BINARY CLASSIFICATION OF BRAIN MRI IMAGES FOR DETECTION OF BRAIN TUMOR USING CONVOLUTIONAL NEURAL NETWORK

Anika Bushra Chowdhury[1], Emrana Kabir Hashi[2]

[1]Rajshahi University of Engineering and Technology
[2]Rajshahi University of Engineering and Technology
E-mail(s): anikabushra6069@mail.com emranakabir@gmail.com

Abstract: *Classification of brain tumor is a critical assignment to assess the tu-mors and settle on a treatment resolution as per their category. Although there are numerous imaging methods used to identify brain tumors, nevertheless MRI is most frequently used. The popularity of MRI relies on the fact that it has a condescending image quality despite being non-invasive and not involving any ionizing radiation. In recent times deep learning has shown an extraordinary per-formance on classification problems. Acknowledging that, CNN (Convolutional Neural Network) a class of deep neural networks is used in this research for bi-nary classification of brain MRI images. The dataset includes 253 images in total, among which, 98 are non-tumor images and 155 are tumor images. The best results are shown by augmented dataset and pre-trained models. The proposed methodology uses 1 to 8 convolutional layers to spot the most accurate one. The best result was given by model with 5 layers which is 94.19% on unseen test data. Other evaluation metrics is also the highest for this model. The pre-trained models have rather finer results other than RseNet50 which has the lowest accuracy rate of 81.29%. VGG-16 has obtained 97.42%, and InceptionV3 has obtained 97.09% accuracy on test data.*

1. INTRODUCTION

The unusual and immoderate growth of abnormal brain cells is known as brain tumor. As the human skull is rigid and the volume is limited, sudden development may be extremely harmful. If the tumor spreads into other body organs, it might impair human functionality to a great extent [1]. There are two main varieties of brain tumors. One is non-cancerous and known as benign tumor. This type of tumor's growth is slow and they do not spread to other body organs. But the more harmful type is known as malignant tumor. They are cancerous and they grow abruptly. This type of tumor also tends to metastasize which means they spread to other sites in the body by metastasis [2].

According to the American Cancer Society in 2021 about 24,530 (13,840 in males and 10,690 in females) malignant tumors will be diagnosed and among them about 18,600 (10,500 males and 8,100 females) people will die from it. Cancer research corporation in the United Kingdom reported that there are about 5,250 deaths every year by the act of brain, other Central Nervous System (CNS), and intracranial tumors in the UK [3]. Magnetic Resonance Imaging (MRI) is a noninvasive diagnostic test that takes detailed images of the soft tissues of the body and it is widely used in brain tumor detection because of the fact that it uses no ionizing radiation during the scan [4].

Doctors or researchers use many different techniques to observe and classify brain images among them some popular methods are Support Vector Machine, Decision Tree, Linear Discriminant Analysis, Logistic Regression, etc. The main disadvantage of these methods is they all need pre-processing like normalization, intensity, shape, and texture feature extraction. Also, their performance was not up to the mark as the medical sector requires exceptionally high accuracy.

At present, Convolutional Neural Networks (CNN) a class of deep neural networks are becoming more and more popular as they can detect features without any human supervision. So, pre-processing has become rather an easy task compared to its predecessors. Furthermore, they are very efficient in image-oriented tasks and give higher accuracy than any other machine learning approach.

2. RELATED WORK

In recent times the increase in the use of deep learning in the medical sector is noteworthy. Authors Hassan Ali Khan, Wu Jue, Muhammad Mushtaq and Muhammad Umer Mustaq [5] proposed a CNN architecture with 8 convolution layers and using Brain MRI Images for Brain Tumor Detection by Novoneel Chakrabarty as the dataset, their proposed model had 100% accuracy while VGG16 had 96%, ResNet50 had 89% finally InceptionV3 had 75% accuracy. The paper authored by Priyansh Saxena, Akshat Maheshwari, Shivani Tayal, Saumil Maheshwari [6] applied transfer learning on the dataset by Novoneel Chakrabarty and got an accuracy of 95% for ResNet50, InceptionV3 suffered from overfitting with an accuracy of 55% as well as VGG16 has a result in between with an accuracy of 90%.

