Brain Tumor Detection Using Different CNN Architectures and MRI scans

Israk Ahmed
Department of CSE
East West University
2017-2-60-015@std.ewubd.edu

Saber Hosen
Department of CSE
East West University
2017-2-60-043@std.ewubd.edu

B.M. Jahidul Haque
Department of CSE
East West University
2017-2-60-133@std.ewubd.edu

***Abstract*—The brain is without question one of the most intricate and large organs of the human body. The brain provides us with awareness of ourselves as well as our surroundings. The brain's key functions include storing sensory information, regulating blood pressure and breathing, and releasing hormones. Distinguishing brain tumors is vital as these tumor cells shorten an individual's life expectancy rate and injure the brain cells. So, early identification of brain disorders might aid in receiving the most effective treatment as soon as possible. Magnetic resonance imaging (MRI) technology is one of the only utilized ways to identify such diseases. Other imaging techniques have not proven to be as fruitful in identifying problems in the brain as MRI. Our goal is to analyze the images of the MRI scans in our dataset using different CNN architectures and predict if a patient has a brain tumor. CNN is one of the most well-received and widely used deep learning algorithms for visual learning and image recognition. In our paper, we have put forth how accurately different convolutional neural network (CNN) architectures, namely VGG16, VGG19, and Inception V3 models, detect whether a patient has a brain tumor or not.**

***Keywords*—Neural Network, CNN, VGG 16, VGG 19, Inception V3, Machine learning, Deep learning, etc.**

I. INTRODUCTION

The key focus of our study was to find a better and efficient method to correctly and accurately detect brain tumors in the human brain. Brain tumors or intracranial tumors are aggregates or masses that form in any brain region when clusters of abnormal cells multiply rapidly without control [1]. Now some of these are benign cells, and some are malignant, i.e., cancerous. Our study focuses on using various deep learning algorithms to detect if a person has a brain tumor or not. We have acquired a data set of MRI scans for this very purpose. We chose MRI images because MRI or magnetic resonance imaging is a beneficial and thorough imaging technique that incorporates magnetic fields and radio waves to generate detailed images of different organs of the human body.

Statistically, it has been seen that people living in western countries have been affected by brain tumors more so than the people living in eastern nations of the world. Developed countries have also shown a higher rate of brain tumors in people than that in developing countries. Especially in countries and regions like Australia, North America, and Northern Europe, the rate of brain tumors is at an all-time high. Whereas in countries like Africa, the rate is relatively much lower. In 2021 for the US alone, it is expected that 24,530 new cases of brain and other nervous system cancer will be diagnosed, with 18,600 people dying from the disease [2]. Among the 18,078,957 cases of cancer recorded in the 2018 cancer registry, 296,851 were brain cancers originating as malignant tumors [3]. Our focus is to add to the already prevalent brain tumor detection techniques and progress in this field.

Basic CNN architectures have shown significant results for brain tumor detection [4]. But by using deep learning models trained on the ImageNet data–set and using their weights, we can gain higher levels of accuracy in brain tumor detection than what we would have achieved using traditional CNN models made from scratch. Using Keras in this regard is very advantageous. It contains the pre-trained models (trained on the ImageNet data–set) of popular deep learning algorithms such as VGGNet, EfficientNet, ResNet, GoogLeNet, and many more [5].

We have incorporated three different deep learning algorithms: VGG 16, VGG 19, and Inception v3 used pre-trained models of these algorithms from Keras and used the MRI images from our dataset to correctly classify the person who has a brain tumor, i.e., binary classification. The cluster of images that were used in our project consisted of two different classes ("No" encoded as "0" and "Yes" encoded as "1"). We have tried to find and have gained different accuracies for each of the three algorithms we have applied during experimentation. All of the algorithms we have incorporated in our research fall under the CNN architecture, i.e., convolutional neural network. Each algorithm utilizes a different number of hidden and fully connected layers to accomplish the task of classification. Our data set of images has been proportioned accordingly into-training, validation, and test sets'. For each of the models of CNN we have implemented, we have gotten different results in terms of accuracy and loss function (binary cross-entropy).

Our motivation for pursuing this topic stems from the fact that not enough research has been done on using deep learning to detect brain tumors, and thus it piqued our interest in contributing to this sector. The various sections in this paper cover the details of our research and experimentation, including the Introduction, Literature review, Methodology, Results and Analysis, Conclusion, etc.

II. Literature Review

Panda et al. [6] evaluated and analyzed the phases of brain tumor classification: preprocessing, feature extraction & classification. Their study emphasizes that: Various segmentation approaches have emerged to automate the process of detecting tumor sites from medical pictures. Tumor categorization can aid in selecting an appropriate segmentation approach, although it's not always necessary. Thresholding, region expansion, K-means Algorithm, Evolutionary algorithms, fuzzy C-means Algorithm, etc., are widely used for image segmentation purposes. Discrete wavelet transformation (DWT) is a professional preprocessing approach as well. Before classification, the penultimate step is extracting texture, shape, statistical, and intensity-based characteristics from the provided images. Afterward, algorithms like- SVM, NN, KNN are applied for actual classification.

