(7088CEM)
Artificial Neural Networks

Brain MRI Images for Brain Tumor Detection

Priyanka Nelli Komath, Student ID: 11979142, nellikomap@coventry.ac.uk
Faculty Of Engineering, Environment And Computing, Coventry University
MSc Data Science And Computational Intelligence
Coventry, United Kingdom

Abstract: A brain tumor is the development of brain cells that multiply in an uncontrollable and abnormal way. There are two types of brain tumors benign or non-cancerous and malignant or cancerous tumors, which are graded based on how frequently they return after the treatment and how quickly they are growing and spreading [1]. In this coursework, I am using the MRI image set to train the selected 3 models and to classify the images as tumorous or non-tumorous. There are different data preprocessing techniques and 3 different classification methods used in this project to train and test chosen data set. They are (1) convolutional Neural Network (CNN), VGG-19 and Inception V3 classification. The performance of all these methods are evaluated and compared by plotting 'accuracy' and 'value loss'. After analysing result parameter it is found that the best model to classify brain MRI image is VGG-19 method with 98% accuracy and 0.05 value loss.

Keywords: Artificial Neural Network, Brain Tumor, Convolutional neural Netwok, CNN, VGG19, Inception V3, prediction, One hot encoding, image resizing, Augmentation

I. INTRODUCTION

Brain tumor is a dangerous disease and are cause by the uncontrolled growth of brain cells. There are two forms of tumors; cancerous and non-cancerous. Brain tumor is one of the major and most common disease that increases the mortality rate across the world. Cancer death rate is increasing rapidly every year. According to world health organization (WHO) brain tumor is detected in 2.5 lakh people every single year globally which contributes to the 2% of variations of cancers detected globally. Each year, more than 4,200 people in the UK receive a brain tumour diagnosis (2007 estimates).

Cancerous tumor damages the brain lobes which results in inappropriate behaviour, depression and anxieties like psychological symptoms. As a result of all these changes one can have severe damaging effect in the personal life such as unstable relationships, unemployment etc.

One of the greatest tools available now for diagnosing brain tumours is magnetic resonance imaging (MRI), which produces more precise images. Automatic flaws identification in MRI is highly helpful to detect additional growth in the cells. One imaging technique that aids academics and medical professionals in studying the brain non-invasively is magnetic resonance imaging (MRI).

There are multiple neural network methods used to predict the brain tumor using MRI scan image; Convolution neural network, VGG-19, Inception-V3 and RESNET etc. In this study, the prediction of brain cancer is carried out using 5 different neural network models. Models are evaluated with the raw image data and then with the augmented image data. The best model with high accuracy and low loss values are discussed in this research paper.

II. THE DATASET

The dataset used for this course work is extracted from the Kaggle repository which was originally collected from the google image repository. Dataset has 213 images, which are stored in 2 different folders. The first data folder is named as ’Yes’ and it contains 155 images that have the tumor and the second folder with the name ‘No’ consists of 98 brain MRI image with no indication of a tumor.

Link to the dataset: https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

Figure 1 shows the sample MRI image of a tumorous brain and Figure 2 shows the sample MRI image of a non-tumorous brain.