The authors Onkar Rajesh Mulay and Hemprashad Yashwant Patil [7] used the pre-trained AlexNet architecture but they used average pooling instead of max pooling layers, which amplified the result and gave an accuracy of 97.7%. In their paper authors Mesut Toğaçar, Burhan Ergen, Zafer Cömert proposed a new architecture named BrainMRNet [8], which achieved 96.05% accuracy whereas VGG16 has 84.48%, GoogleNet has 89.66% and AlexNet has 87.93%. In the paper authors Rayene Chelghoum, Ameur Ikhlef, Amina Hameurlaine and Sabir Jacquir. [9] have 9 pre-trained models among them VGG-16 had 98.71% accuracy, the highest. Authors M. A. Bakr Siddiaue, S. Sakib, M. M. Rahman Khan, A. K. Tanzeem, M. Chowdhury, and N. Yasmin [10] proposed a Deep Convolutional Neural Network (DCNN) model which got 96% accuracy, 93% precision, 100% sensitivity and 97% F1-score.

In their paper authors, A. Kumar Pandey and K. C. James [11] applied five different classification techniques including CNN, SVM, Decision Tree, Linear Discriminant Analysis, Logistic Regression as their objective was to review different types of MRI brain tumor classification techniques and compare their performance measures subsequently they concluded that the best model to use for brain tumor detection is Convolutional Neural Network.

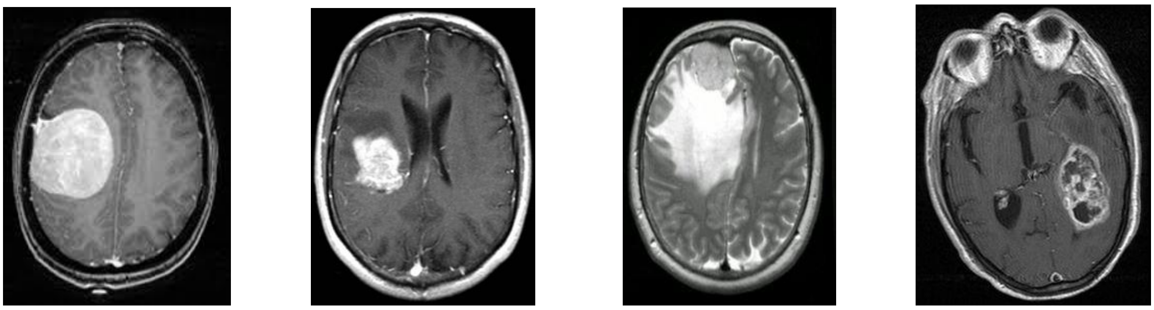
Author Donghyu Kim [12] used transfer learning to propose 2 methods of classification in his paper, where the first method combining VGG-16 and Resnet-50 achieved test accuracy of 98% and the second method was an extended convolutional neural network with an accuracy of 86.30%. In the paper authored by Sultan, Hossam H.Salem, Nancy M.Al-Atabany,

Walid [13] proposed model first got 96.13% accuracy classifying among Meningioma, Glioma, Pituitary tumors and then using the same model on a different dataset they detected if the tumor is Grade II, III or IV with an accuracy of 98.7%.

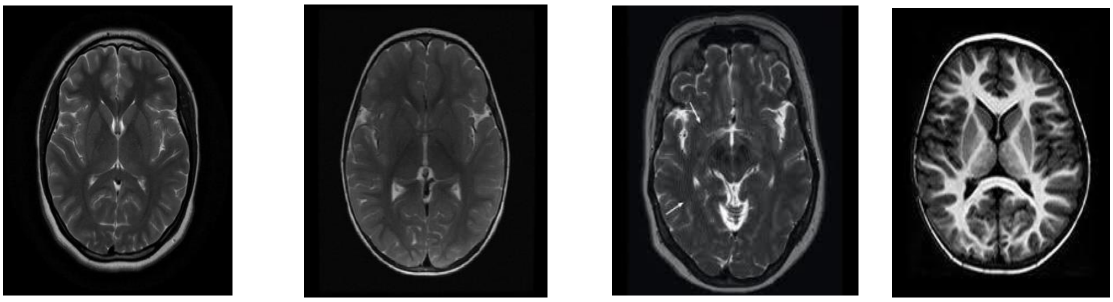
3. METHODOLOGY

3.1 Dataset

The title of the dataset used is Brain MRI Images for Brain Tumor Detection. It consists of free accessible MRI images. It has 253 images in two folders named yes and no. The no folder has 98 images of healthy brain with no tumors and the yes folder has 155 images with brain tumors [14].



