1-2023

Brain Tumor Detection Using MRI Images

Mayur Patel

BRAIN TUMOR DETECTION USING MRI IMAGES

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Mayur Nileshkumar Patel

January 2023

BRAIN TUMOR DETECTION USING MRI IMAGES

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Mayur Nileshkumar Patel

January 2023

Approved by:

Dr. Amir Ghasemkhani, Advisor, Computer Science and Engineering

Dr. Jennifer Jin, Committee Member

Dr. Ronald Salloum, Committee Member

ABSTRACT

When abnormal cells develop within the brain, a tumor is formed. Early tumor detection improves the likelihood of a patient's recovery. Compared to CT scan pictures, magnetic resonance imaging (MRI) is a trustworthy method for finding malignancies. In this project, we will use deep learning methods to detect tumors faster with higher accuracy using MRI images. Specifically, we will investigate the performance of transfer learning models based on convolutional neural networks (CNN) structures on the tumor detection problem. A machine learning approach called transfer learning uses a model already trained for the present task. The advantage of this technique is that we do not need to train the model from scratch, which will save time and increase accuracy.

With the help of the Visual Geometry Group (VGG 16), Inception V3, and Resnet 50, this study attempts to identify brain tumors. It also uses a methodical approach for hyperparameter tuning to improve the trained models' accuracy. The main objective is to develop a practical approach for detecting brain tumors using MRIs to make quick, efficient, and precise decisions regarding the patients' conditions. Our suggested methodology is evaluated on the Kaggle dataset, taken from BRATS 2015 for brain tumor diagnosis using MRI images, including 3700 MRI brain images, with 3300 showing tumors. The simulation results show that training the deep learning models could achieve an accuracy of 96.0% for VGG-16, 94% for Resnet50, and 90.7% for the InceptionV3 model. In order to improve the accuracy even further, Bayesian Optimization is leveraged as a hyperparameter tuning technique to obtain the best set of parameters. We could achieve the accuracy of 97.5% for VGG-16, 95% for Resnet50, and 91.5% for InceptionV3.

Keywords: Brain Tumor; Artificial Intelligence (AI); Transfer Learning; Convolutional Neural Network (CNN); Hyperparameter Tuning; Bayesian Optimization

ACKNOWLEDGEMENTS

I am writing to express my gratitude to Dr. Amir Ghasemkhani, my Committee Chair, for his continuous support and for pushing the boundaries each time to create this research work as the best version. Moreover, I want to thank my committee members, Dr. Jennifer Jin and Dr. Ronald Salloum, for believing in me and agreeing to join the committee. I am grateful for their trust in me while working on this project. I'd also like to express my gratitude to all the University's professors for assisting me in getting to this point in my academic career.

I am thankful that the School of Computer Science at California State University San Bernardino has modeled a curriculum to help me achieve my future goals and endeavors.

Someone once told me that my education is the one thing that can never be taken away from me, and I carried this with me through my master's program. This project and completing my master's degree show my bravery, determination, and dedication toward my future.

I owe a debt of gratitude to my sister and parents for paving the way for me to pursue a career in computer science.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGEMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER ONE : INTRODUCTION 1

Literature Review 2

CHAPTER TWO : SYSTEM MODEL DESCRIPTION 5

Workflow 5

2.1 Data Import and Pre-Processing 6

2.2 Data Augmentation 8

2.3 Image Pre-Processing 10

2.4 Image Segmentation 12

2.5 Feature Extraction 13

2.6 Training and Testing 14

CHAPTER THREE : DEEP LEARNING 15

3.2 Transfer Learning 17

3.2.1 VGG-16 17

3.2.2 Resnet-50 18

3.2.3 Inception V3 19

3.3 Hyperparameter Tuning 21

CHAPTER FOUR : EVALUATION 23

4.1 Dataset 23

4.2 System Configuration 23

4.3 Evaluation Metrics 24

4.4 Simulation Result 26

CHAPTER FIVE : CONCLUSION 30

REFERENCES 31

LIST OF TABLES

Table 1. Comparision Of The Models (Before Hyperparameter Tuning) 28

Table 2. Comparision Of The Models (After Hyperparameter Tuning) 28

LIST OF FIGURES

Figure 1. Machine Learning Pipeline For Brain Tumor Detection. 5

Figure 2. Crop The Image Using Normalization Technique. 7

Figure 3. Image After Applying Normalization(Tumor Is Not Present). 7

Figure 4. Image After Applying Normalization(Tumor Is Present)........................ 8

Figure 5. Original Image Before Augmentation Process...................................... 9