In their study, Banerjee et al. [7] have used new ConvNet models built from the ground up using MRI patches, slices, and multi-planar volumetric slices MRI for image classification. Their data–set was categorized as HGG (high-grade gliomas) =262 and LGG (low-grade gliomas) =199. All three of their ConvNet models had 84.91, 90.18 & 97.19 percent accuracies, respectively, on their test set. Additionally, they tested using VGGNet & ResNet as well, gaining 83.86, 84.91 percent accuracies, respectively.

For their paper Sajjad, et al. [8] proposed a CNN model for brain tumor grade classification evaluated using both original and augmented data and assessed the results. They performed segmentation using MRI and ultimately determined accuracy with & without augmenting. Their training, cross-validation, and testing sets were 50 percent, 25 percent, and 25 percent, respectively. Accuracy gained by their data–set before data augmentation is 90.03 percent, 89.91 percent, 84.11 percent, and 85.50 percent for brain tumor grades I, II, III, and IV. Subsequently, after data augmentation, their data—set yielded accuracy in the range of 95.5 percent, 92.66 percent, 87.77 percent, and 86.71 percent for brain tumor grades I, II, III, and IV.

Abdullah et al. [9] used SVM to develop a classification model for normal and pathological brain MRI images in this paper. The pictures in their data–set were classified using feature vectors extracted from MRI scans and SVM classifiers. Their procured data–set containing about 32 individual's images was obtained from the Advanced Medical and Dental Institute (AMDI). As a feature vector, they recommended using the coefficients of both haar and db4 wavelets. These vector's represent the estimated coefficients derived from the MRI brain pictures used as SVM inputs. Using SVM with kernel type RBF, their model's accuracy was just 65%.

A. R. A. Abdulraqeb et al. [10] proposed an algorithm for segmentation to extract tumor regions from MRI images through image processing. They followed two steps to locate tumors, namely: 1) automatic threshold finding and 2) tumor localization. Firstly, they applied automatic thresholding to segment the image. This threshold was chosen based on the histograms of the MRI images. Based on the maximum count of pixels both vertically and horizontally, they located the position of the tumor. They used two datasets of weighted post-contrast MRI images to assess their proposed algorithm which garnered good results.

Another novel automated brain tumor detection & segmentation approach was suggested by Hao Dong et al. [11]. They developed a 2D complex segmentation-based network based on U-Net architecture. Some sub-tumoral areas only account for a small portion of the whole tumor volume, so they introduced the Soft Dice-based loss function. Their proposed method was mainly targeted at LGG and HGG patients. Their data–set had MRI scans with HGG patients = 220 and LGG patients= 54. To gain the definite shape during preprocessing, they applied elastic distortion. Rather than using the original U-net architecture, they use zero padding to keep the output dimension for all convolution layers. They used DSC (Dice similarity co-efficient) and Sensitivity to evaluate their model's performance.

In their paper, Mohsen et al. [12] have proposed a method to classify three different brain tumors using brain MRI images and DNN (deep neural network). Firstly, they acquisition their brain MRI dataset consisting of 66 images. They mainly focused on three types of malignant tumors- glioblastoma, metastatic bronchogenic carcinoma, and sarcoma. For image segmentation, they started off by using the Fuzzy C-means clustering technique. Afterwards for the purpose of feature extraction, discrete wavelet transformation (DWT) was used. Finally, during classification an accuracy rate of about 96.97 percent was achieved by them using DNN. They also compared their results with the results of other classifiers such as KNN, LDA & SMO.

In their article, Min, et al. [13] have put forward MRI image segmentation and enhancement techniques to aid in brain tumor detection. They used Stationary Wavelet Transform (SWT) to fuse the median and wiener filter results to enhance the quality of MRI images. They applied Adaptive K-means clustering for segmenting brain tumors from MRI images, and afterward, they conducted the morphological operation and applied the median filter to get desired results. By using their proposed method of segmentation, they reached accuracies of over 98 percent.

Chithambaram et al. [14] approached the brain tumor detection problem as such: They used two separate data–sets for their experiments. A content-based active contour (CBAC) model indicated tumor sites in MR images and saved them as segmented data. Afterward, a module was used for extracting features from tumor locations. Feature selection was then made using the Genetic Algorithm (GA) to choose a set of salient characteristics from input features. Finally, they performed classification using two separate classification modules (hybrid-SVM and hybrid-ANN) that they developed.

III. METHODOLOGY

Our study aims to detect brain tumors in patients at a preliminary stage by using MRI images and deep learning algorithms, and we have made progress in our work as such. The objective that we have put forth in our proposed method is to detect brain tumors from MRI images using the binary classification method and to achieve this. A few steps were needed to be undertaken, which have been explained in detail below.

