Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 27s (2025)
https://www.jneonatalsurg.com

Deep Learning-Based Brain Tumor Detection from MRI Images via Transfer Learning

Sowmya K S[1], Anitha H M[2], Jayarekha P[3], Nalina V[4]

[1]Asst. Professor, Dept. of ISE, BMSCE, Bengaluru, India

Email ID: sowmyaks.ise@bmsce.ac.in

[2]Asst. Professor, Dept. of ISE, BMSCE, Bengaluru, India

Email ID: anithahm.ise@bmsce.ac.in

[3]Professor, Dept. of ISE, BMS College of Engineering, Bengaluru, India

Email ID: jayarekha.ise@bmsce.ac.in

[4]Associate Professor, Dept. of ISE, BMSCE, Bengaluru, India

Email ID: nalinav.ise@bmsce.ac.in

Cite this paper as: Sowmya K S, Anitha H M, Jayarekha P, Nalina V, (2025) Deep Learning-Based Brain Tumor Detection from MRI Images via Transfer Learning. *Journal of Neonatal Surgery*, 14 (27s), 1119-1128.

ABSTRACT

Doctors have a difficult time detecting a brain tumor at an early stage. Noise and other environmental disturbances are more common in MRI pictures. As a result, doctors have a tough time identifying malignancies and their origins. In the realm of medical image processing, a convolutional neural network is widely employed. Many academics have worked over the years to develop a model that can more accurately detect tumors. Here is an attempt to develop a model that can accurately classify tumors from 2D MRI scans of the brain. Although a fully connected neural network can detect tumors, we chose CNN, ResNet-50 and VGG-16 as our models due to parameter sharing and connection sparsity. These models will be incorporated to identify the type of tumor, such as a glioma, meningioma, or pituitary tumor, and recommend treatment options. Preprocessing is required to convert the image to grayscale. Filters are applied to photographs to reduce noise and other environmental influences. The image must be chosen/input by the user. The image will be processed by the system using image processing techniques. To detect malignancies in brain pictures, we used a deep learning algorithm. Convolutional Neural Network (CNN) which is implemented using Keras and Tensorflow and Pre-trained models such as ResNet-50 and VGG-16 are also built to compare the results.

Keywords: *ResNet-50, Convolutional, Neural Network, Accuracy, VGG-16, Transfer Learning*

1. INTRODUCTION

A cluster of abnormal cells in the brain is referred to as a "brain tumor." Brain tumors have the ability to damage brain cells while also causing inflammation. There are two types of tumors: benign and malignant. Malignant or cancerous tumors can be split into two types: (a) primary brain tumors that begin in the brain. (b) A secondary tumor is one that has spread from another part of the body, commonly known as a brain metastasis tumor. A benign tumor is a non-cancerous mass of cells or tissues that cannot spread. The use of medical imaging in the diagnosis and prognosis of brain tumors, as well as in the treatment of the disease, is crucial. Because it is non-invasive, magnetic resonance imaging (MRI) is now one of the most commonly used medical imaging modalities for brain tumors (it does not use ionizing radiation).

Brain tumor diagnosis that is precise MR images play a significant role in clinical diagnosis and treatment planning for patients. The most typical processes in traditional machine learning classification systems include pre-processing, feature extraction, feature selection, dimension reduction, and classification. Using traditional machine learning methods in this situation is tough for a non-expert. Despite the fact that only a few methods have been used due to different inherent constraints, deep learning methods, particularly CNN, have shown outstanding success in bioinformatics. CNN, also referred to as ConvNet, is a profound machine learning technique for image analysis.

Deep Learning (DL) is a subfield of machine learning that focuses on learning hierarchical feature representations and learning hierarchical feature learning. To extract features, DL algorithms employ a system of many layers of nonlinear processing techniques. As we progress further into the network, the result of each successive layer is becoming the input of the next, promoting data abstraction. CNNs are a type of deep learning (DL) that is frequently used in image processing and is intended to require as little preparation as feasible. It is based on biological processes at work in the human brain and can handle data in a variety of formats. One of the most recent CNN methodologies looked at how well deeper CNN could segment brain tumors. ResNet-50 and VGG-16 are other deep learning concepts, more precisely transfer learning; where the model is pre-trained on a different dataset, say weights but the last layers like dense and output layers are trained on the user-defined dataset. Thus, it is known as transfer learning.

