

Brain Tumour Detection Using Convolutional Neural Networks

B. Kalyan

Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, Telangana
kalyanbura16@gmail.com

Dr. P. Chandrasekhar Reddy

Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, Telangana
pchandureddy77@gmail.com

Abstract—Processing of medical images pose the greatest challenge when it comes to finding brain tumors. However, the traditional techniques that are used in MRI brain tumor segmentation are extremely time-consuming, and the classification can only be performed at a few diagnostic centers. In this paper we are proposing the VGG-19, ResNet50 and DenseNet121 architectures for classification and detection. The proposed system aims to categorize and distinguish the brain tumor images to identify and categorize the MRI images as “TUMOUR DETECTED” or “TUMOUR NOT DETECTED”. This proposed model architectures performed well and achieved 92% accuracy using ResNet50, 90% of validation accuracy using DenseNet121 and 88% validation accuracy using VGG19 architectures. ResNet50 achieved highest among other two architectures. Also, in this experimentation we validated all the three architectures using the Accuracy, Recall, Precision and F1-scores.

Keywords—VGG16, DenseNet121, Brain Tumor detection, Image Segmentation.

I. INTRODUCTION

However, primary brain tumours remain a significant cause of morbidity and poor overall prognosis, despite improvement in diagnosis and oncological treatment. There are two classes of primary brain tumours: grade I and II low-grade gliomas and grade III and IV high-grade gliomas (grade III anaplastic glioma and grade IV glioblastoma). Primary brain tumors begin from glial cells (termed gliomas) and are categorized based on their histopathological appearance as low-grade gliomas (LGG) and high-grade gliomas, respectively. Anteriorly, there often exists a hidden growth period and enlargement of the white matter, usually accompanied by a subtle neuro-cognitive deficit and seizures, which is sometimes subsequent changes in the region or a progression to higher grade glioma. A high-grade glioma can present both as a spontaneous primary glioblastoma or as a secondary tumor that has transformed into a high-grade tumor (e.g. secondary glioblastoma) [1].

In addition to scanning with computerized tomography, x-ray images are also used to aid in cancer detection in the brain [2]. Throughout this section you will discover the anatomical details of the different brain tissues and the structure of the entire brain. A watershed segmentation method and morphological operation are required to separate the brain tissues from the tumor during the extraction stage. However, this method does not effectively separate the brain tissue from the tumor. In order to fix the detection problem and improve current solutions, deep learning and segmentation of magnetic resonance imaging (MRI) images should be taken into consideration [3]. Especially for treating brain tumors, deep learning is advantageous in the medical

field. A huge part of image processing in the medical field is facilitated by deep learning (for example, image segmentation, computer-aided diagnosis, and spatial image reconstruction. As an example, deep learning can be used to localize anatomical structures, but it can also be used to segment vessels, the liver, and the heart. The goal of this study is to look at the possibility of convolutional neural networks (CNNs) to detect and classify brain tumors of diverse types. An artificial neural network involves a number of hidden layers that are used to extract features from an input image. Many methods have been identified to deal with the problem of the classification of brain tumors. Several researchers have proposed increasing the number of datasets, layering, and incorporating regularization and activation functions into the classification process. There are several limitations to this study solution, including the fact that it should improve accuracy, time constraints on processing, as well as weight variables. So, we propose a CNN model that is accurate, has minimal processing time, and has low weight matrices.

For the detection and classification of brain tumors using deep learning, different algorithms and techniques are used. Images are processed, features are extracted, selections are made, and calcifications are performed using different algorithms or procedures. This study was one of the first to try to increase the accuracy of the learning and testing process by attempting to improve the system's accuracy. Deep learning has the advantage of being more precise and accurate than SVM [4] as well as being more accurate. Meanwhile, CNN's are constrained by the quantity of datasets, filter sizes, layer counts, and activation functions.

The following is the outline of this paper: Section-II presents a comprehensive review of recent papers on brain tumor detection techniques. Section-III describes an effective architecture and preprocessing methodology for detecting brain cancer. In section-IV we described an explanation of the experimental comparison. This is followed by a description of the results and analysis in section-V. In section-VI, the conclusions are presented.