(a) Images containing brain tumor



(b) Images with no brain tumor

Figure 1. Sample images from the dataset.

3.2 Data Preprocess

Data preprocessing is an important data mining technique. It is of great importance while working with convolutional neural networks as it makes the data more interpretable and prevents bad classification performances. In data preprocessing the raw images are modified in a manner that they are more acceptable and fitted for succeeding operations.

For this research, the images are resized and crop normalization is done as pre-processing operation. In the input dataset, images have different dimensions and aspect ratios. So, they are all resized to a preset format, which is (240,240,3) = (image_width, image_height, number of channels) before feeding into the architecture.

After resizing, crop normalization is used to determine the extreme points of an image. It helps to get the portion of the image only containing the brain so that the CNN model only concentrates on the subject and avoids unnecessary information as well as noises. This method is used to determine the farthest north, south, east, and west (x, y)-coordinates along a given contour. This technique applies to both raw contours and rotated bounding boxes.

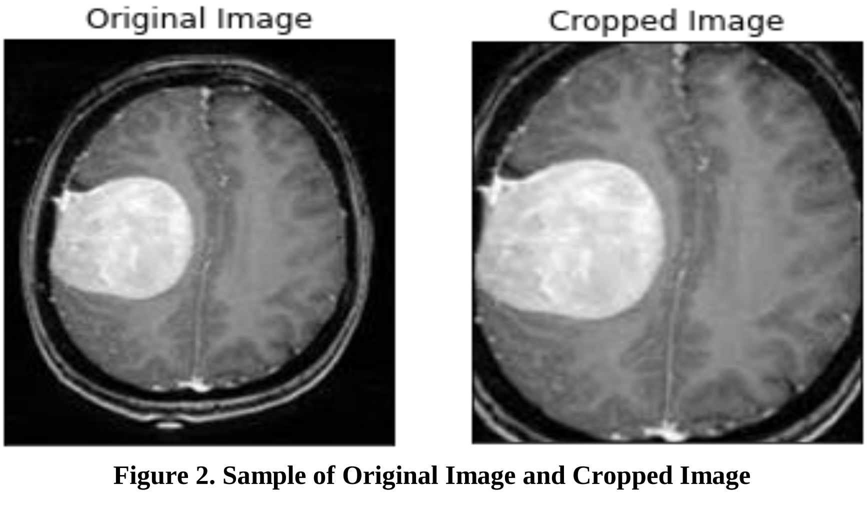


Figure 2. Sample of Original Image and Cropped Image

3.3 Data Augmentation

Deep neural networks provide a significant boost in performance and produce skillful models whenever trained on more data. So, in order to boost the performance of the models‘ data is augmented. After augmentation, the dataset has 2065 images among which 1085 are positive examples and 980 negative examples. Also, the data is more balanced subsequent augmentation.