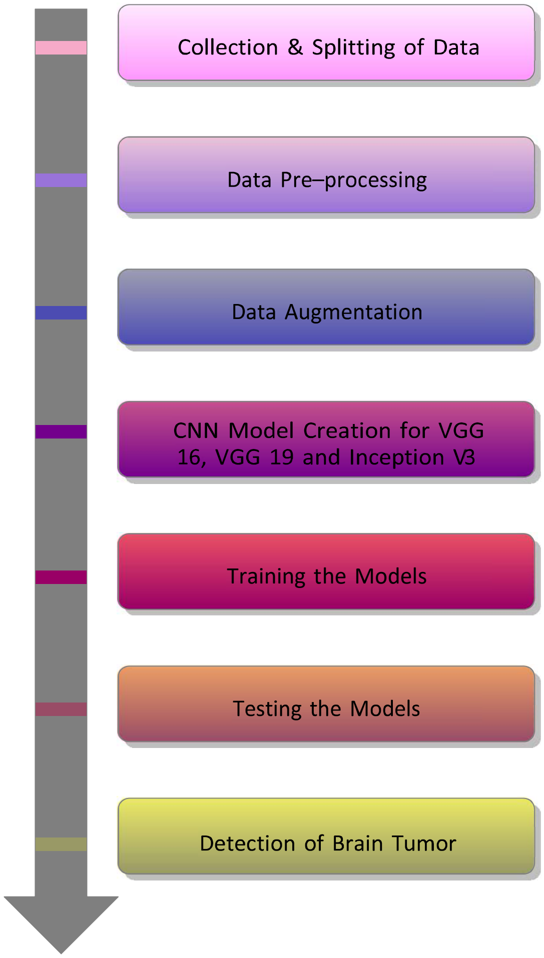
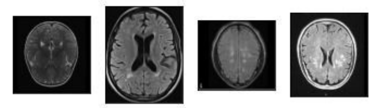


Fig. 1: Flow Diagram for Brain Tumor Detection.

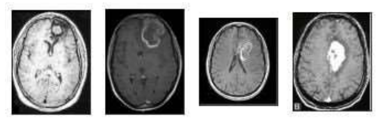
Our proposed method comprises seven steps in using different CNN architectures to find brain tumors in patients. After successfully going through these steps, our models of CNN have been able to detect brain tumors with higher levels of accuracy with less amount of loss.

A. Collection & Splitting of Data (Images)

The data for our study was collected using Kaggle, namely the Br35H brain tumor data–set. The data–set contains a total of 3000 MRI images of different patients are divided into two categories-1500 tumorous and 1500 non-tumorous MRI brain scan images [15].



(a) Non-Tumorous MRI scan



(b) Tumorous MRI scan

Fig. 2: Sample MRI images from Br35H data–set

Some samples of MRI images from our data–set have been shown above. After acquiring the data–set, we split the images of our dataset into three separate categories: Training Set (for training our CNN models), Validation Set (for use during the training phase to inspect our models' performance) and Test set (for final evaluation after classification) [16]. The following table shows how the data–set has been split to accommodate our needs:

TABLE I: Data–set Splitting Details

Type	Total no. of images	No. of Tumorous images	No. of Non-Tumorous images
Train	2000	1000	1000
Validation	600	300	300
Test	400	200	200

B. Data Preprocessing

Data Preprocessing is a very crucial step often used in image processing. To use any data, especially images with CNN models, the images need to be prepared using various preprocessing techniques. The following steps illustrate the preprocessing methods we have applied to make our data ready for use by our CNN models:

After splitting all the images into train, validation & test sets, the first step was getting all the images into the proper shape for use by the three CNN models we have worked with. The input shape for the images in the train, test, and validation directories was transformed into (224, 224, 3) dimensions where image height = 224, image width = 224, and input channels = 3. Doing so gives all the images a similar input shape, making it easier for us to feed the input into our CNN models.

The next step in our data preprocessing approach is cropping the images to remove unnecessary sections using different OpenCV library functions. The OpenCV library contains other functions that can perform various operations required for image processing [17]. The following steps were adhered to for completing the cropping process:

- Initially, the images were read using the cv2.imread() library function. Afterwards cv2.cvtColor() function was used to convert the images from RGB to grayscale. This was done so that our models could work with only one matrix per image instead of three—the function cv2.GaussianBlur() was used to add smoothness to the images.
- After successfully executing the above, we carried out the segmentation process (binary thresholding to be exact) using the function cv2.threshold() and setting the thresholding technique parameter to cv2.THRESH _ BINARY. We performed binary thresholding to create binary images from our existing grayscale images based on the tour selected threshold values. The equation for binary thresholding (segmentation) is shown below:

$$dst(x,y) = \begin{cases} maxval, & if\, src(x,y) > thresh \\ 0, & otherwise \end{cases}$$

- Now, at this stage, we were almost ready to crop the images for preprocessing. We used additional functions like cv2.erode() and cv2.dilate() to remove any small white noises and dilate the pictures after noise removal. After that, we used the cv2.findContours() function to find each image's most prominent contour and the extreme most points and cropped the images. Then we resized the cropped images back to their original size (224,224) and saved them in new directories for later use.