II. LITERATURE SURVEY

In 2021, Tingting Liu et al., proposed an automated brain tumour diagnosis system [5]. In total, four main elements make up the method: preprocessing, segmentation, features extraction, and final categorization. Discrete Wavelet Transform (DWT) and Gray Level Cooccurrence Matrix (GLCM) are used to extract characteristic features of MRIs which are forwarded to an optimized CNN diagnosis. Sparrow Search Algorithm (ESSA) classification is used to optimize CNN in this method. Finally, the efficiency of the method is compared with that of three other state-of-the-art

algorithms on the Whole Brain Atlas (WBA) database. This proposed algorithm performed the best with a score of 94.77%. The proposed technique has the highest specificity with 67.16 % [5].

A genetic algorithm-based edge detection technique was introduced by AHMED H et al., in 2020. The authors detect the fine edges using the proposed method, applied to the appropriate training dataset, with the implemented GA edge detection algorithm. Furthermore, MR scan images are compared with the number of MR scans. Edge detection techniques of both classical and fractional order are outperformed by the proposed GA edge detection method, according to the study. The performance of the system is analysed qualitatively and quantitatively based on several well-known metrics. Taking into account Pratt's Figure of Merit (FOM) and average sensitivity, the proposed method yielded an average accuracy of 99.09%. The edge detection method, which outperforms all the other methods with a sensitivity of 85.61% [6].

Using the approach proposed by Abd-Ellah et al., brain tumors can be detected and localized with improved accuracy [2]. The images were preprocessed to yield better results. The model was also modified to reduce complexity and to train faster using the existing neural network AlexNet. By using the visual geometry group (VGG-16) and visual geometry group (19), detection accuracy is 99.55%, 92.95%, and 95.15 %, respectively. Additionally, the dice score localization was 0.87. Based on this method, the accuracy reached 99.955%, indicating a significant improvement over the 66.76% accuracy obtained in the previous work. MRI images from 349 patients were used for the study. To evaluate response to a standard reference image database (RIDER), the following images are extracted from the data.

Using deep neural network (DNN) architectures, Amin et al., proposed a new architecture for the detection of brain tumors [7]. In the model, three convolutional layers, three rectified linear units (ReLU), and a softmax layer are used for classification. This study utilized eight different datasets, including BRATS, ISLES, and five different MRI modalities, including FLAIR and DWI. With FLAIR and DWI, the dice similarity coefficient (DSC) was 99.8%. The proposed model took an average of 5.502 seconds to run. Nevertheless, it is worthwhile to consider a feature that reduces the processing time of different MRI techniques.

The modern imaging technique of contrast-enhanced 3D gradient echo (CE 3D GRE) has been proposed by Jun et al., to detect brain metastases by suppressing blood vessels. To train and to test (39 sets), patients were selected randomly [8]. The depth of blood vessel suppression and lesion conspicuity of deep-learned and original Black Blood (BB) images were assessed independently by two neuroradiologists. Both deep-learned and original BB imaging had 100% sensitivity on the per-patient analysis. For the original BB imaging, the deep learned 3D BB imaging provided 100% sensitivity and 90.3% overall sensitivity in the per lesion analysis. The BB imaging technique achieved 98% sensitivity for lesions over 2 millimeters in subgroup analyses. Due to the use of the same scanner and acquisition protocols, the training set's sensitivity was reduced overall.

III. BACKGROUND METHODOLOGY

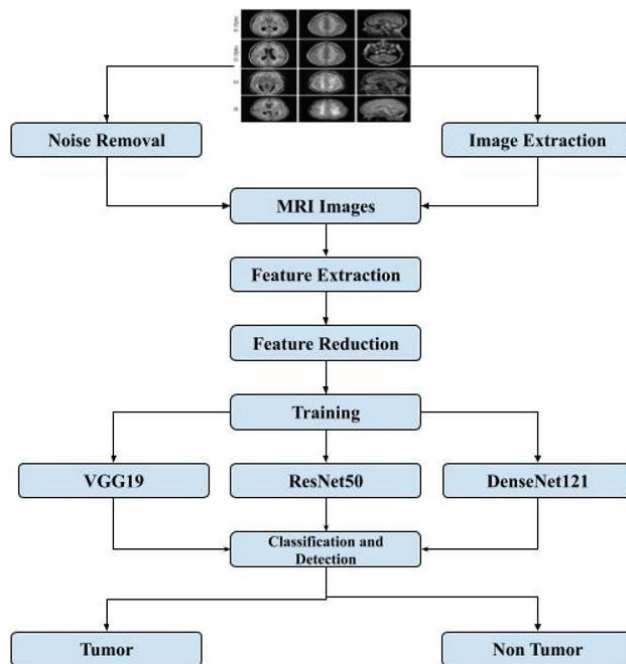


Fig. 1. Flowchart

A. Objective

In this research study, we are aiming to improve and compare the accuracy of Brain Tumour detection and classification, reduce the size of the feature vector, and detect Brain Tumours in the images of MRI scans with full accuracy [9]. The goal of this part of the paper is to provide detailed explanations of the detection and classification of tumours, as well as the algorithms used to do so. Following is a flow chart illustrating the entire process of developing the BTM model.