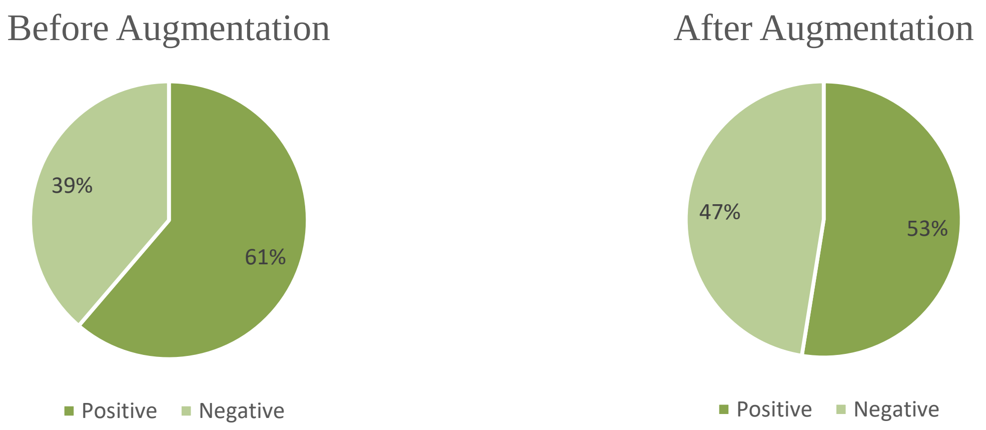


Figure 3. Before and after augmentation

3.4 Data Split

In this step, the complete dataset is divided into three segments, namely, Train, Test, and Validation. 70% of data is kept as training data to train the model. Another, 15% are kept as validation data and the remaining 15% are used to test the accuracy of the model. That means 1445 images are for the training set, 310 images for the validation set, and 310 images for the test set.

3.5 CNN Model

In the proposed CNN model, the size of an input image is (224,224,3). Different numbers of convolutional and max pooling layers are used and observed to obtain the most accurate one. The deepest architecture has 8 convolutional layers as well as 4 max pooling layers. The shallowest model has only 2 layers other than dense layers. They are one convolutional layer and one max pooling layer. In between them, six more models are trained and tested with different numbers of convolutional layers and max pooling layers.

The number of filters is doubled for every set of convolutional layer. It starts with 16 and then gradually increases to 32, 64 and finally 128. The filter size is always (3,3) and the stride for the convolutional layer is 1. The non-linearity used in convolutional layers is the ReLU activation function.

The max pooling layers have (2,2) window size and the stride is equal to 2. The height and width of the image are reduced by a factor of two in every pooling layer. So, as the model becomes deeper the height and width of the image decreases while the number of channels increases because of the filters.

After that, the 3-dimensional matrix is flattened into a one-dimensional vector and then it goes through 3 dense layers. The first two layers have ReLU as an activation function. But the finishing layer is a dense fully connected layer with a sigmoid activation which gives a result between 0-1 and helps to reach conclusions in binary classification.

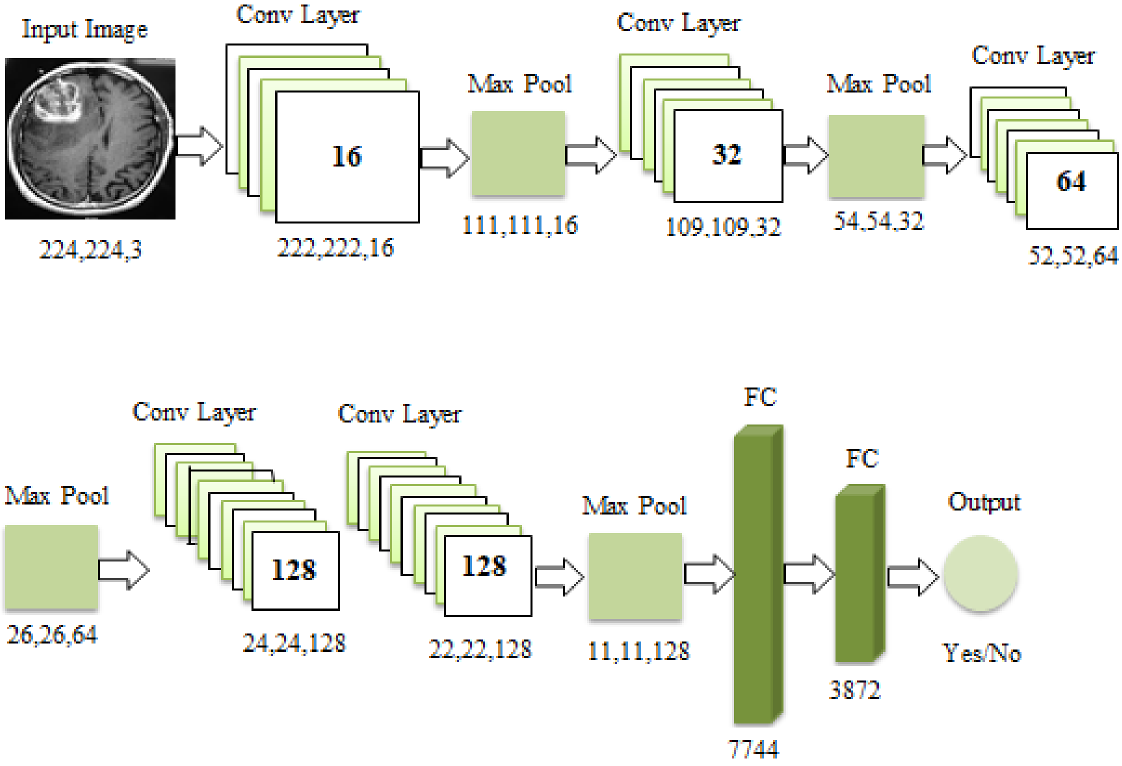


Figure 4. Proposed CNN Architecture.