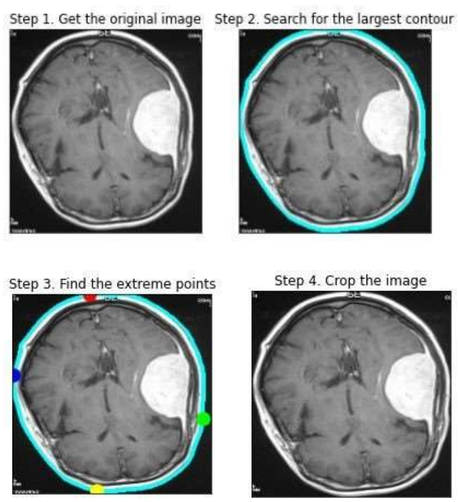


Fig. 3: Image cropping steps.

C. Data Augmentation

In practice, data augmentation is a very proficient technique to use when dealing with more minor data–sets. Large quantities of data about the medical field are hard to come across. Data Augmentation utilizes several different techniques that increase the size and quality of training data sets so that better deep learning models can be developed using them [19]. In our case, we used the ImageDataGenerator class from Keras, which uses various augmentation techniques present in its arsenal to augment the images of our data set. The methods that we used are namely: 1) Random Rotations (to rotate the images to a certain degree), 2) Random Shifts (to shift the images both vertically & horizontally), 3) Shearing (to shear the images in the counter-clockwise direction up to a certain degree), 5) Random Brightness (to brighten or darken the images), 6) Random Flips (to flip the images vertically & horizontally) [20].

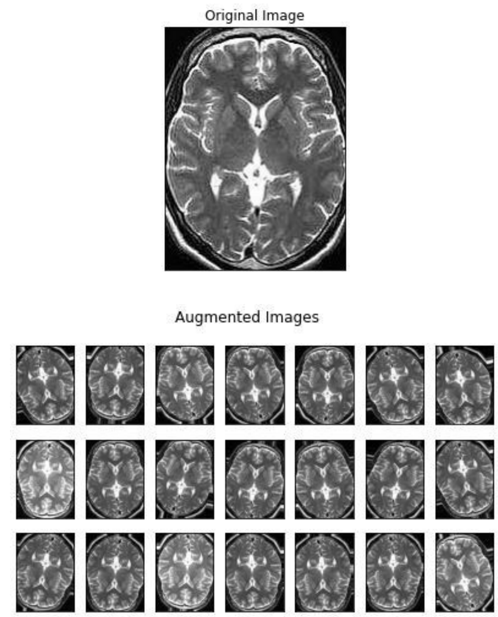


Fig. 4: Image Augmentation.

D. CNN Model Creation for VGG 16, VGG 19, and Inception V3

After successfully preprocessing and augmenting our images, it was finally time to create our deep learning models for VGG 16, VGG19 & Inception V3. We have made use of transfer learning to implement each of these deep learning models in our experiment. Transfer learning is an approach used in machine learning in which a model incorporated & trained on one data–set is reused with a second similar data– set [21]. Each of the models is pre-trained on the ImageNet data–set and has been implemented through Keras. We have not loaded the fully connected layers of the models we acquired from Keras for our CNN models. Instead, for each model, we have used our own layers at the end for getting output. These are- a flatten layer, a dropout layer (0.2), a dense layer with activation function = 'relu' having 1024 units (neurons), and a final output (dense) layer with activation function = sigmoid, which will be used for the final classification.

- **VGG 16:** VGG 16 is a CNN architecture consisting of 16 layers. It is a CNN model used widely for image classification purposes. The layers used in VGG 16 architecture are convolution, max-pooling, fully connected, and dense layers. It conforms to the standard CNN structure (Convolutional layers stacked before max-pooling layers stacked before fully Connected Layers and finally ending with dense layers.)

- **VGG 19:** VGG 19 is another CNN architecture that is 19 layers deep. The only difference between the VGG 19 and VGG 16 architectures is that VGG 19 was designed using three extra convolution layers.

- **Inception V3:** Inception v3 is another immensely popular CNN architecture used widely for image recognition and classification purposes. Unlike the VGG architectures, this model is 48 layers deep and has additional layers within its architecture like average pooling, concatenation & dropout layers alongside the traditional CNN layers like convolution, max-pooling, fully connected & dense layers [22].

E. Model Training

After our CNN models for VGG 16, VGG 19 & Inception V3 was ready, it was time for the training phase to begin. In the training phase, we used the training and validation sets we had put aside earlier. The train set is, in fact, the part of the data–set that we utilize to train our CNN models. The validation set is also a part of the data set that is used during training to get an idea of how proficiently our models are performing, using images that aren't being used for training. To train our CNN models, we have adjusted accordingly & worked with the following hyperparameters, loss function, and optimizer for each model:

TABLE II: Details regarding the loss function, optimizer & hyper-parameters used while training each CNN model

Training batch size	32
Validation batch size	16
Epochs	20
Steps per epoch	50
Validation steps	30
Loss function	Binary cross-entropy
Optimizer	Adam
Learning rate	0.0005

F. Model Testing

After finishing the training phase, it was time to test out the performance of each model using the test set that we kept aside earlier [23]. The performance of each of our models (VGG 16, VGG 19 & Inception V3) was tested on the test set. Different levels of accuracy were observed in classifying whether or not the images in our test data set were tumorous. For VGG 16, the highest accuracy that was observed was 96.25%, for VGG 19, it was 96.50%, and subsequently, for Inception V3, the highest accuracy seen on the test set was 96.50% as well, according to our experiments. Other evaluation metrics besides accuracy and confusion matrices, graphs depicting training loss and accuracy for each model have been discussed in detail in the Results and Discussion section of the paper.