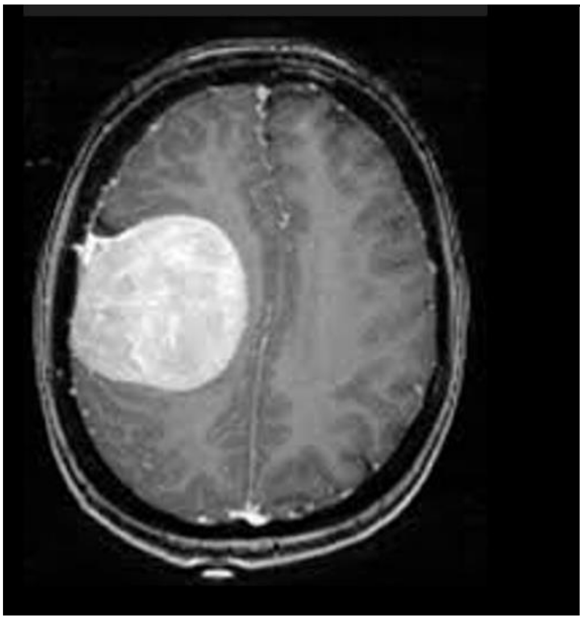

Fig 1: Tumorous Brain Image

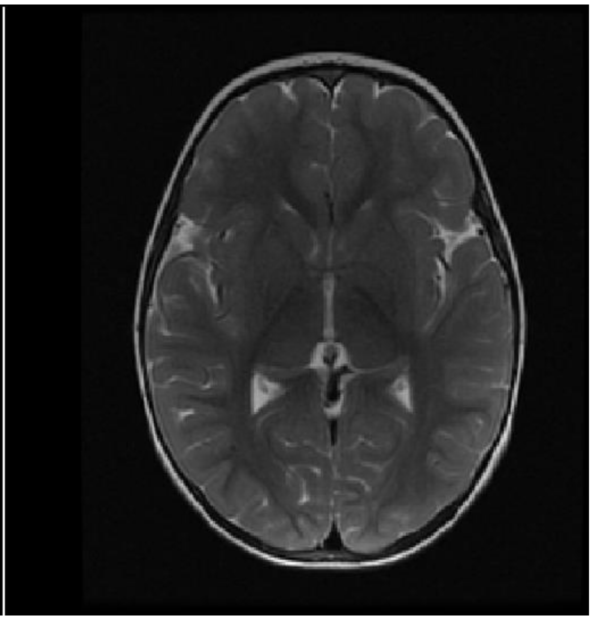

Fig 2: Non-Tumorous Brain Image

III. DATA ANALYSIS

The data analysis process is used to understand the input data for better data preparation for the project.

There are 253 brain MRI images used for the analysis in this project. These 253 images are tagged as ‘yes’ or ‘no’ to identify it as tumorous or non-tumorous. There are 98 tumorous images and 155 non-tumorous images used for this study.

After analysis, it is found that the images are of different sizes and resolutions, so we have to do adjustments to make all the images into a similar resolution and size for the ease of processing the data.

Image label indicates whether is tumorous or non-tumorous and string ’yes’ or ‘no’ is used for it. Since these are not numbers we have used one-hot encoding to convert the labels to 0 or 1.

Distribution of the tumorous and non-tumorous image classes are represented as below. Code written to generate the plot is given in the link attached in the appendix.

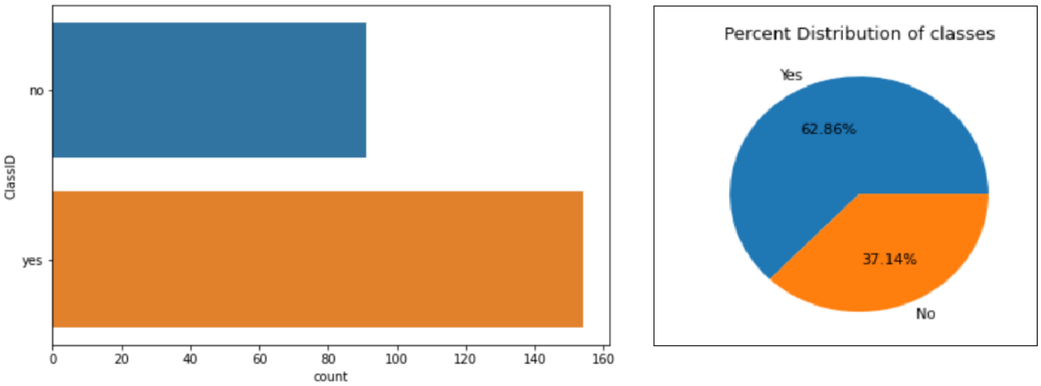

Fig 3: Distribution of tumorous and non-tumorous images

IV. METHODOLOGY

This research paper discusses on the different methods used for the prediction of tumor using brain MRI scanned image. The data set consists of 253 images and the result of the MRI scan is labelled in each image as ‘yes’ or ‘no’ representing if the patient has a tumor in their brain or no.

There are 3 different neural network models used in this project.

- Convolution neural network (CNN) using Keras
- VGG-19
- INCEPTION V3

V. EXPERIMENTAL SETUP

After data analysis, necessary data readiness or pre-processing steps were implemented to make the data ready in the acceptable format for the neural network models.

(A) DATA-PREPROCESSING

(1) Image Re-sizing

The data set has 253 images, and all the images are of different sizes. It is important to address this and make all the images to a standard size as convolutional neural networks need similar image size to work properly. In this research, I have resized all the images to a size of 224 X 224 for the VGG-19 model, 299 X 299 for the Inception V3 model and 125 X 125 for the CNN model using keras.