3.6 Transfer Learning

Transfer learning is a well-known machine learning approach. In transfer learning a model trained and developed for a task is reapplied on a different task. The huge amount of compute and time resources necessary to develop a model is being reused due to it. Keras is a widely used open-source library for ANN. It provides a number of pre-trained models to use for prediction, feature extraction, and fine-tuning [15]. Among them VGG-16, InceptionV3, and ResNet50 are often used in brain tumor researches.

3.6.1 VGG16

VGG-16 is a convolutional neural network model proposed by authors K. Simonyan and A. Zisserman from the University of Oxford [16]. The remarkable thing about this model is it does not use too many hyperparameters. The convolutional layers always have (3,3) filters with the same padding and stride 1. And the pooling layers have a window size of (2,2) with a stride of 2. The principle of this model was doubling the filter numbers for every set of convolutional layers and reducing the height and width of the image by a factor of two in every pooling layer. After the convolution and pooling layers, the dimension becomes (7,7,512), which then goes through 3 fully connected layers and a softmax layer. The name of the architecture has 16 in it because it has 16 layers that have weights. It is a large network with almost 138 million parameters [17] yet its simplicity is noteworthy.

3.6.2 InceptionV3

The most recent used inception architecture was first introduced by Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna in their paper [18]. Inception cell is the basic building block of this model. In an inception cell, a number of convolutions are performed at the same time and consequently totaled the outcomes. Here features are extracted at different scales from the input by using filters of different sizes like (1,1), (3,3), and (5,5). In this way, as opposed to picking a clear filter size and working with it, inception works with every option possible. It doesn't miss out on anything and that is the best advantage of it. The primary version V1 has 5 million parameters and the latest version of inception which is V3 has 23 million parameters and the layer depth is 159 [19].

3.6.3 ResNet50

The introduction of ResNet was groundbreaking work as it introduced residual blocks [20]. It was conventional for neural networks that each layer gets the previous layer's output as its input. But with networks having residual blocks each layer feeds into the next layer and also into the layers which are 2-3 jumps ahead of it. ResNet50 has 50 layers and about 26 million parameters [21]. Here the convolutional layers are outlined in 5 stages and both max pooling and average pooling are used.

4. EXPERIMENTAL RESULTS AND ANALYSIS

This experiment covers a huge area. From simple CNN with a single layer of convolution to very deep CNN with 159 layers all are fed the same data and observed. They were evaluated on how they work on unseen test data. The proposed architecture was tuned with convolutional layers and max pooling layers keeping the number of fully connected layers the same. The model with 1 convolutional layer had 81.22% accuracy on test data. Every evolution metric showed increment with the additional layers until the number of layers was 5. After that, the downfall was noticeable so the tuning process was stopped.

Table 1. Performance of different CNN models

Convolutional/ Pooling layer	Train Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1 Score
1/1	100	83.55	81.22	81.57	80.51	81.04
2/2	100	84.52	87.09	88.16	88.69	88.42
3/3	100	84.84	87.70	87.20	90.36	88.75
4/4	100	92.90	92.26	91.77	92.95	92.35
5/4	99.93	93.23	95.79	95.40	97.08	96.23
6/4	100	90.00	91.29	93.33	89.17	91.21
7/4	54.32	53.87	52.90	52.90	100	69.19
8/4	57.11	47.10	49.68	49.68	100	66.38

The best performance was given by the model with 5 convolutional layers and 4 max pooling layers. The test accuracy was 95.79%, precision was 95.40%, recall was 97.08% and F1 score was 96.23%. Although it showed overfitting as the train accuracy was 99.93% whereas the accuracy for validation set was 93.23%.