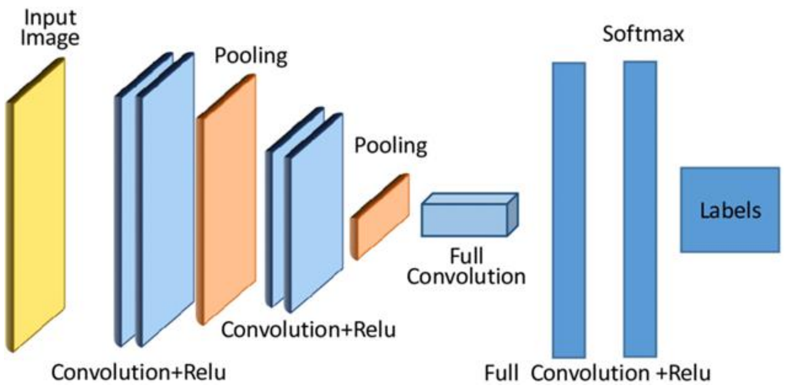

Figure 1.1 CNN model layers and hyperparameters

Figure 1.1 shows two hyperparameters: Softmax and ReLu. Convolution is the layer that forms the whole CNN model. Softmax is the activation function used in the dense layers for image classification. Relu is another activation function which is used in internal layers which is very helpful in image classification. The real non-linear function defined by ReLU(x)=max (0, x).

Instead of learning unreferenced functions, Residual Networks (or ResNets) train residual functions with reference to the layer inputs. Residual nets let these layers fit a residual mapping instead of expecting each few stacked layers to directly match a desired underlying mapping. They build a network by stacking residual blocks on top of one another: a ResNet-50, for example, has fifty layers.

The VGG16 architecture is a convolutional neural network (CNN). It is regarded as one of the best vision model architectures ever created. The most distinctive feature of VGG16 is that, rather than having a huge number of hyper-parameters, they focused on having 3x3 filter convolution layers with a stride 1 and always used the same padding and maxpool layer of 2x2 filter stride 2. Throughout the architecture, the convolution and max pool layers are arranged in the same way. It has two FC (completely connected layers) in the end, followed by a softmax for output. The 16 in VGG16 refers to the fact that it has 16 layers with different weights. This network is quite large, with approximately 138 million (approx) parameters.

To shorten diagnosis time and eliminate human errors before making any decisions, automatic tumor detection and classification technologies are required. Mostly basic images to localize a brain tumor, segmentation techniques were used. Deep learning algorithms have been widely used in a variety of applications. Visual object detection in color images. For detection and classification, the R-CNN method was chosen because it is faster. As a base network, VGG-16 is used. The algorithm emphasizes the tumor region, as well as the confidence scores, are published. The algorithm was programmed to recognize three different types of threats. Glioma, meningioma, and pituitary tumors are all examples of brain cancers.

2. LITERATURE SURVEY

[1] Weiuang Wang et al. introduced 'Learning Methods of Convolutional Neural Network Combined with Image Feature Extraction in Brain Tumor Detection' shows that the study covers a brain tumor detection approach that employs Convolution Neural Networks to execute convolution operations to increase identifying efficiency and rate, as well as artificially selected features mixed with machine learning features and MRI detection technologies. The high-dimensional data was translated into low-dimensional space using a linear transformation, such as PCA, extracting the primary features of the data in the original feature space while minimizing correlation between the features.

[2] Işın, A Direkoğlu, C. and Şah, M. in their paper 'Review of MRI-based brain tumor image segmentation using deep learning methods' proposed that SVM is used to classify tumor or normal brain MR images for classification purposes. CNN has the learning representative complex features for both healthy brain tissues and malignant tissues. SVM employs the concept of decision planes. The total number of classifiers constructed using algebraic principles is referred to as the ensemble base classification.

[3] Hossain, T., Shishir, F.S., Ashraf, M., Al Nasim, M.A. and Shah, F.M.,in their paper 'Brain tumor detection using convolutional neural networks' introduced that since brain tumors have such a wide range of appearances and because tumor and normal tissues are so comparable, extracting tumor regions from images becomes difficult. Extracting brain tumor from 2D Brain MRI Images using CNN model. CNN gained an accuracy of 96.87%.

[4] Gumaei, A., Hassan, M.M., Hassan, M.R Alelaiwi, A. and Fortino, G. in the paper ‘A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification’ stated that Current systems to assist radiologists and clinicians in identifying brain tumors need to be improved in order to provide appropriate therapy. The method consists of three steps: (A) preprocessing of brain images, (B) feature extraction from brain images, and (C) classification of brain tumors. The approach's input is brain scans, and the approach's output is the sort of brain tumor. The performance of the random holdout strategy improved from 91.51 percent to 94.233 percent in terms of classification accuracy.