(2) One hot encoding

The images are labelled as ‘yes’ for a tumorous brain image and ‘no’ for a non-tumorous image. In this project we are training the neural network models using supervised learning by passing the labelled input to the chosen model, and it gives us the predicted output.

We are using one-hot encoding technique here to convert labeled values ‘yes’ and ‘no’ to 1 and 0 respectively

(3) Image Augmentation

Image augmentation is a method to artificially increase the size of the training dataset by generating an altered version of available images in the chosen dataset. It is used to improve the chosen model ability to generalize their learning to the newly generated images which will result in the improvement of model performance.

In this research, I have used 'ImageDataGenerator' class of 'The Keras deep learning neural network' library. I have used the horizontal flip method, in which the randomly selected images will be horizontally flipped and used for training the model.

(B) IMPLEMENTATION METHODS

In this research, python language is used to implement the selected neural network algorithms and the IDE used for the implementation is Google Colab. Original dataset extracted from Kaggle repository was uploaded in google drive. We have mounted the drive in the google colab in order to use and process the uploaded dataset. Data pre-processing steps were implemented after the dataset was successfully accessed from the mounted drive. I have used one hot encoding to convert categorical images labels from 'yes', 'no' to 1 and 0 respectively.

After pre-processing has completed, images stored in the two different folders were converted into numpy format as we cannot feed our models in a raw image format, and then stored into a data list to carry out the further steps.

In the next step, data was split into training and testing set with a ratio of 80:20. All 3 required models were created after the data split and validated the model performance using the test data set.

3 different neural network models were used in this research, they are:

(1) Convolution Neural Network (CNN) using Keras:

The first model I have used for the implementation is the 'convolution neural network' which is abbreviated as CNN. It is a subset of deep learning techniques that have gained prominence in a number of computer vision applications and are generating attention in many different fields, including radiology. CNN is made of multiple layers and is designed to adaptively and automatically learn spatial feature hierarchies using the algorithm known as backpropagation.

The architecture of the CNN algorithm is as given below.

Convolution layers, pooling layers (such as max pooling), and fully connected (FC) layers are some of the building pieces that make up a CNN.

<u>Convolutional layers:</u> The first layer in the architecture, is used to extract the different features from the input photo. Convolution is a mathematical process that is carried out at this layer between the input image and a filter of a specific size.

<u>Pooling Layer:</u> This is the second layer in the CNN architecture. This layer's main goal is to lower the convolved feature map's size in order to save on computational expenses.

<u>Fully Connected Layer:</u> This is the layer placed before the output layer which consists of weights and biases which is basically used to connect the neurons.

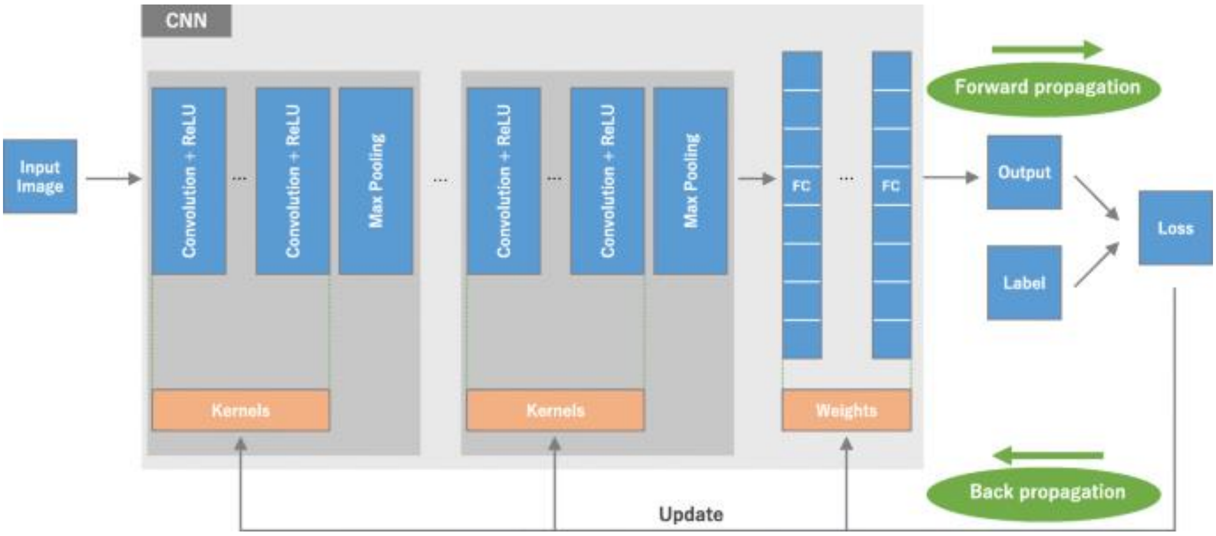

Fig 4: Architecture of CNN method

The model's performance under specific kernels and weights is calculated with a loss function through forwarding propagation on a training dataset.