B. Dataset:

The dataset contains both real and synthetic tumors from MRI scans and was collected from Kaggle [10]. Figure 2 illustrates sample images that can be used as references. For the purpose of this study, original patient data was used to create this dataset. MRI images of tumours and non-tumours encompassing multiple scans are uploaded to a dataset for my proposed model to use, and my model will classify the offered MRI images as tumours and non-tumours based on the images. With those systems, we test and validate our algorithms after first training them with the data set. Within this system, in this case, 70% of the data is allocated for training, 15% is allocated for testing, and 15% is allocated for validation.

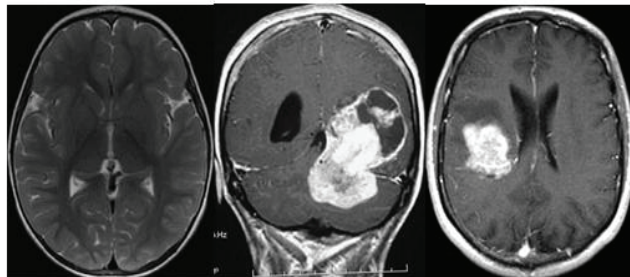


Fig. 2. Brain Tumour Dataset

C. Data Preprocessing

Model training and inference require the preprocessing of images. Preprocessing involves converting images so that they can be used by the model. Among these are resizing, orientation, and color correction. In order to expose the model to more examples, image augmentation involves manipulating images so that they look different versions of the same content. An image subject can look different under a variety of circumstances when randomly changing rotation, brightness, or scale of the input image [11]. A crucial difference between image preprocessing steps and image augmentation is that image preprocessing stages are used on both training and test sets, whereas image augmentation is only used on training data. Therefore, in some cases, an augmentation could be best considered as a preprocessing step [11].

D. Feature Extraction and Reduction

The extraction process takes into account many features that are located differently in the ROI (Region of Interest) of that particular image. A neighbourhood and statistical feature extraction technique was utilized in this study in order to segment the tumors. These extracted features can be useful for various purposes in the area of Image Processing, such as increasing the quality of an image, and so on. However, there is also an extraction of higher contrast highlights from MRI pictures. In the higher-order situation, pixels' forces are computed using a wide range of statistical features, such as skewness, histogram, standard deviation, and kurtosis. Insights of the third order will also be computed. In the feature data, all of these highlights are joined together along with their names [9].

As part of a resource-intensive computation, feature reduction may be referred to as dimensionality reduction, where features are reduced without sacrificing important information [12]. By reducing the number of features, the computer's work can be easier and faster, since there are fewer variables. There are two parts to the feature reduction process: selection and extraction. Features are reduced using many different techniques. There are several types of generalized discriminant analysis that are quite popular among experts. Examples of these include non-negative matrix factorization, autoencoders, and PCA [12].

E. ImageNet and Transfer Learning

Images are organized into a hierarchical structure in ImageNet according to the WordNet database. To provide assistance to researchers, students, and other researchers in the field of image and vision research.

Researchers from around the world are challenged to come up with solutions that yield the lowest top-1 and top-5 error rates in ImageNet's Large-Scale Visual Recognition Challenge (ILSVRC). A 1,000-class training set of 1.2 million images is available as well as a 50 thousand image validation set and a 150 thousand image test set [13].

This process involves reusing a model created for one task as the basis for another model. Using pre-trained models is a popular approach in deep learning, obtaining skill jumps on related problems using the vast amount of computing power as well as the time necessary to create neural network models for these challenges [14].

F. VGG19

VGG19 consists of 19 layers (16 convolutions, 3 fully connected, 5 Max Pool layers and 1 SoftMax layer) as a variant of the VGG. VGG also comes in versions such as VGG11, VGG16, etc. There are 19.6 trillion FLOPs in VGG19.