[5] Bhanothu, Y., Kamalakannan, A. and Rajamanickam, G.,in the paper ‘Detection and classification of brain tumors in MRI images using deep convolutional networks’ proposed that The self-brain tumor detection and classification of MRI using deep learning algorithms is discussed in this work. With Region Proposal Network, a faster R-CNN deep learning algorithm was developed for recognizing tumors and marking the area of their presence (RPN). For all classes, the algorithm had a mean average precision of 77.60 percent.

[6] Siar, M. and Teshnehlab, M. in the paper ‘Brain tumor detection using deep neural networks and machine learning algorithms’ stated that the application of Deep neural networks model applied with machine learning algorithms for the detection of brain tumor from MRI images. Explaining the cause of brain tumors, their classification and the need for computer aided detection systems for brain tumors, the paper talks about the use of MRI images for the purpose. The paper continues to use CNN for identification and categorization of tumor which sequentially includes input layer, output layer, convolutional layers, pooling layers, normalization layers and Fully Connected layers.

[7] Pathak, K., Pavthawala, M., Patel, N., Malek, D., Shah, V. and Vaidya, B. in the paper ‘Classification of brain tumors using convolutional neural networks’ stated that a method for classifying and segmenting tumor parts using a convolutional neural network and the Watershed Algorithm to detect brain tumors and calculate the area affected by them.

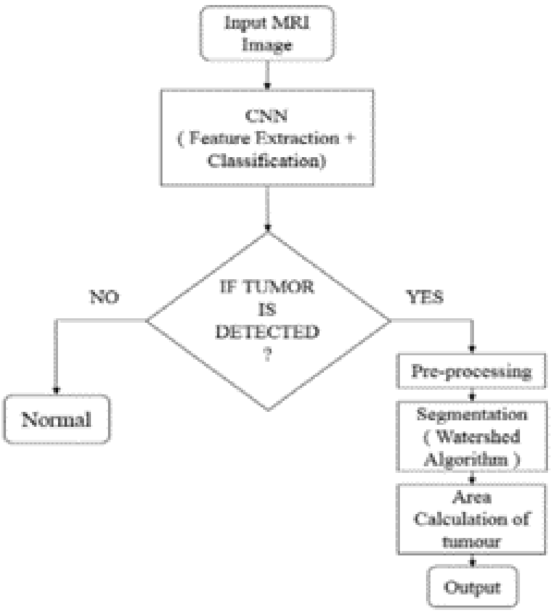

Figure 2.1 CNN Model

Fig 2.1 shows the flowchart of the CNN model used in the paper. Brain scanned MRI images are the input type. The system will initially determine whether or not there is a tumor using CNN, as the use of small kernels allows for a more detailed design while also reducing overfitting.

[8] Kumari, N. and Saxena, S. in the paper ‘Review of brain tumor segmentation and classification’ proposed that Brain tumor segmentation is a critical task in bio-medical image processing, according to this article. It takes a long time and a lot of effort to manually segment a brain tumor for cancer detection from a big number of MR images obtained in clinical practice. Brain tumor images must be segmented automatically. A literature review of many brain tumor segmentation experiments is conducted in this work. Every segmentation algorithm strives to produce an effective and precise system capable of reliably detecting tumors in the shortest amount of time.

[9] Choudhury, C.L., Mahanty, C. Kumar, R. and Mishra, B.K. in the paper ‘Brain tumor detection and classification using convolutional neural network and deep neural network’ stated how automatic brain tumor identification in health care services could be beneficial.

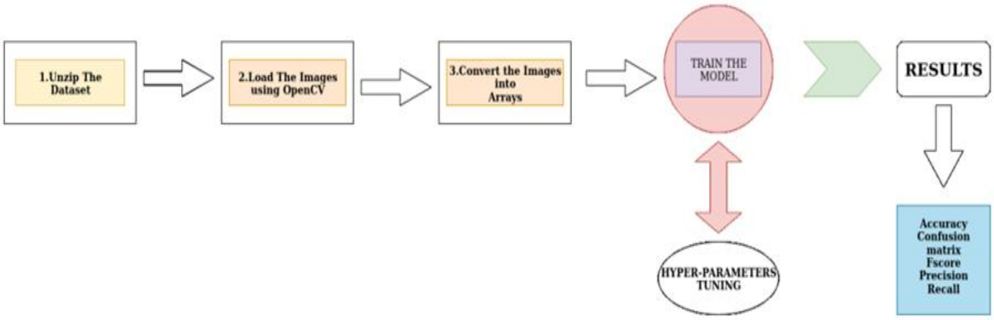

Figure 2.2 Block Diagram of model

Fig. 2.2 shows the layer retrieved with learned kernels has a varying amount of complexity, with the first layer extracting components like edges and the later layers extracting high-level features.