To implement this model, I imported the necessary libraries and then loaded the dataset folder. Images that were stored in two different folders were then combined in a single list in NumPy format and applied all necessary pre-processing steps.

Figure 5: Value loss and accuracy were calculated and plotted after the training data was fit into the model.

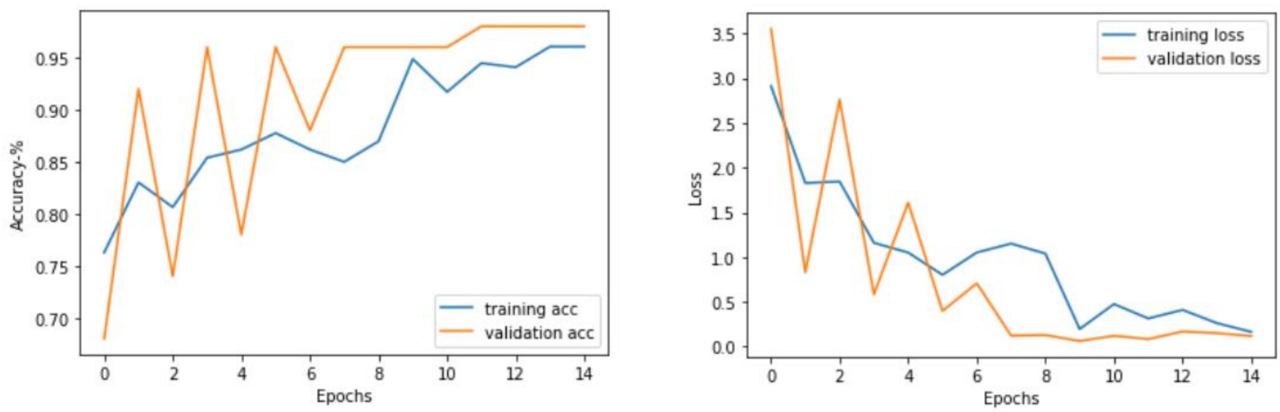

Figure 5

Figure 6: Random individual images were then passed through the model to check the accuracy and the result was:

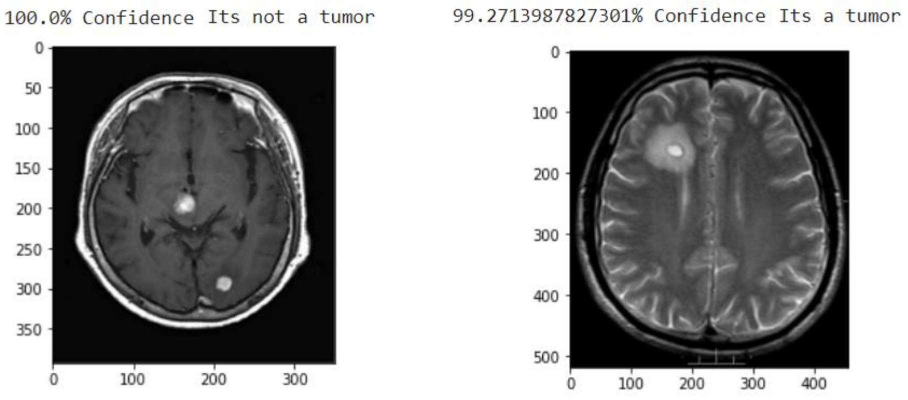

Figure 6

Code written to create, test the model and the result generation is given updated the link attached in the appendix.

(2) VGG-19:

The second model I have used for the implementation is 'VGG-19' model. The ImageNet database's VGG-19 convolutional neural network was trained using more than a million photos. The 19-layer network can categorize photos into 1000 different object categories, including a keyboard, mouse, pencil, and numerous animals. The network has therefore acquired rich feature representations for a variety of images.

The architecture of the VGG-19 method is as given below.

To implement this model, I imported the necessary libraries and then loaded the dataset. All necessary pre-processing steps were applied to the loaded data set. The training set was then fit into the model and have used 25 epochs for a full training dataset.