G. Detection of Brain Tumor

Since our CNN models have generated higher levels of accuracy in classifying brain tumors using our test data–set, we can deem our models ready for brain tumor detection.

IV. RESULTS & DISCUSSION

In this section of our paper, the results of our experimentation has been discussed, which includes: graphs of each CNN models' performance during training, the confusion matrices of each model generated using the test set, comparison between different evaluation metrics of each model and lastly, a brief analysis of the overall results of our experimentation.

A. Model Accuracy & Loss

As we mentioned in an earlier section, during the training phase, we utilized the training set for training three of our CNN models' and at the same time, we also incorporated the validation set to evaluate each of our models' performance. This is done by observing both the accuracy and loss during each epoch.

VGG 16 training history: From our VGG 16 training history, it has been observed that after each epoch, accuracy gradually increases and loss decreases, with fluctuations here and there. By the 20th epoch, the accuracy for training and validation set had increased to about 97.98% for a train set and 98.33% for the validation set. Meanwhile, the loss had decreased to about 0.0584 and 0.0576 for train and validation set's respectively. This model took the second-longest time for training. Fig. 5 represents the overall training history of our VGG 16 model.

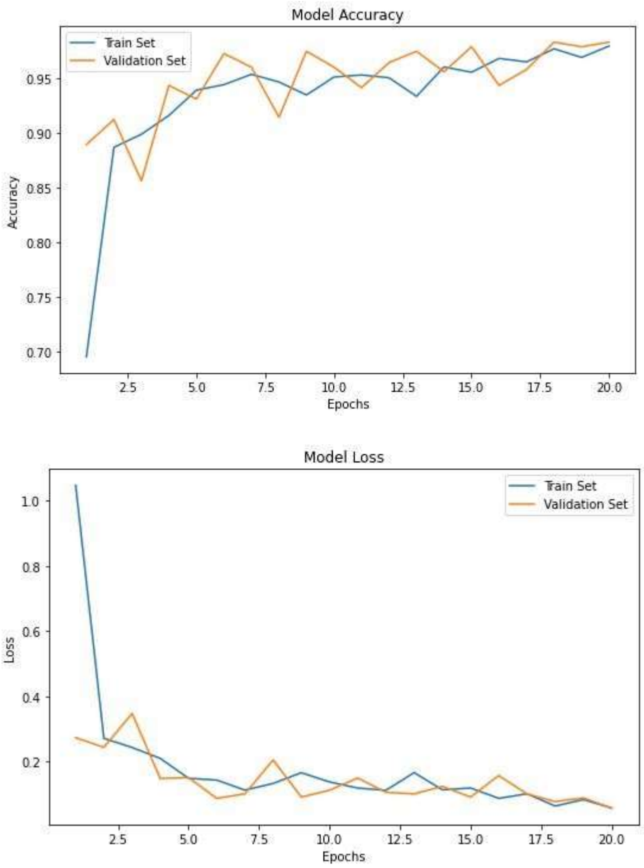
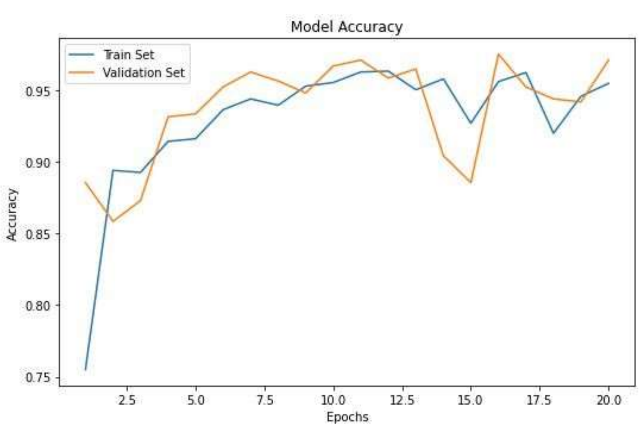


Fig. 5: VGG 16 Model Accuracy and Loss during training.

VGG 19 training history: Our VGG 19 model's training history is quite similar to VGG 16 such that, as we progress through each epoch, accuracy increases and loss decreases, with sudden fluctuations. By the final epoch, the accuracy for the training and validation set had reached 95.45% for the train set and 97.08% for the validation set. Simultaneously, the loss had decreased to about 0.1161 and 0.0904 for train and validation set's respectively. This model took by far the longest time to train. Fig. 6 depicts our VGG 19 model's overall training performance.