Computer-aided diagnosis and biomedical informatics are already helping neuro-oncologists. The authors discuss how they used Convolutional Neural Networks (CNN) to extract features and then used a fully connected network to classify them. CNNs are commonly used in deep neural networks. This consists of a multitude of nonlinear levels of activity, such as neural networks with many hidden layers. It includes the ability to display movie pictures and frames.

[10] Hussain, S. Anwar, S.M. and Majid, M. in the paper 'Brain tumor segmentation using cascaded deep convolutional neural networks' stated that Tumors have an uneven shape and can be found in any part of the brain. To improve output segmentation, a deep convolutional neural network (CNN) with a tree-like topology was deployed. The proposed network architecture appears to be promising and works exceptionally well, according to the results.

3. PROPOSED METHODOLOGY

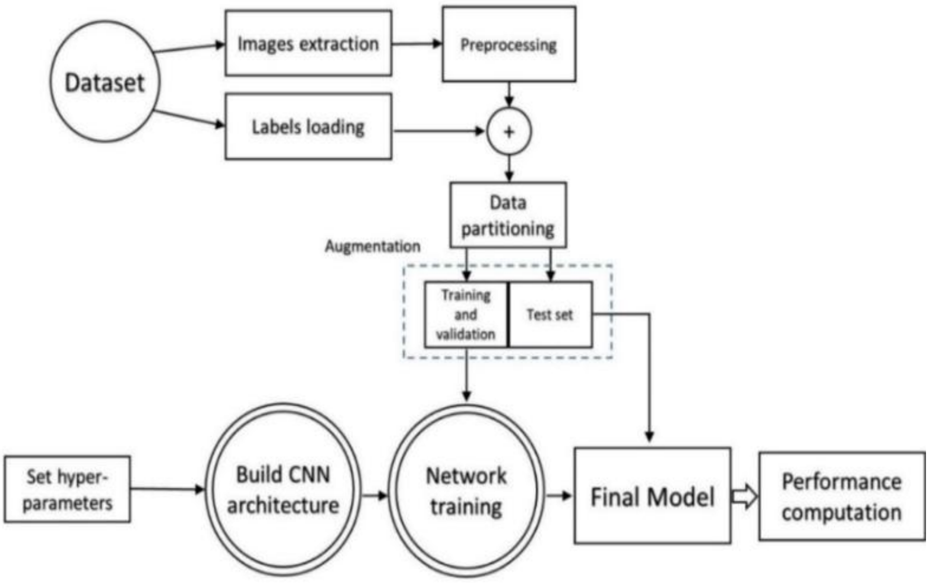

Figure 3.1 Proposed system.

We propose a system in which the approach is taken by performing Image Segmentation (The primary goal of this is to develop an automatic preprocessing step for isolating the brain) on MRI Images, which will separate the zones of Tumors from rest of the brain by the process such as Contour Detection and Gray Scaling. Now feeding this data into the classification model of Convolutional Neural Network model and training, validating and testing the model. Now, feed the input images from the frontend and using this model it will predict the type of brain tumor and suggest some diagnosis based on that. Thus, Preprocessing, Modeling, Detecting and Diagnosing Brain Tumor from MRI Images. Size of the Dataset is more than 5000 MRI Images. The dataset would be divided into training, validation and testing datasets respectively according to the requirement such that the model provides high accuracy and high validation accuracy.