Then carried out the augmentation technique, for which I used horizontal flip method and generated random images keeping the original set as the base. Data set generated after the augmentation was then fit to the model and generated the results.

Training data was fit into the model, and value loss and accuracy were calculated and plotted, and compared before and after the augmentation technique.

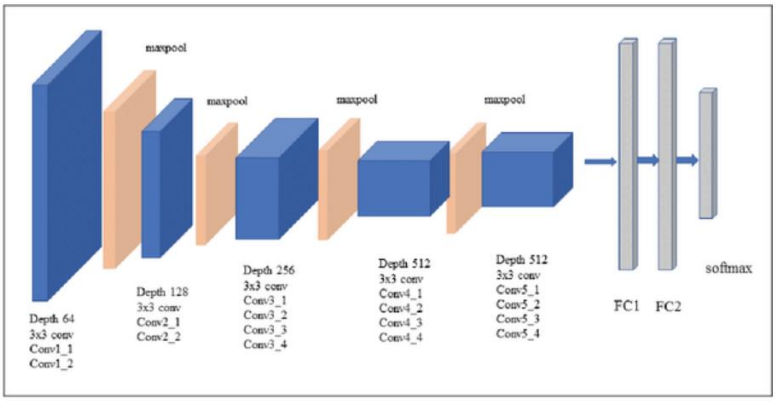

Figure 7: Architecture of VGG-19 model

Figure 8: Accuracy and value loss before the augmentation technique was implemented in the dataset.

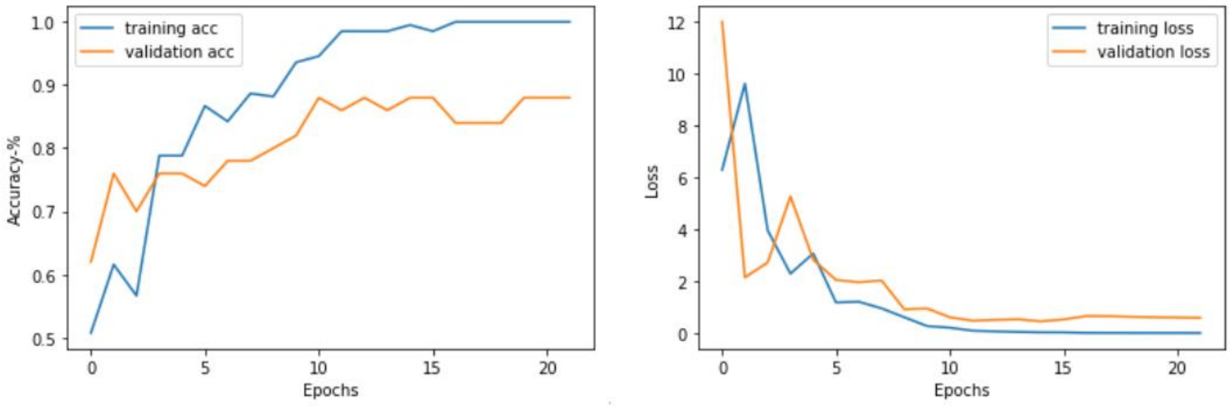

Figure 8

Figure 9: Accuracy and value loss after the augmentation technique was implemented in the dataset.