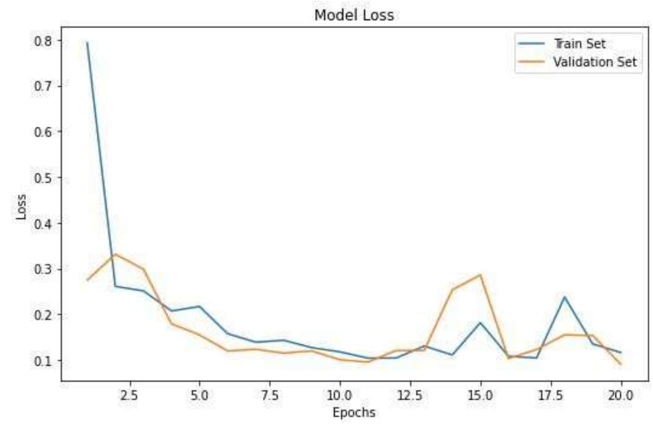


Fig. 6: VGG 19 Model Accuracy and Loss during training.

Inception V3 training history: Like the other two models, our inception v3 model also follows the gradual increase in accuracy and decrease in loss, with sudden fluctuations. At the final epoch, the model had 94.44% and 96.25% accuracy for training and validation sets. Concurrently, the loss had decreased to about 0.1496 and 0.1104 for train and validation set's respectively. Inception v3 took the shortest time to train. Fig. 7 puts our Inception v3 model's training performance on display.

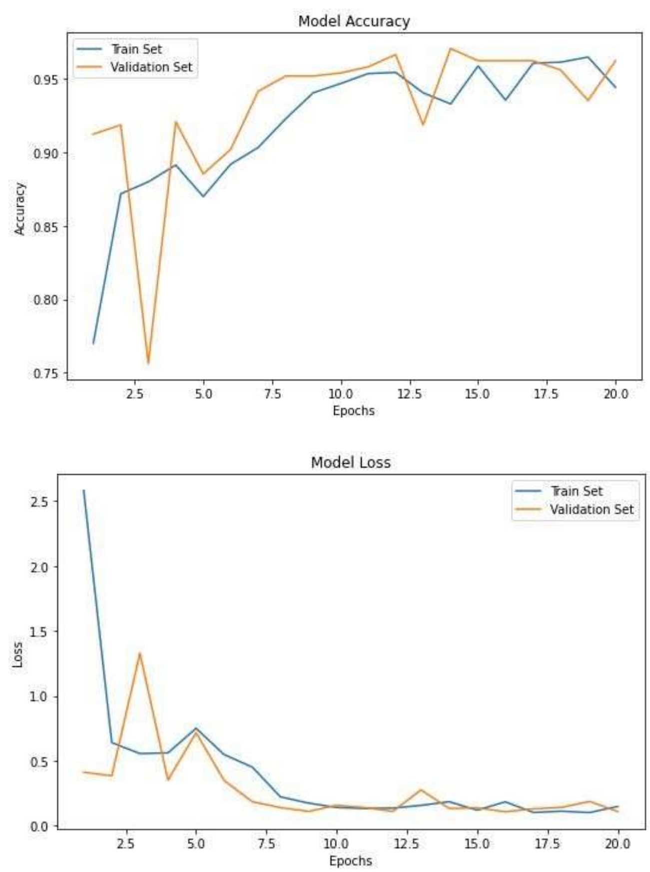


Fig. 7: Inception V3 Model Accuracy and Loss during training.

B. Confusion Matrix

A confusion matrix intended for binary classification is a 2×2 matrix constructed using the binary classifier's outcomes [24]. These four outcomes are namely- FN (False Negatives), FP (False Positives), TP (True Positives) & TN (True negatives).

TP: True positive values are those values that are were predicted as positive & are positive.

FP: False positives are those values that were predicted as positives but are negative.

TN: True negatives are such values that were predicted as negative and are indeed negative.

FN: False negatives are those values that were predicted negative but are positive.

Below are the confusion matrices of each of our models generated using the test set.

- **VGG 16:** TP = 197, TN = 188, FP = 12, FN = 3.

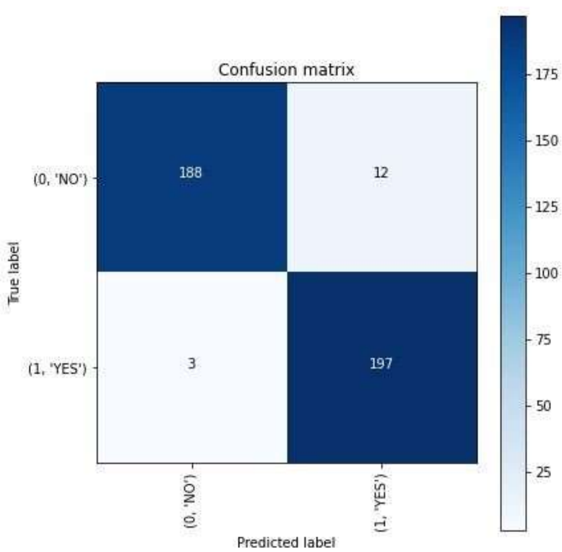


Fig. 8: Confusion matrix for VGG 16 using the test set.