Architecture:

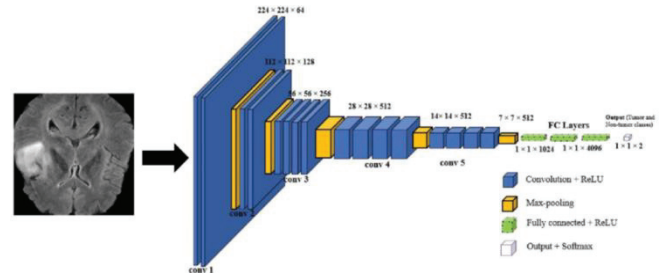


Fig. 3. VGG19 Architecture

- This network was fed a fixed-size (224 * 224) RGB image and therefore had a (224,224,3) matrix as input.
- Only one preprocessing step was performed, which was to subtract from each pixel the average RGB value calculated throughout the whole training set.
- The kernels were of size (3 * 3) and had a stride length of one pixel, so the image was covered from top to bottom.
- In order to preserve spatial resolution, spatial padding was applied.
- Sride 2 was used to do max pooling over 2 x 2 pixel windows.
- Previously, tanh or sigmoid functions were utilised in the models; the rectified linear unit was used to incorporate non-linearity to increase classification and computation speed..
- In the first two layers, which were connected fully, the size was 4096, and in the third layer, a channel count of 1000 was implemented for a 1000-way ILSVRC classification, and finally the last layer is a softmax function.

G. ResNet50

ResNet50 is one of the variant of ResNet architecture model. It is developed using 48 Convolution layers along with one MaxPool and one Average Pool layer. The number of floating point operations is 3.8×10^9 . ResNet50 architecture has been thoroughly explored and is a widely-used ResNet model.

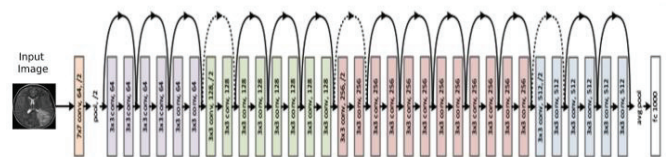


Fig. 4. ResNet50 Architecture

The RESNET 50 architecture is summarized in fig.2 as follows:

- We have a convolution of kernel size $7 * 7$ and 64 different kernels with strides of size 2 forming 1 layer.
- The next step is to max pool with strides of size 2.
- We then proceed with a $1 * 1,64$ kernel, followed by a $3 * 3,64$ kernel, followed by a $1 * 1,256$ kernel, which is repeated three times in all, giving us a total of nine layers in this step.
- This technique was done four times, giving a total of 12 layers. Next, we see the kernel of $1 * 1,128$ and then $3 * 3,128$ and finally $1 * 1,512$.
- This is followed by a kernel of $1 * 1,256$, followed by another one of $3 * 3,256$, followed by a final one of $1 * 1,1024$, which is repeated six times, giving us a total of 18 layers.
- In total, we had nine layers composed of three $1 * 1,512$ kernels, then three $3 * 3,512$ kernels, and two $1 * 1,2048$ kernels.
- Our next step is an average pool, followed by 1000 fully connected nodes, and finally we perform a softmax function to produce 1 layer.

Our calculation does not include the activation functions or the maximum/average pooling layers.

Therefore, we have a Deep Convolutional Network with $1 + 9 + 12 + 18 + 9 + 1 = 50$ layers.

H. DenseNet121

DenseNets are divided into DenseBlocks, which can each contain the same size feature map, but differ in filter number between them. Known as Transition Layers, these layers between blocks lower the number of channels to half of the current number of channels.

The H_1 function for every layer, according to the preceding equation, comprises three successive algorithms: batch normalization (BN) followed by a rectified linear unit (ReLU) and by a convolution.

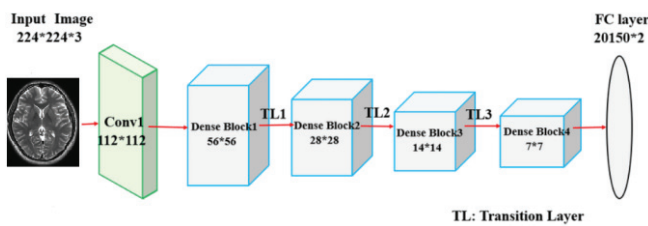


Fig. 5. DenseNet121 Architecture

Understanding the DenseNet-121 architecture allows us to recognize that each dense block contains different layers with two convolutions each; a bottleneck layer of 1×1 size and a convolution layer of 3×3 size.