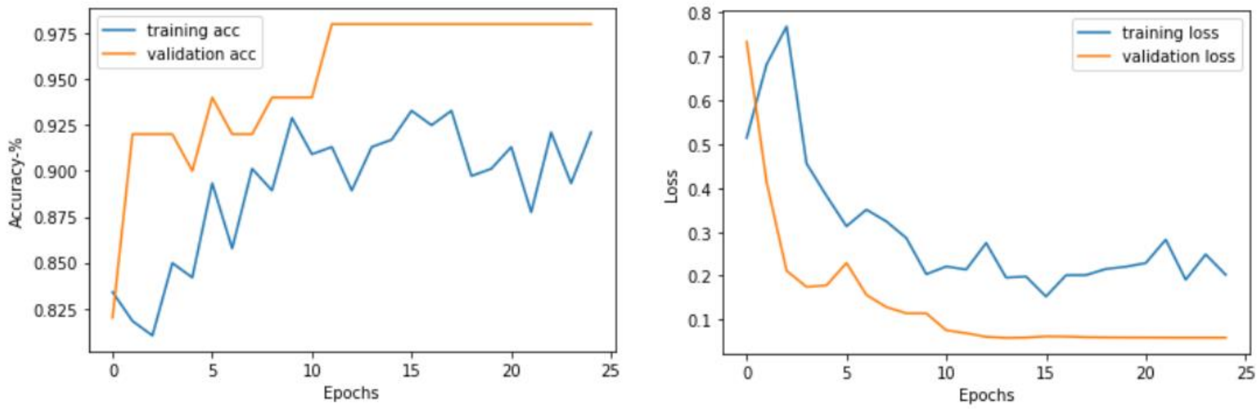

Figure 9

The confusion matrix generated using the VGG-19 method after the augmentation technique was implemented in the image dataset is as follows.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	18
1	1.00	0.97	0.98	32
accuracy			0.98	50
macro avg	0.97	0.98	0.98	50
weighted avg	0.98	0.98	0.98	50

Code written to create, test the model and the result generation is given updated the link attached in the appendix.

(3) INCEPTION-V3

The third model I have used for the implementation is ‘Inception V3’. On the ImageNet dataset, it has been demonstrated that the picture recognition model Inception v3 can achieve higher than 78.1 percent accuracy. The model is the result of numerous concepts that have been established by various researchers over the years.

Convolutions, average pooling, max pooling, concatenations, dropouts, and fully linked layers are some of the symmetric and asymmetric building components that make up the model itself. The model makes considerable use of batch normalization, which is also applied to the activation inputs. To calculate loss, Softmax is used.

High level diagram of the ‘Inception V3’ model is as given below:

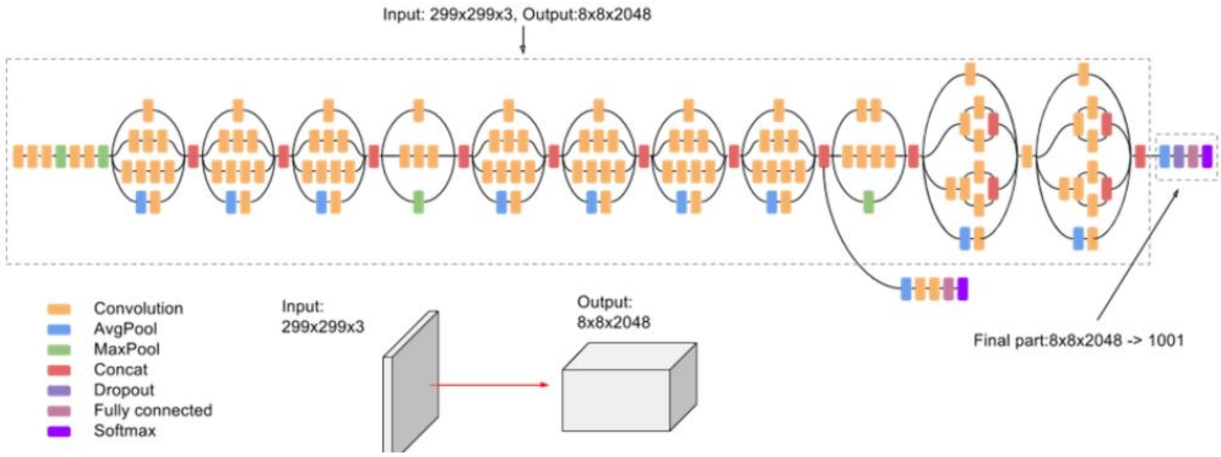

Figure 10: Architecture of Inception V3

To implement this model, I have imported all necessary libraries and then loaded the dataset. All required pre-processing steps were implemented to the loaded data set. I have set the height and width of the images to 299 X 299 to train the model. The training set was then fit into the created model. I have used 25 epochs for a full training dataset. The train set was then fit into the model and generated the results.

After that, I have implemented augmentation technique to the input image set. I have used the horizontal flip method and generated random images for the base image set. Data set generated after the augmentation method was then fit to the model and generated the results.

Value loss and accuracy were calculated and plotted for comparing the result before and after the augmentation technique was implemented.

Figure 11: Loss value and before the augmentation technique was implemented in the dataset.