- **VGG 19:** TP = 190, TN = 196, FP = 4, FN = 10.

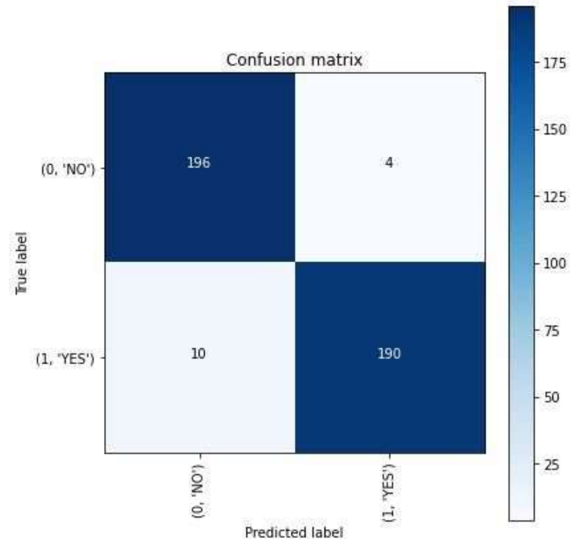


Fig. 9: Confusion matrix for VGG 19 using the test set.

- **Inception V3:** TP = 194, TN = 192, FP = 8, FN = 6.

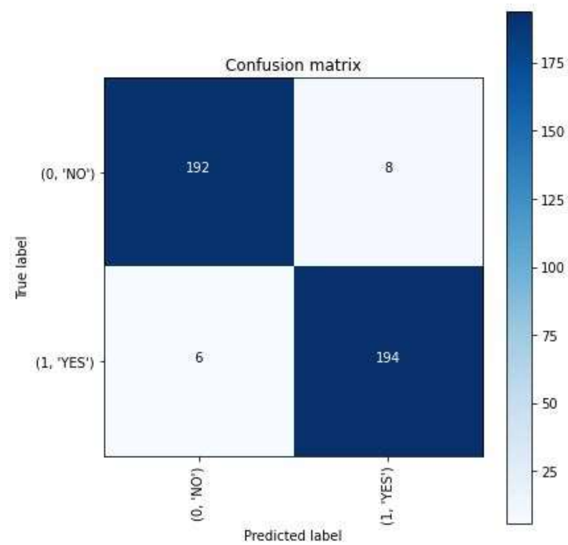


Fig. 10: Confusion matrix for Inception V3 using the test set.

C. Evaluation metrics/Classification Report

After successfully performing classification using our CNN models, one task still remained. And that task is the evaluation of the results of our classification. There are quite a few metrics for evaluation available out there [25]. To evaluate each of our CNN model's performance, we've used some specific metrics: Accuracy, F1 score, Recall & Precision.

- **Accuracy:** Accuracy provides a fraction of the total number of accurate predictions made by a model.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F}$$

- **Precision:** Precision is an evaluation metric often used to quantify the accurately identified positive values out of all the values that have been positively predicted.

$$Precision = \frac{TP}{TP+}$$

- **Recall:** Recall provides us with the fraction of accurately classified positive values concerning all the positive values.

$$Recall = \frac{TP}{TP+TN}$$

- **F1 score:** F1 score refers to the harmonic mean generated using the precision & recall values.

$$F1\ score = 2 * \frac{Presicion * Recall}{Precision + Recall}$$

Table III represents the classification report generated using each of our CNN models & our test set.

TABLE III: Classification report of VGG 16, VGG 19 & Inception V3 models

CNN model	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
VGG 16	96.25	94.25	98.50	96.32
VGG 19	96.50	97.93	95.00	96.44
Inception V3	96.50	96.03	97.00	96.51

By observing our results, we can discern that all three of our CNN models have outstanding performance in terms of accuracy. All three models' exhibit very high percentages of accuracy (over 96%), which means that the models' display very high rates for accurately classifying brain tumors. Both of our VGG 19 & Inception V3 models' astoundingly garnered the same percentage (96.5%) of accuracy during testing, while VGG 16 was not far behind with 96.25%. Other evaluation metrics such as precision, recall & F1 score also accumulated very high percentages for each of our models' as can be seen in Table III, which is a testament to our CNN models' proficiency.

V. Conclusion

In this paper, we have proposed using three separate CNN architecture models' (namely-VGG 16, VGG 19 & Inception V3) to perform binary classification for detecting brain tumors'. We've used the pre-trained models' of these CNN architectures and altered the models' according to our needs. We performed various preprocessing operations, including segmentation via binary thresholding & and image augmentation. We made the images in our data--set go through these processes to ensure that we obtain the highest possible accuracy using each model, as can be seen from our results & discussion section. To conclude we can say that when it comes to classifying whether or not an individual has brain tumor, our models' have proven to be quite capable at least from our experiments.