Furthermore, each transition layer has a 2×2 average pooling layer with a stride of 2 and a 1×1 convolutional layer. This results in the following layers:

- An eight-by-eight convolution layer with 64 filters from 7×7 and a stride of 2
- Layer of pooling with 3×3 maximum pooling and 2 strides

- Repeated 6 times with Block 1 dense convolutions
- 1 Conv+1 AvgPool as transition layer 1
- Dense Block 2 repeated 12 times with two convolutions
- The transition layer 2 consists of 1 convolution followed by 1 average pool
- in dense block 3, where 2 convolutions are repeated 24 times in dense block 3
- 1 Convergence and 1 Average Pool on Transition Layer 3
- in the Dense Block 4 with 16 repeats of 2 convolutions
- Intended to allow all the feature maps on the network to be used for classification, this layer accepts all the feature maps
- that can be displayed on the output layer.

IV. EXPERIMENTATION

An image dataset of 253 brain MRI images was processed and augmented in this study [14]. Using a transfer learning approach we trained them with a simple 8 layered CNN model and compared their accuracy with pre-trained VGG19, ResNet-50, and DenseNet121 models. There are 155 images in the dataset that correspond to malignant tumors and 98 images that correspond to benign tumors. To train, validate, and test our dataset, we divided the whole dataset into three segments. Validation data is sample data that is used for model evaluation and parameter tuning. Training data is used for learning models. Lastly, we use test data to determine our final model performance. We propose a method that includes several phases. Authors and Affiliations.

Training the data using VGG19, ResNet50 and DenseNet models with ImageNet weights was done after splitting off the training and test data. Our model summary has been augmented with additional layers on top of the model to train the images. This experiment uses a pretrained model so it might overfit or underfit if we were to train it again. As a result, the model's precision is very uncertain. Then, rather than retraining the entire model, we added layers on top of the pretrained models. We took all the parameters or neurons for the extra layers and put them into a single vector, or may we say, we call it a flatten layer. On top of the dense layer, we have a dropout layer which is fully convoluted after we have converted it into a single vector. We relied on ReLU activation to train the dense, dropout, and final layers, and Softmax functions since it is a two-class classification.

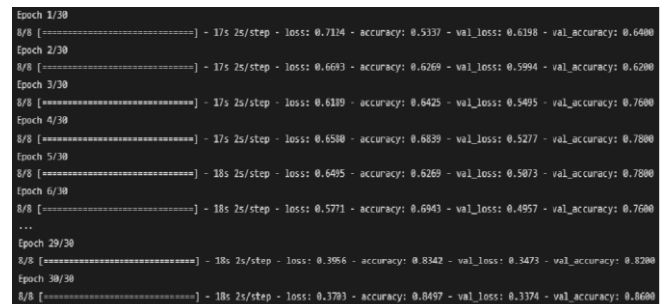


Fig. 6. VGG19 Training model

We used the RMSProp optimizer to compile the model. This optimizer implements gradient descent, which allows us to adjust the weights of the model in an optimized way in order to improve the accuracy. Due to the fact that this is a two-class classification, categorical class entropy has been set as the hyperparameter for loss function and used one hot encoding for classification. With model.fit, the training has been started, and the hyper-parameters are set at 27 batches and 30 iterations in order to pass the entire data set through the algorithm. To evaluate whether the model has been trained accurately, we preferred to use the Keras model in this experiment. This model does a cross validation to determine how accurately it was trained. In figure 16, you can see how accurate the trained model of VGG19, DenseNet121 and ResNet50 training models.

```

Epoch 1/30
8/8 [=====] - 1s 1s/step - loss: 1.3251 - accuracy: 0.5393 - val_loss: 0.6226 - val_accuracy: 0.6398
Epoch 2/30
8/8 [=====] - 9s 1s/step - loss: 0.8648 - accuracy: 0.6218 - val_loss: 0.5525 - val_accuracy: 0.7289
Epoch 3/30
8/8 [=====] - 9s 1s/step - loss: 0.8622 - accuracy: 0.6684 - val_loss: 0.5283 - val_accuracy: 0.7889
Epoch 4/30
8/8 [=====] - 9s 1s/step - loss: 0.8622 - accuracy: 0.6588 - val_loss: 0.4824 - val_accuracy: 0.8288
Epoch 5/30
8/8 [=====] - 9s 1s/step - loss: 0.6134 - accuracy: 0.7282 - val_loss: 0.4535 - val_accuracy: 0.8888
Epoch 6/30
8/8 [=====] - 9s 1s/step - loss: 0.7739 - accuracy: 0.6736 - val_loss: 0.4438 - val_accuracy: 0.8888
8/8 [=====] - 9s 1s/step - loss: 0.5653 - accuracy: 0.7977 - val_loss: 0.3553 - val_accuracy: 0.9888
Epoch 13/30
...
Epoch 29/30
8/8 [=====] - 9s 1s/step - loss: 0.3737 - accuracy: 0.8653 - val_loss: 0.3289 - val_accuracy: 0.8488
Epoch 30/30
8/8 [=====] - 9s 1s/step - loss: 0.3991 - accuracy: 0.8653 - val_loss: 0.3726 - val_accuracy: 0.8488