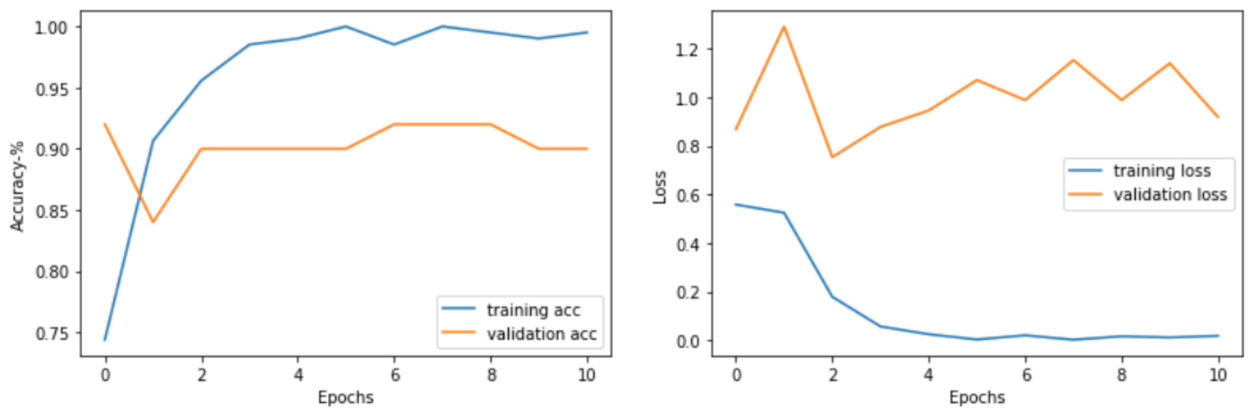

Figure 11

Figure 12: Loss value and after the augmentation technique was implemented in the dataset.

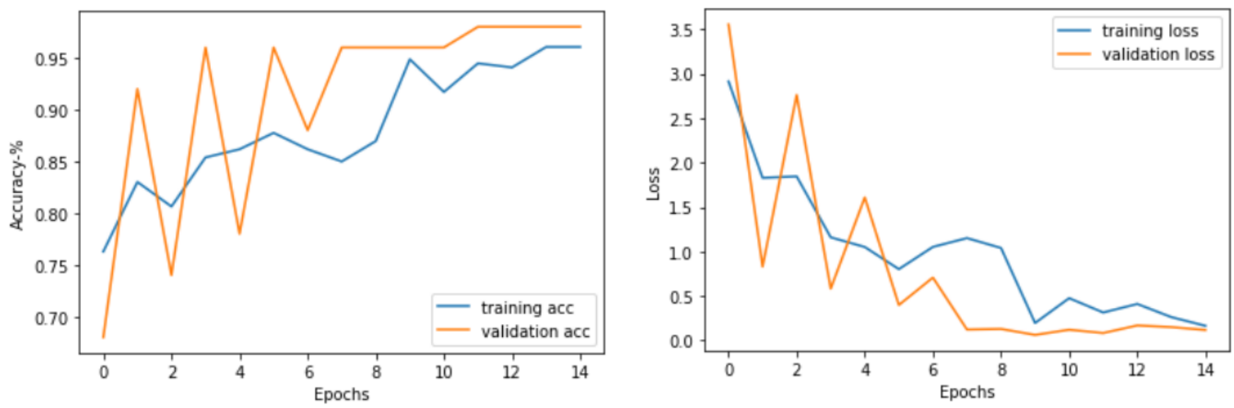

Figure 12

The confusion matrix generated using the INCEPTION V3 method after the augmentation technique was implemented in the image dataset is as follows.

```
Classification_report:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95        21
           1       0.94      1.00      0.97        29

    accuracy                           0.96        50
   macro avg       0.97      0.95      0.96        50
weighted avg       0.96      0.96      0.96        50
```

Code written to create, test the model and the result generation is given updated the link attached in the appendix.

VI. RESULT AND CONCLUSION

In this research, I have successfully implemented 3 artificial neural network methods to train and classify images to check if the input bran MRI image has a tumor or not. The best model was selected considering the following parameter values.

Accuracy: The number of successfully predicted values from the selected dataset is represented by the accuracy value. The total number of true positives and negatives divided by the total number of true and false positives and negatives is used to compute this.

Validation Loss: One of the key parts of neural networks is the loss function. Loss is nothing more than a neural network's prediction mistake. Loss Function is the name of the procedure used to calculate the loss. We can extract the gradients needed to update the weights from the loss function. The cost is determined by taking the average of all losses.

F1 Score: F1 Score represents the average of recall and accuracy, a significant factor in evaluating the effectiveness of a model. Regarding the skewed dataset, Accuracy is less significant than F1 score.

Precision and Recall: Recall refers to the percentage of relevant results that are correctly categorized by the selected algorithm, whereas Precision denotes the proportion of relevant outcomes that are.