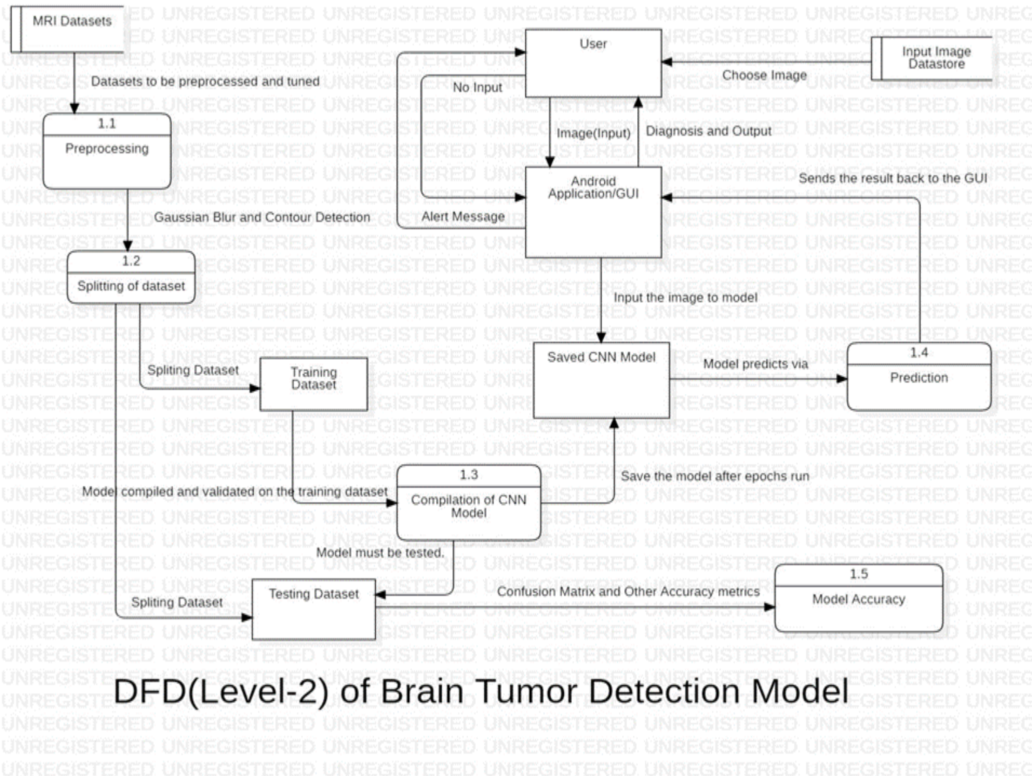

Figure 3.2 The data flow diagram

The above fig 3.2 shows the data flow diagram (level-2) of the proposed model. As the picture shows, how a user uses the UI and how the prediction works and the incorporation of the deep learning model and the frontend mobile application. Various functions of the model such as compilation of the model and prediction from the model can also be seen in the picture.

4. IMPLEMENTATION

Various models have been compiled in order to create a comparison between our own created model and other pretrained models such as VGG-16 and Resnet50 over a Brain Tumor Dataset. These models, with their training and validation accuracy, provide the required prediction needed to justify if a given image has tumor or not; and if tumor is present, then what type it is with what confidence the model predicts and the follow up procedures required to be followed by the user.

The dataset [11] was stored in the local computer, which was taken online from a trusted source. Since it is very hard to get medical records from the internet, as not many are available, we still managed to find a good dataset to create our model. Once the dataset was loaded in our notebook, we preprocessed each image during its loading, in which we applied Cropping, contour Detection, Gray scaling and noise removal. Then the training data was splitted into training and validation dataset and the data was normalized and categorically labeled by One-Hot encoding.

The CNN Model

Then, came the turn to create the model. Various layers were added and removed according to the requirement[12]. Techniques such as MaxPooling(), BatchNormalization() are used to take the maximum features out of these layers so that a good accuracy is achieved.

```
In [18]:

#Summary
model.summary()

Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_34 (Conv2D)           (None, 150, 150, 64)      1792
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 150, 150, 64)      36928
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 75, 74, 64)        0
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 75, 74, 128)       73856
_________________________________________________________________
conv2d_37 (Conv2D)           (None, 75, 74, 128)       147584
_________________________________________________________________
batch_normalization_12 (Batc (None, 75, 74, 128)       512
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 37, 36, 128)       0
_________________________________________________________________
conv2d_38 (Conv2D)           (None, 37, 36, 256)       295168
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 37, 36, 256)       590080
_________________________________________________________________
batch_normalization_13 (Batc (None, 37, 36, 256)       1024
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 18, 17, 256)       0
_________________________________________________________________
conv2d_40 (Conv2D)           (None, 18, 17, 512)       1180160
_________________________________________________________________
conv2d_41 (Conv2D)           (None, 18, 17, 512)       2359808
_________________________________________________________________
batch_normalization_14 (Batc (None, 18, 17, 512)       2048
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 9, 7, 512)         0
_________________________________________________________________
conv2d_42 (Conv2D)           (None, 9, 7, 512)         2359808
_________________________________________________________________
batch_normalization_15 (Batc (None, 9, 7, 512)         2048
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 4, 2, 512)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 4096)              0
_________________________________________________________________
dense_9 (Dense)              (None, 4096)              16781312
_________________________________________________________________
dense_10 (Dense)             (None, 4096)              16781312
_________________________________________________________________
dense_11 (Dense)             (None, 4)                 16388
=================================================================
Total params: 40,629,828
Trainable params: 40,627,012
Non-trainable params: 2,816
_________________________________________________________________
```

Figure 4.1 Summary of the CNN model

Now the model is compiled with optimizers as 'Adam' and 'Accuracy' as the metrics, 'categorical_crossentropy' as loss. Then the training data was fitted. Evaluation was carried out using the same model using validation and test datasets. Various plots of the accuracy and losses with the heatmap were also generated.