```

Fig. 7. DenseNet121 training model

```

Epoch 1/30
8/8 [=====] - 14s 1s/step - loss: 1.8935 - accuracy: 0.5878 - val_loss: 0.5638 - val_accuracy: 0.7088
Epoch 2/30
8/8 [=====] - 11s 1s/step - loss: 0.7789 - accuracy: 0.5987 - val_loss: 0.4889 - val_accuracy: 0.8088
Epoch 3/30
8/8 [=====] - 11s 1s/step - loss: 0.6697 - accuracy: 0.6831 - val_loss: 0.6639 - val_accuracy: 0.5888
Epoch 4/30
8/8 [=====] - 11s 1s/step - loss: 0.7496 - accuracy: 0.6289 - val_loss: 0.4279 - val_accuracy: 0.8288
Epoch 5/30
8/8 [=====] - 11s 1s/step - loss: 0.8374 - accuracy: 0.6839 - val_loss: 0.3972 - val_accuracy: 0.8888
Epoch 6/30
8/8 [=====] - 11s 1s/step - loss: 0.6288 - accuracy: 0.6684 - val_loss: 0.3511 - val_accuracy: 0.9288
Epoch 7/30
8/8 [=====] - 11s 1s/step - loss: 0.5799 - accuracy: 0.7047 - val_loss: 0.3768 - val_accuracy: 0.8488
...
Epoch 29/30
8/8 [=====] - 11s 1s/step - loss: 0.4312 - accuracy: 0.7979 - val_loss: 0.2282 - val_accuracy: 0.9888
Epoch 30/30
8/8 [=====] - 11s 1s/step - loss: 0.4433 - accuracy: 0.8883 - val_loss: 0.2370 - val_accuracy: 0.8888

```

Fig. 8. ResNet50 training model

V. RESULTS

TensorFlow and Keras libraries are used for the experiment in Python on a core i5 processor with 8 Gb of memory and a 2.3 GHz processor. On our validation dataset, we observed a 88%, 92% and 90% accuracy rate for proposed VGG19, ResNet50 and DenseNet121 models. The ResNet50 achieved a 80% of accuracy rate on training data and an 92% accuracy rate on validation data, The evaluation was conducted on test dataset to detect the misclassified images shown below.