Figure 6. Augmented Images. 9

Figure 7. Binary Thresholding Is applied, Tumor Is Not Present. 11

Figure 8. Binary Thresholding Is applied, Tumor Is Present. 12

Figure 9. Single Layer CNN. 16

Figure 10. VGG 16 Architecture. 18

Figure 11. Resnet 50 Architecture. 19

Figure 12. Inception V3 Architecture. 21

Figure 13. Confusion Matrix Overview... 25

Figure 14. Training And Validation Accuracy Of VGG16. 26

Figure 15. Training And Validation Loss Of VGG16. 27

CHAPTER ONE

INTRODUCTION

Brain tumor disease is one of adults and children's most perilous and fatal cancers. For effective treatment, brain tumors must be recognized and classified as early as possible. The diagnosis of a brain tumor is based on image data analysis of the images of the brain obtained by MRI, CT scan, and other images. The primary step in establishing the state of a patient with a brain tumor is an accurate interpretation of brain tumor pictures. Brain tumor detection is complex, and the situation becomes more challenging when no automated detection process exists.

When tumor cells form in the human brain, the probability of significant mortality increases. Brain tumors are particularly unstable for twenty-five days due to the intricacy of tissues. The person's survival rate is often less than 12 months if they are not treated adequately. More precise computer-based and automated tumor detection/diagnosis methods are necessary to comprehend and intervene in this actual situation. Several efforts have recently been made to investigate machine-learning techniques for automating this procedure. Magnetic Resonance Imaging (MRI) identifies abnormal tissues that must be treated. Because of the complexities and variety of tumor types, brain tumor detection is difficult. Image collection, organization, and analysis have become standard procedures, which makes it possible to use data-driven techniques for brain tumor identification.

Deep learning algorithm [9] is a branch of AI that provides excellent capabilities for the tumor detection problem when using data-driven methods. They utilize artificial neural networks to perform feature extraction [4] and classification.

Reviewing the literature and research papers [9] shows that CNN models offer the best performance compared to other deep learning techniques. Therefore, our goal is to leverage the CNN models and work around them to improve their accuracy by using hyperparameter tuning.

Literature Review

Adel Kermi et al. [38] proposed an automatic brain cancer segmentation procedure in three-dimensional magnetic resonance imaging, utilizing similarity analyses of the brain and standard group. The image is preprocessed to reduce noise. The FBB method (Fast Bounding Box) is practical and unsupervised, automatically detecting tumors. The computation time for identifying and segmenting tumors is around five minutes [38], proposing an accuracy of almost 96%.

Anitha et al. [39] presented the K-means technique for classifying and detecting tumors. A two-tier approach uses the K-means technique to achieve successful segmentation and classification. The feature extraction [4] obtained after applying the discrete wavelet transforms is then used to learn the neural network's self-organizing map. The outcome filter factors are then known by the KNN neighbor and testing procedure, which is similarly done in two phases. They have achieved almost 85% overall accuracy using the K-means technique. However, there is one issue with the K-means: it depends on the initial values, and we need to choose the K values manually.

F.Milletrai [42] proposed a CNN-based model with an overall accuracy of 91.3% for different types of brain tumors. A deep learning architecture based on 2D convolutional neural networks was used to classify different types of brain tumors from MRI images. To improve the accuracy of the results, techniques such as data collection, data preprocessing, pre-modeling, model optimization, and hyperparameter tuning are used [42].

In [2], we investigated the performance of the VGG-16 model as a classifier, fine-tuner, and feature extractor. To improve the result, they used Bayesian optimization [2] to choose the optimal value of hyperparameters. VGG-16 outperformed previously proven methods by leveraging a more than 97% test accuracy while requiring less training time. However, we will use the same setup for brain tumor detection problems in this project.

R. Tamalarasi et al. [43] have leveraged the VGG-16 and Inception models for brain tumor detection. They achieved an overall accuracy of 92.8% and 95.1%, respectively. This article uses image processing, max-pooling layer, transfer learning, and model optimization. The authors in [43] have employed the trial-and-error method to identify the ideal hyperparameters.

We will leverage VGG-16, Inception v3, and Resnet 50 [21] [20] models for this project's brain tumor detection problem by considering different transfer learning models from the literature survey. To enhance the accuracy of each model, we will incorporate Bayesian optimization to tune the hyperparameters systematically.

CHAPTER TWO

SYSTEM MODEL DESCRIPTION

Workflow

Figure 1 shows the machine learning pipeline for brain tumor detection problem. In the first step, we import data and perform pre-processing using normalization techniques to remove the noise.