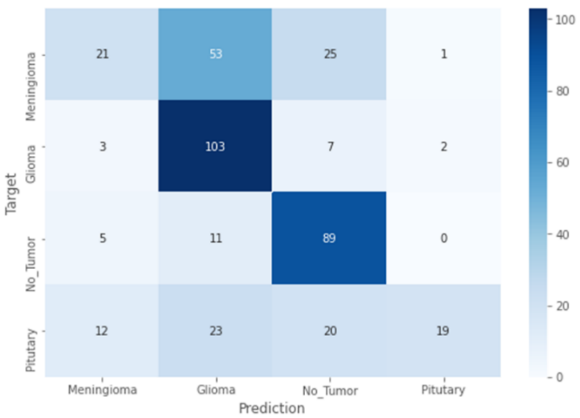

Figure 4.2 Heatmap of the CNN model.

VGG-16 Model

VGG is a cutting-edge object-recognition model with up to 19 layers of support. VGG beats baselines on a variety of tasks and datasets outside of ImageNet, thanks to its deep CNN architecture. ImageDataGenerator() function was used to create an image with rotation of the same images so that the model is trained on the images of each angle and thus helps in predicting the new image much better.

```
[ ] # ImageDataGenerator transforms each image in the batch by a series of random translations, rotations, etc.
    datagen = ImageDataGenerator(
        rotation_range=10,
        width_shift_range=0.05,
        height_shift_range=0.05,
        horizontal_flip=True)

    # After you have created and configured your ImageDataGenerator, you must fit it on your data.
    datagen.fit(x_train)
```