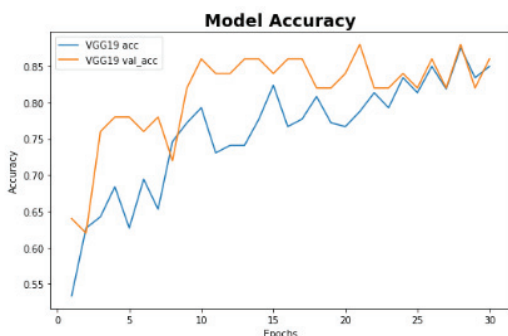


Fig. 9. VGG19 Model Accuracy curve

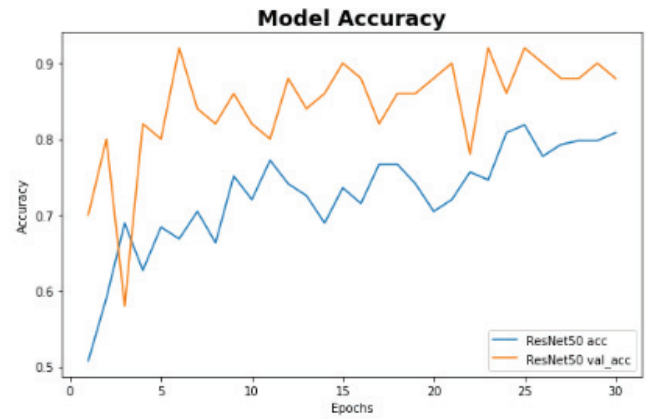


Fig. 10. ResNet50 Accuracy graphs

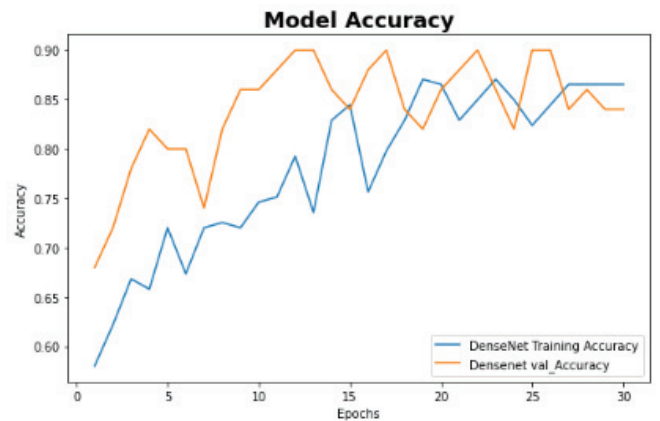


Fig. 11. DenseNet121 Accuracy graphs

According to a confusion matrix, a particular classification model or classifier that is derived from a particular set of data will always show the true performance based on the data set that is given. The terminology used in a confusion matrix may be ambiguous, but the concept itself is relatively simple to comprehend. Here, the predicted test data is plotted with the use of confusion matrix after experimenting with the VGG19, DenseNet121 and ResNet50 model. The confusion matrix represented how many misclassified outputs present in these model are shown below fig.12, 13, 14.

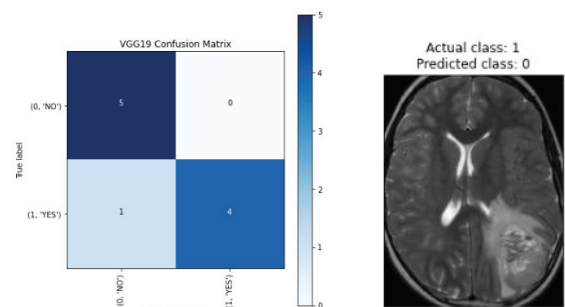


Fig. 12. VGG19 confusion matrix and missclassified image

Observing the confusion matrix in fig 12 it showed the misclassified image as the image belongs to tumour class the the model predicted as non tumour using VGG19 model.

The same way the Fig.13 depicts the DenseNet121 model detection model with confusion matrix. It detected the non-tumor image as tumor image which is mentioned in

confusion matrix. The same misclassification occurred with ResNet50 architecture.

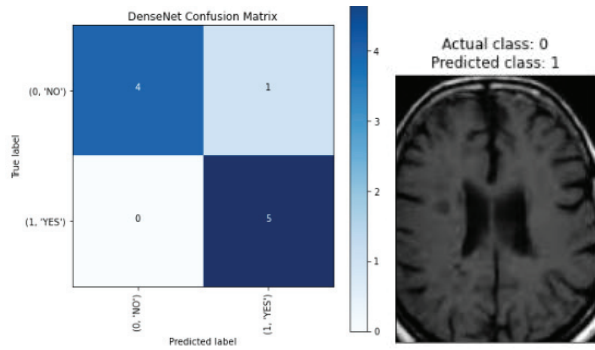


Fig. 13. DenseNet121 confusion matrix and missclassified image

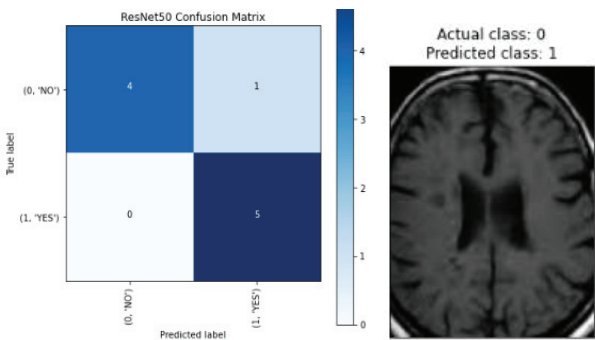


Fig. 14. ResNet50 confusion matrix and missclassified image

In this evaluation we are intended to use the four combinations of TP, FP, TN and FN to plot the ROC curve which can be plotted Precision VS Recall or True positive rate (TPR) vs False positive rate (FPR). The properties precision and recall are actual positives cases that are correctly identified.