In the case of CNN architectures, added layers mean extracting more features and getting better results but up to a certain extent. There is a limit to the number of layers and after the threshold is crossed instead of extracting features, data tend to overfit. Overfitting causes a decrease in accuracy and gives errors like false positives. For this reason, the proposed architecture performs well until the fifth convolutional layer and after that, the performance decreases and more error or loss is noticed.

Rather than them, the pre-trained models had better results except ResNet50. The best result was VGG-16 with a test accuracy of 98.71%. But it was very time-consuming and computationally expensive. Precision, recall, and f1 measure metrics were also satisfactory as they were all above 98%. Although InceptionV3 had a train accuracy of 99.72%. But overfitting happened and the validation accuracy dropped to 96.45%. Then again the test accuracy was a little better which was 98.06%. Other evolution metrics were very close to VGG16.

The worst test accuracy was by ResNet50. But it was the only model that had better accuracy for the test set than the training set. It had a train accuracy of 72.81% whereas the test accuracy was 81.29%. Therefore, it did not show any overfitting.

Table 2. Accuracy of pre-trained CNN models

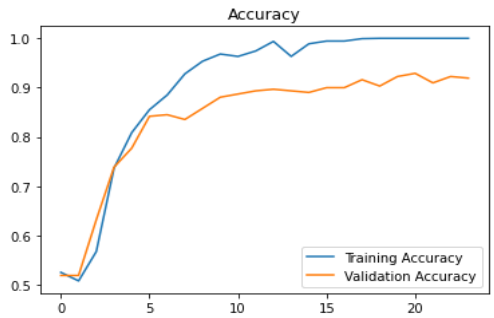
CNN Architecture	Train Accuracy	Validation Accuracy	Test Accuracy	Precision	Recall	F1 Score
VGG16	99.86	98.39	98.71	99.36	98.10	98.73
InceptionV3	99.72	96.45	98.06	98.82	97.66	98.23
ResNet50	72.81	82.58	81.29	76.64	87.07	81.53

Table 3. Different hyper-parameters used to reach the final results

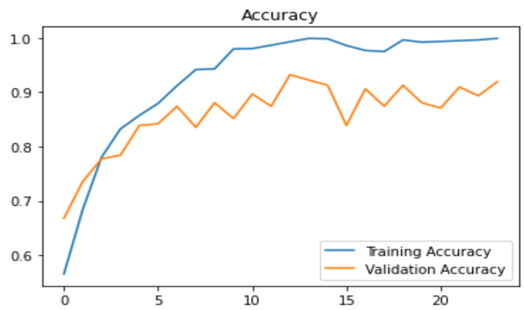
Factor	Values
Mini-batch size	32
Maximum epochs	10, 24
Optimizer	ADAM
Learning rate	0.001
Dropout rate	0.5

Table 4. Comparison of accuracy between this research and previous related work

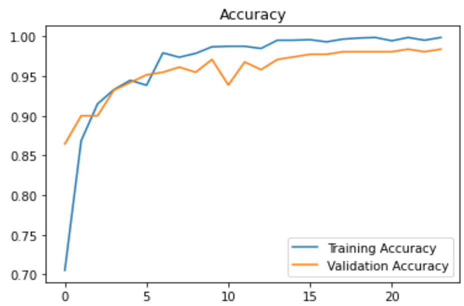
	Proposed Model	VGG16	InceptionV3	ResNet50
Hassan Ali Khan et al.[5]	100%	96%	75%	89%
Priyansh Saxena et al.[6]	N/A	90%	55%	95%
Mesut Toğaçar et al.[8]	96.05%	84.48%,	N/A	N/A
This research	94.52%	97.42%	97.09%	81.29%



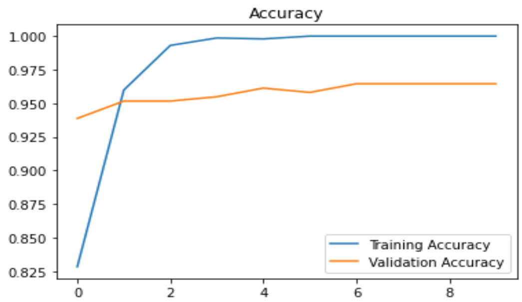
(a) Model trained with 4 convolutional layers