Figure 4.3 ImageDataGenerator() function

Now the model is trained over these images. But, before that, we will import the VGG model that is pretrained and stored in Keras. This is better because, now we do not have to build the same model from scratch and it saves resources as well as time. Weights and other hyper-parameters are assigned during retrieval. The 'ImageNet' weight was assigned to the model. ImageNet is a picture database arranged according to the WordNet hierarchy (currently only the nouns), with hundreds of thousands of photos for each node of the hierarchy.

```
model = Sequential()
model.add(vgg)
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(4, activation='softmax'))

model.layers[0].trainable = False

from keras.optimizers import adam_v2
opt = adam_v2.Adam(learning_rate=0.001)
model.compile(optimizer=opt, loss = 'categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
vgg16 (Functional)          (None, 7, 7, 512)         14714688

flatten (Flatten)           (None, 25088)             0

dropout (Dropout)           (None, 25088)             0

dense (Dense)               (None, 4)                 100356

=================================================================
Total params: 14,815,044
Trainable params: 100,356
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 4.4 VGG Model Summary

Now the model is compiled with optimizers as 'Adam' and 'Accuracy' as the metrics, 'categorical_crossentropy' as loss. Then the training data was fitted. Evaluation was carried out using the same model using validation and test datasets. Various plots of the accuracy and losses with the heatmap were also generated.

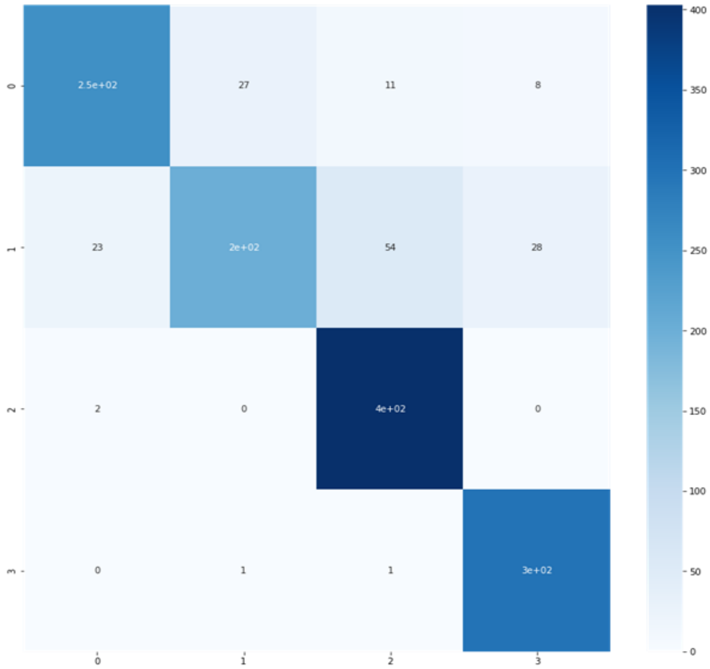

Figure 4.5 Heatmap of VGG-16 Model

Resnet-50 Model

ResNet50 is a variant of the ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. For this model, another dataset was used with more images. Same preprocessing steps were used. Images were rescaled, reshaped and gray scaled.

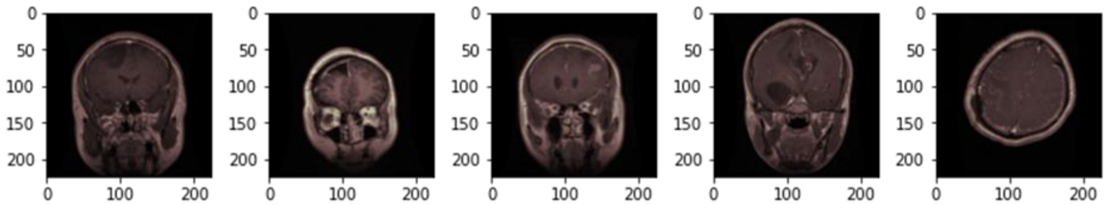

Figure 4.6 Preprocessed Images

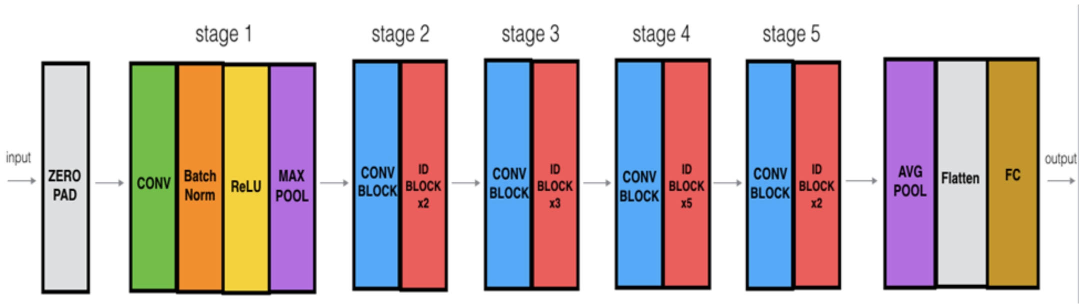

Figure 4.7 The various layers of ResNet-50 Model

Now the model is compiled with optimizers as ‘Adam’ and ‘Accuracy’ as the metrics, ‘categorical_crossentropy’ as loss.

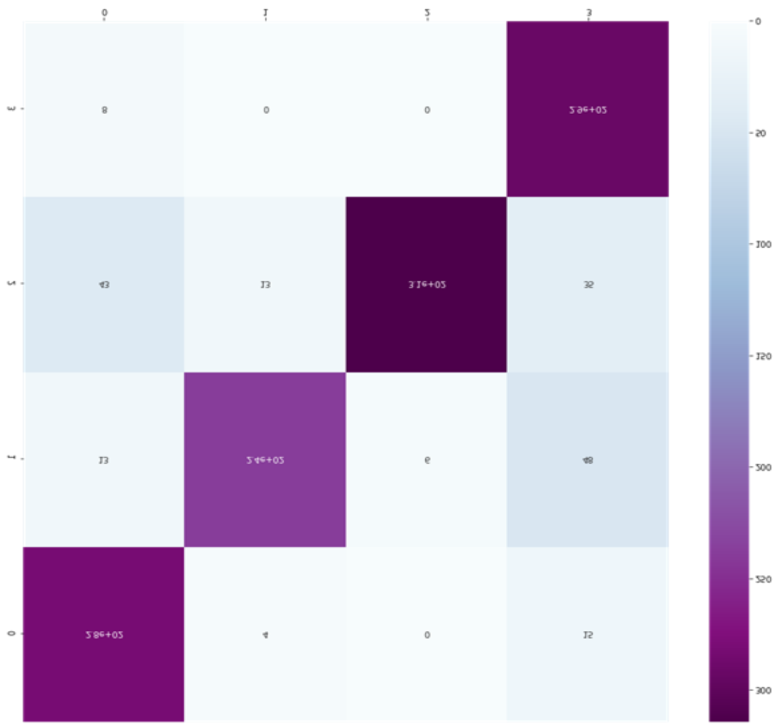

Figure 4.8 Heatmap of ResNet-50 Model

5. RESULT

The CNN Model was utilized to achieve a 98.95 percent training accuracy and an 88.5 percent validation accuracy. The Resnet-50 Model achieved a maximum training accuracy of 98% and validation accuracy of 86.53%. The VGG-16 model achieved a maximum training accuracy of 90.90% and validation accuracy of 88.18%.

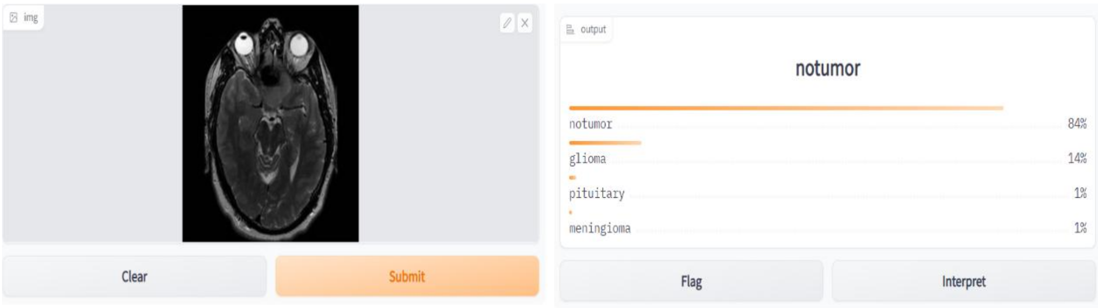

Figure 5.1 No Tumor Detection

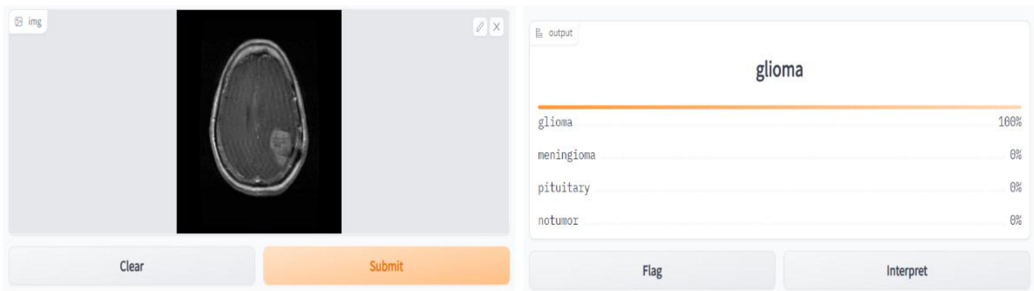

Figure 5.2 Glioma Tumor Detection

6. CONCLUSION