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

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

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

The evaluation metrics of each model is shown below in table.1

TABLE I. EVALUATION METRICS

Model	ACC	Precision	Recall	F1-Score	ROC
VGG19	90	0.92	0.90	0.90	0.90
Resnet50	70	0.71	0.70	0.70	0.70
DenseNet121	90	0.92	0.90	0.90	0.90

VI. CONCLUSION

We propose a framework for diagnosing growths in brain tumor patients that can be prognostic. To enhance performance, we can consider model productivity, complete hyperparameter tuning, and a preprocessing method that is superior. We propose a framework for binary classification;

however, in future work, it may be extended to deal with straight-out labeling issues, such as identifying confirmed brain growth types like gliomas, meningiomas, and pituitaries or could be applied to distinguish other brain irregularities. We also propose a framework that could be used to distinguish risky illnesses associated with imaging in other areas of clinical practice, especially the early detection of cell death in the lungs and breast cancers, where death rates are extremely high in most developed countries. If such a methodology is spread to other logical regions where there are challenges to the accessibility of vast information or if we utilize the different learning strategies such as exchange learning with a similar proposed approach, we could have success with the methodology.

REFERENCES

- [1] "Soltaninejad, Mohammadreza, et al. "Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in FLAIR MRI." International journal of computer assisted radiology and surgery 12.2 (2017): 183-203."
- [2] "Abd-Allah, Mahmoud Khaled, et al. "Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks." EURASIP Journal on Image and Video Processing 2018.1 (2018): 1-10."
- [3] "Xu, Yan, et al. "Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation." 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015."
- [4] "Bahadure, Nilesh Bhaskarrao, Arun Kumar Ray, and Har Pal Thethi. "Image analysis for MRI based brain tumor detection and feature extraction using biologically inspired BWT and SVM." International journal of biomedical imaging 2017 (2017)."
- [5] "Liu, Tingting, et al. "An Optimal Brain Tumor Detection by Convolutional Neural Network and Enhanced Sparrow Search Algorithm." Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 235, no. 4, Apr. 2021,"
- [6] "A. H. Abdel-Gawad, L. A. Said and A. G. Radwan, "Optimized Edge Detection Technique for Brain Tumor Detection in MR Images," in IEEE Access, vol. 8, pp. 136243-136259, 2020, doi: 10.1109/ACCESS.2020.3009898."
- [7] "Amin, Javeria, et al. "Big data analysis for brain tumor detection: Deep convolutional neural networks." Future Generation Computer Systems 87 (2018): 290-297."
- [8] "Jun, Yohan, et al. "Deep-learned 3D black-blood imaging using automatic labelling technique and 3D convolutional neural networks for detecting metastatic brain tumors." Scientific reports 8.1 (2018): 1-11."
- [9] "Ramya Manaswi, V., and B. Sankarababu. "A Flexible Accession on Brain Tumour Detection and Classification Using VGG16 Model." Smart Intelligent Computing and Applications, Volume 1. Springer, Singapore, 2022. 225-238."
- [10] N. CHAKRABARTY, "Kaggle," 2019. [Online]. Available: <https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>.
- [11] J. Nelson, "Roboflow," 26 Jan 2020. [Online]. Available: <https://blog.roboflow.com/why-preprocess-augment/>.
- [12] "DEEP AI," [Online]. Available: <https://deepai.org/machine-learning-glossary-and-terms/feature-reduction#:~:text=Feature%20reduction%20leads%20to%20the,mach ine%20learning%20model%20in%20use..>
- [13] D. Infosec, "Machine Learning Blog," 06 March 2021. [Online]. Available: <https://blog.techcraft.org/vgg-19-convolutional-neural-network/>.
- [14] J. Browniee, "Machine Learning Mastery," 16 September 2019. [Online]. Available: <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>.
- [15] "Asodekar, B. H., Gore, S. A., & Thakare, A. D. (2019, September). Brain tumor analysis based on shape features of MRI using machine learning. In 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA) (pp. 1-5). IEE".