(b) Model trained with 5 convolutional layers

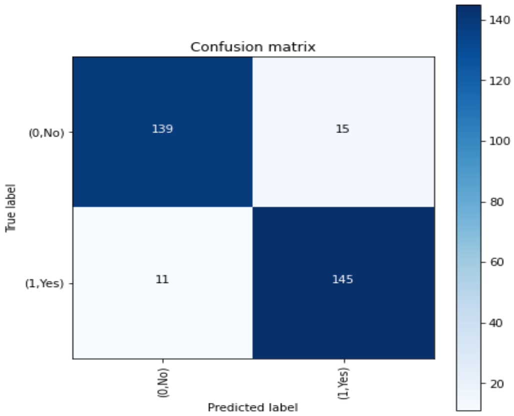


(c) Pre-trained model VGG16

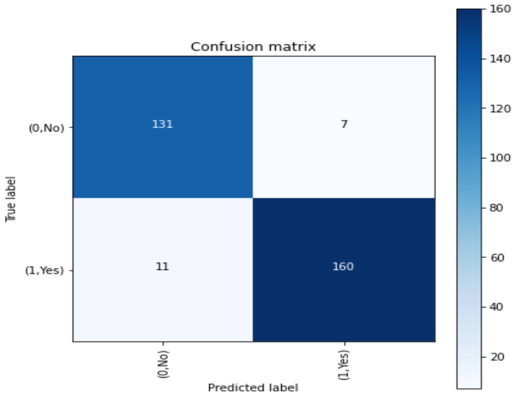


(d) Pre-trained model Inception

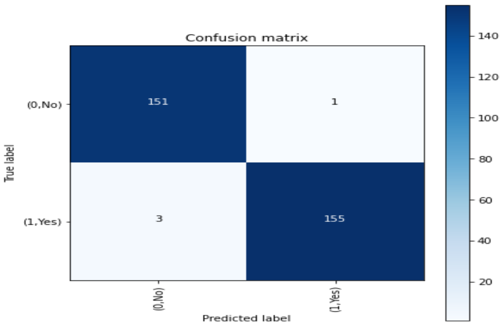
Figure 5. Accuracy graph of some of the models with highest test accuracy(%)



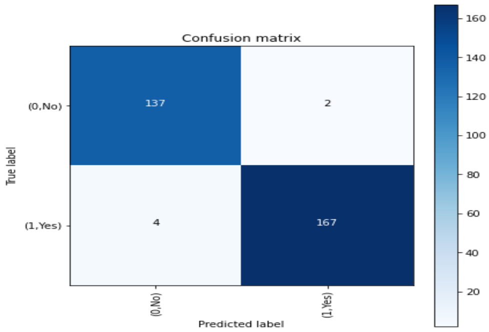
(a) Model trained with 4 convolutional layers



(b) Model trained with 5 convolutional layers



(c) Pre-trained model VGG16



(d) Pre-trained model Inception

Figure 6. Confusion Matrix of some of the models with highest test accuracy(%)

5. CONCLUSION

This research aimed to learn how different CNN models both simple architecture and complex architecture behaves on the same dataset. Often small datasets perform much better on basic architectures with a small number of layers than deep neural networks with numerous layers. But in the case of this dataset, the pre-trained models showed better results.

Tuning with a different numbers of convolutional layers was done but as the results were dropping so it was stopped after training the model with eight convolutional and four max pool layers. As the layers of the model increased, so did the over-fitting. The architecture modeled on the training data so well that it could not perform well on the unseen test data as it picked and learned the noise or random fluctuations in the training data too.

The best accuracy was achieved by the VGG16 model which is 98.71%. But the time for per epoch was also very high and that is 9.05 minutes. On the other hand, InceptionV3 had obtained 98.06% accuracy by only taking 5.45 minutes per epoch. The proposed model with 5 convolutional layers took 2.48 minutes per epoch and achieved 95.79%. Thereby, it reaches a moderate accuracy within a brief amount of execution time. Therefore, the most convenient model to use among the models trained for this research would be InceptionV3 since it has reached the most accuracy in the shortest amount of time.