These Brain tumor detections are of great use in modern science to reduce human effort in detecting the tumor, types of tumors with continuous availability. In order to create a foundation for further innovation and inquiry, this work has put the traditional and existing approaches of brain tumor detection into perspective.

Nevertheless, every model has its drawbacks and this model suffers from complex image quality issues and is not accurate with other size and blurred images. But more preprocessing and smoothing the image can overcome this problem. Using various other Deep Learning models, higher validation accuracy can be achieved.

REFERENCES

[1] Wang, W., Bu, F., Lin, Z. and Zhai, S., 2020. Learning methods of convolutional neural networks combined with image feature extraction in brain tumor detection. IEEE Access, 8, pp.152659-152668.

[2] Işın, A., Direkoğlu, C. and Şah, M., 2016. Review of MRI-based brain tumor image segmentation using deep learning methods. Procedia Computer Science, 102, pp.317-324.

[3] Hossain, T., Shishir, F.S., Ashraf, M., Al Nasim, M.A. and Shah, F.M., 2019, May. Brain tumor detection using convolutional neural networks. In 2019 1st international conference on advances in science, engineering and robotics technology (ICASERT) (pp. 1-6). IEEE.

[4] Gumaei, A., Hassan, M.M., Hassan, M.R., Alelaiwi, A. and Fortino, G., 2019. A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. IEEE Access, 7, pp.36266-36273.

[5] Bhanothu, Y., Kamalakannan, A. and Rajamanickam, G., 2020, March. Detection and classification of brain tumors in MRI images using deep convolutional networks. In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 248-252). IEEE.

[6] Siar, M. and Teshnehlab, M., 2019, October. Brain tumor detection using deep neural networks and machine learning algorithms. In 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 363-368). IEEE. [7] Pathak, K., Pavthawala, M., Patel, N., Malek, D., Shah, V. and Vaidya, B., 2019, June. Classification of brain tumors using convolutional neural networks. In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 128-132). IEEE.

[7] Kumari, N. and Saxena, S., 2018, March. Review of brain tumor segmentation and classification. In 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) (pp. 1-6).

[8] Choudhury, C.L., Mahanty, C., Kumar, R. and Mishra, B.K., 2020, March. Brain tumor detection and classification using convolutional neural network and deep neural network. In 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA) (pp. 1-4). IEEE.

[9] Hussain, S., Anwar, S.M. and Majid, M., 2017, July. Brain tumor segmentation using cascaded deep convolutional neural networks. In 2017 39th annual International Conference of the IEEE engineering in medicine and biology Society (EMBC) (pp. 1998-2001). IEEE.

[10] Dataset- https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

[11] Chandana, R. K., and A. C. Ramachandra. "Real time object detection system with YOLO and CNN models: A review." arXiv Prepr. arXiv2208 773 (2022).