learning and Quantum Variational Classifier. *Computational Intelligence and Neuroscience*. 2022;2022:1–13. Available from: https://doi.org/10.1155/2022/3236305.
22) Raza A, Ayub H, Khan JA, Ahmad IS, Salama AS, Daradkeh YI, et al. A Hybrid Deep Learning-Based Approach for Brain Tumor Classification. *Electronics*. 2022;11(7):1146–1146. Available from: https://doi.org/10.3390/electronics11071146.
23) Charulatha G, Balaji B. Mobile Application to Detect Brain Tumor Using Transfer Learning". *Journal of Science, Computing and Engineering Research*. 2022;3(2):247–252. Available from: https://doi.org/10.46379/jscer.2022.030204.