References

[1] Aans.org, "Brain Tumors - Classifications, Symptoms, Diagnosis and Treatments," 2021. [Online]. Available: https://www.aans.org/en/Patients/Neurosurgical-Conditions-andTreatments/Brain-Tumors

[2] U. National Cancer Institute, "Brain and Other Nervous System Cancer — Cancer Stat Facts," 2021. [Online]. Available: https://seer.cancer.gov/statfacts/html/brain.html

[3] Z. Khazaei, E. Goodarzi, V. Borhaninejad, F. Iranmanesh, H. Mirshekarpour, B. Mirzaei, H. Naemi, S. M. Bechashk, I. Darvishi, R. Ershad Sarabi, and A. Naghibzadeh-Tahami, "The association between incidence and mortality of brain cancer and human development index (HDI): an ecological study," *BMC Public Health*, vol. 20, no. 1, pp. 1–7, 2020.

[4] P. Gokila Brindha, M. Kavinraj, P. Manivasakam, and P. Prasanth, "Brain tumor detection from MRI images using deep learning techniques," *IOP Conference Series: Materials Science and Engineering*, vol. 1055, no. 1, p. 012115, 2021.

[5] S. Verma, "A Simple Guide to Using Keras Pretrained Models — by Shiva Verma — Towards Data Science," 2019. [Online]. Available: https://towardsdatascience.com/step-by-step-guide-to-usingpretrained-models-in-keras-c9097b647b29

[6] B. Panda and C. S. Panda, "A Review on Brain Tumor Classification Methodologies," *International Journal of Scientific Research in Science and Technology*, vol. 6, no. 6, pp. 346–359, 2019.

[7] S. Banerjee, S. Mitra, F. Masulli, and S. Rovetta, "Deep Radiomics for Brain Tumor Detection and Classification from Multi-Sequence MRI," *Computer Vision and Pattern Recognition; arXiv.org*, pp. 1–15, 2019. [Online]. Available: http://arxiv.org/abs/1903.09240

[8] M.Sajjad, S.Khan, K. Muhammad, W.Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," *Journal of Computational Science*, vol. 30, pp. 174–182, 2019. [Online]. Available: https://doi.org/10.1016/j.jocs.2018.12.003

[9] N. Abdullah, U. K. Ngah, and S. A. Aziz, "Image classification of brain MRI using support vector machine," *2011 IEEE International Conference on Imaging Systems and Techniques, IST 2011 - Proceedings*, pp. 242–247, 2011.

[10] A. R. Abdulraqeb, W. A. Al-Haidri, and L. T. Sushkova, "A novel segmentation algorithm for MRI brain tumor images," *Proceedings 2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology, USBEREIT 2018*, pp. 1–4, 2018.

[11] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks," *Communications in Computer and Information Science*, vol. 723, pp. 506–517, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-60964-5 44

[12] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, "Classification using deep learning neural networks for brain tumors," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 68–71, jun 2018.

[13] A. Min and N. War, "Stationary Wavelet Transform(SWT) Based MRI Images Enhancement and Brain Tumor Segmentation," *International Journal of Computer (IJC)*, vol. 39, no. 1, pp. 26–35, aug 2020.

[14] T. Chithambaram and K. Perumal, "Brain tumor segmentation using genetic algorithm and ANN techniques," *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, pp. 970–982, 2018.

[15] Kaggle.com, “Br35H :: Brain Tumor Detection 2020 — Kaggle,” 2020. [Online]. Available: https://www.kaggle.com/ahmedhamada0/braintumor-detection

[16] J. Solawetz, “The Train, Validation, Test Split and Why You Need It,” 2020. [Online]. Available: https://blog.roboflow.com/train-test-split/

[17] G. B. Garc´ıa, O. D. Suarez, J. L. E. Aranda, J. S. Tercero, I. S. Gracia, and N. V. Enano, *Learning Image Processing with OpenCV*. Packt Publishing Ltd, 2015. [Online]. Available: https://www.packtpub.com/application-development/learningimage-processing-opencv

[18] A. K. Yadav, R. Roy, Rajkumar, Vaishali, and D. Somwanshi, “Thresholding and morphological based segmentation techniques for medical images,” *2016 International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2016*, 2016.

[19] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, no. 1, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[20] A. Bhandari, “Keras ImageDataGenerator for Image Augmentation — Python Use Case,” 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/08/imageaugmentation-on-the-fly-using-keras-imagedatagenerator/

[21] J. Brownlee, “A Gentle Introduction to Transfer Learning for Deep Learning,” 2017. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[22] A. Rosebrock, “ImageNet: VGGNet, ResNet, Inception, and Xception with Keras - PyImageSearch,” 2017. [Online]. Available: https://www.pyimagesearch.com/2017/03/20/imagenetvggnet-resnet-inception-xception-keras/

[23] T. Shah, “About Train, Validation and Test Sets in Machine Learning — by Tarang Shah — Towards Data Science.” [Online]. Available: https://towardsdatascience.com/train-validation-and-test-sets72cb40cba9e7

[24] S. Kumaravel, “Confusion Matrix — What is it?. You have a binary classification... — by Sujeeth Kumaravel — Towards Data Science,” 2019. [Online]. Available: https://towardsdatascience.com/confusionmatrix-what-is-it-e859e1bbecdc

[25] H. M and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, 2015.