Three artificial neural network models were considered for the brain MRI image classification, and the models are CNN using Keras, VGG-19 and Inception V3. I have trained all the models with the base image data set as well as the augmented dataset. After comparing the results it was evident that augmented image set worked well with all models and concluded that ‘VGG-19’ is the best model with the highest accuracy score set to 0.98 (98%) and lowest value loss set to 0.05.

VII. FUTURE WORK

The future work on this project includes enhancing the model performance, by introducing multiple layers and additional functions in each model, parallel we will be improvising the model techniques to train and classify 3D images.

VIII. ETHICAL CONSIDERATION

The ethical issues involved in detecting brain tumor are identical to those involved in any data-driven activity, but they are made more difficult by how important the tasks are. The MRI scan contains medical details of an individual. Hence using this information in research should be considering ethical concerns as there is a chance to lead it into legal issues.

IX. APPENDIX

Link to the dataset:

https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

Link to the code written in colab:

https://colab.research.google.com/drive/1O1rL8tnDfxgs6trT2EbFfOYZf6ktU-cG?usp=sharing

Link to the code uploaded in github:

https://github.com/priyankaharidasnk/ANN_BrainTumor

X. REFERENCES

1. nhs.uk. 2022. *Brain tumours*. [online] Available at: <https://www.nhs.uk/conditions/brain-tumours/>
2. Medium. 2022. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. [online] Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
3. Interscience.in. 2022. [online] Available at: <https://www.interscience.in/cgi/viewcontent.cgi?article=1075&context=ijcsi>
4. Arxiv.org. 2022. [online] Available at: <https://arxiv.org/ftp/arxiv/papers/2001/2001.08844.pdf>
5. A. Wulandari, R. Sigit and M. M. Bachtiar, "Brain Tumor Segmentation to Calculate Percentage Tumor Using MRI," 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), 2018, pp. 292-296, doi: 10.1109/KCIC.2018.8628591.
6. Parveen and A. Singh, "Detection of brain tumor in MRI images, using combination of fuzzy c-means and SVM," 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 2015, pp. 98-102, doi: 10.1109/SPIN.2015.7095308.
7. T. M. Shahriar Sazzad, K. M. Tanzibul Ahmmed, M. U. Hoque and M. Rahman, "Development of Automated Brain Tumor Identification Using MRI Images," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019, pp. 1-4, doi: 10.1109/ECACE.2019.8679240.
8. A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network Francisco Javier Díaz-Pernas, Mario Martínez-Zarzuela, Míriam Antón-Rodríguez, David González-Ortega Healthcare (Basel) 2021 Feb; 9(2): 153. Published online 2021 Feb 2. doi: 10.3390/healthcare9020153
9. https://www.sciencedirect.com/science/article/pii/S2314728817300636
10. Medium. 2022. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. [online] Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> [Accessed 22 July 2022].
11. https://en.wikipedia.org/wiki/Convolutional_neural_network
12. W&B. 2022. *Weights & Biases*. [online] Available at: <https://wandb.ai/ayush-thakur/dl-question-bank/reports/How-to-Handle-Images-of-Different-Sizes-in-a-Convolutional-Neural-Network--VmlldzoyMDk3NzQ>
13. Medium. 2022. *Building a One Hot Encoding Layer with Tensorflow*. [online] Available at: <https://towardsdatascience.com/building-a-one-hot-encoding-layer-with-tensorflow-f907d686bf39>
14. https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9
15. Education, I., 2022. *What are Convolutional Neural Networks?*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>

16. https://en.wikipedia.org/wiki/Brain_tumor
17. upGrad blog. 2022. *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | upGrad blog*. [online] Available at: <https://www.upgrad.com/blog/basic-cnn-architecture/>
18. https://en.wikipedia.org/wiki/Artificial_neural_network
19. Programmathically - A Blog on Building Machine Learning Solutions. 2022. *An Introduction to Neural Network Loss Functions - Programmathically*. [online] Available at: <https://programmathically.com/an-introduction-to-neural-network-loss-functions/#:~:text=The%20loss%20function%20in%20a,all%20losses%20constitutes%20the%20cost.>
20. Medium. 2022. *Understanding different Loss Functions for Neural Networks.*. [online] Available at: <https://shiva-verma.medium.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718>
21. Theaidream.com. 2022. [online] Available at: <https://www.theaidream.com/post/loss-functions-in-neural-networks>