However, some limitations should be noted which would be addressed in future research. The proposed model was only trained and tested on a small dataset but SMOTE, MICE or ADASYN technique was not used. So it would be done in future research. The models would be tuned with a bigger dataset and also be trained as well as tested with other brain MRI datasets. Here percentage split method was used to train the data. Cross-validation would be applied. Also, the dataset would be trained with other pre-trained models such as AlexNet, GoogleNet, etc. In this study, some models showed a lot of over-fitting and necessary steps would be taken to reduce it.

REFERENCES

[1] L. M. DeAngelis, “Brain Tumors,” *http://dx.doi.org/10.1056/NEJM200101113440207*, vol. 344, no. 2, pp. 114–123, Aug. 2009, doi: 10.1056/NEJM200101113440207.

[2] A. Patel, “Benign vs Malignant Tumors,” *JAMA Oncol.*, vol. 6, no. 9, pp. 1488–1488, Sep. 2020, doi: 10.1001/JAMAONCOL.2020.2592.

[3] “Brain, other CNS and intracranial tumours incidence statistics | Cancer Research UK.” https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/brain-other-cns-and-intracranial-tumours/incidence (accessed Jul. 12, 2021).

[4] “MRI | CancerQuest.” https://www.cancerquest.org/patients/detection-and-diagnosis/magnetic-resonance-imaging-mri (accessed Jul. 12, 2021).

[5] H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, “Brain tumor classification in MRI image using convolutional neural network,” doi: 10.3934/mbe.2020328.

[6] P. Saxena, A. Maheshwari, S. Tayal, and S. Maheshwari, “Predictive modeling of brain tumor: A Deep learning approach,” *Adv. Intell. Syst. Comput.*, vol. 1189, pp. 275–285, Nov. 2019.

[7] “Transfer Learning for Classification of 2D Brain MRI Images and Tumor Segmentation,” doi: 10.35940/ijrte.F8321.038620.

[8] M. Toğaçar, B. Ergen, and Z. Cömert, “BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model,” *Med. Hypotheses*, vol. 134, p. 109531, Jan. 2020, doi: 10.1016/j.mehy.2019.109531.

[9] R. Chelghoum, A. Ikhlef, A. Hameurlaine, and S. Jacquir, “Transfer learning using convolutional neural network architectures for brain tumor classification from MRI images,” in *IFIP Advances in Information and Communication Technology*, Jun. 2020, vol. 583 IFIP, pp. 189–200, doi: 10.1007/978-3-030-49161-1_17.

[10] M. A. Bakr Siddiaue, S. Sakib, M. M. Rahman Khan, A. K. Tanzeem, M. Chowdhury, and N. Yasmin, “Deep Convolutional Neural Networks Model-based Brain Tumor Detection in Brain MRI Images,” in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Oct. 2020, pp. 909–914, doi: 10.1109/I-SMAC49090.2020.9243461.

[11] A. Kumar Pandey and K. C. James, “A Review of Different Classification Techniques used in Brain Tumor Detection.”

[12] D. Kim, “BRAIN TUMOR DETECTION: 2 NOVEL APPROACHES,” Aug. 2020, doi: 10.20944/preprints202008.0641.v1.

[13] H. H. Sultan, N. M. Salem, and W. Al-Atabany, “Multi-Classification of Brain Tumor Images Using Deep Neural Network,” *IEEE Access*, vol. 7, pp. 69215–69225, 2019, doi: 10.1109/ACCESS.2019.2919122.

[14] “Brain MRI Images for Brain Tumor Detection | Kaggle.” https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection (accessed Dec. 13, 2020).

[15] “Keras Applications.” https://keras.io/api/applications/ (accessed Jul. 12, 2021).

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” Sep. 2015, Accessed: Dec. 07, 2020. [Online]. Available: http://www.robots.ox.ac.uk/.

[17] “VGG16 and VGG19.” https://keras.io/api/applications/vgg/#vgg16-function (accessed Mar. 25, 2021).

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2818–2826, Dec. 2015, Accessed: Jul. 12, 2021. [Online]. Available: https://arxiv.org/abs/1512.00567v3.

[19] “InceptionV3.” https://keras.io/api/applications/inceptionv3/ (accessed Dec. 07, 2020).

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016. Accessed: Dec. 07, 2020. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/.

[21] “ResNet and ResNetV2.” https://keras.io/api/applications/resnet/#resnet50-function (accessed Mar. 25, 2021).