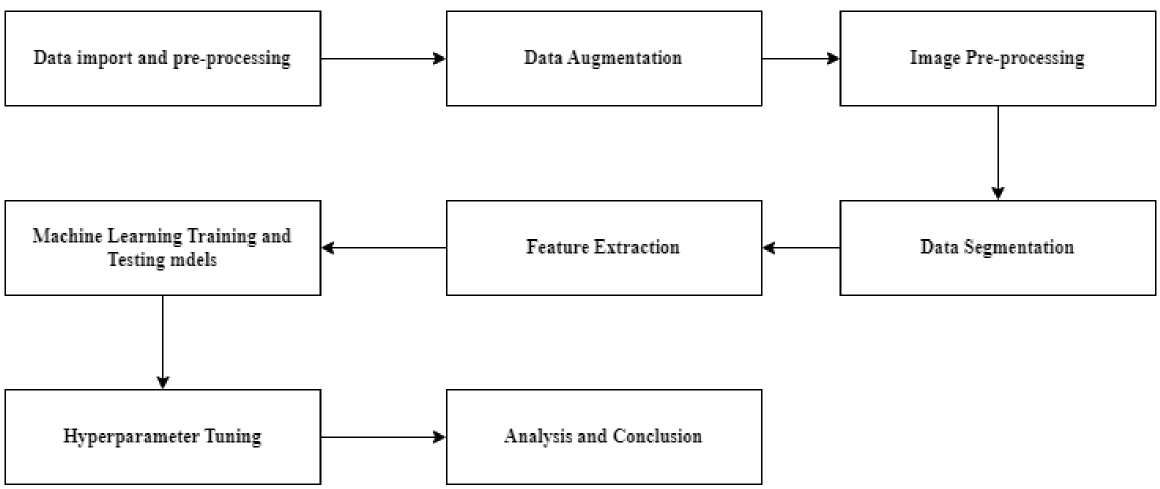


Figure 1: Machine Learning Pipeline For Brain Tumor Detection Problem

The second step is data augmentation, when we artificially increase the amount of data via strategies like reflection, scaling, and linear transformation. In the third step, we remove the undesirable markings and other small features from the MRI images using the binary thresholding method. The fourth step is data segmentation where we mainly focus on obtaining the closed outlines of the brain image. The fifth step is feature extraction where we find the anomalies from the image using the gray level occurrence matrix (GLCM) method. The fifth step is to train and test the different pre-trained models (i.e., VGG-16, Resnet 30, Inception V3). The sixth step is to find the optimal parameters via Bayesian optimizer technique to improve the accuracy of the models. To evaluate the performance of each model, we leverage different evaluation metrics.

2.1 Data Import and Pre-Processing

Data import and pre-processing [30] is a technique that will transform the data into a precise format required by high-level processing, such as data augmentation. Due to this, therefore, images are quickly processed and can be used productively for machine learning models. For this project, we have scaled the images in the range of 0-255 pixels as the brain's contour is not segregated as a tumor. We also used the Gaussian blur filter [30] to remove noise in this project because it produces better results than the median filter technique and other techniques. The normalization technique crops the images and transforms them to the same scale range.

Figure 2 represents a brain image that consists of a tumor and then shows the steps of the normalization technique, where the first step is fetching the original image. The second step is finding the most significant contour. The third step is finding the extreme points. The last step is cropping the image with the help of the extreme point extracted from the previous step.

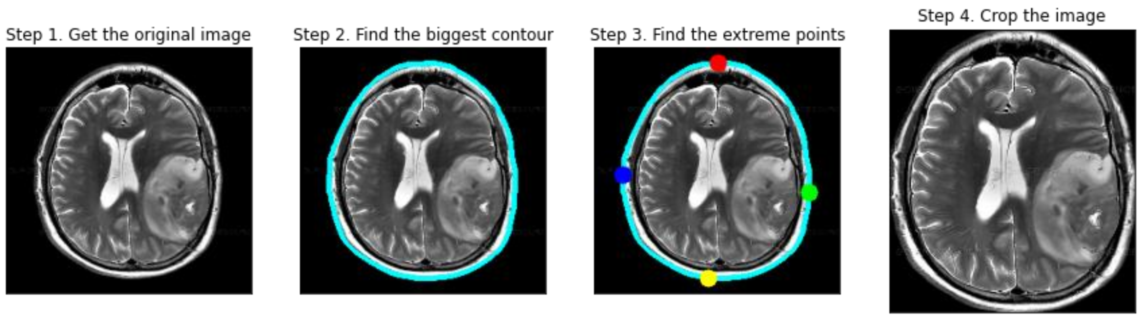


Figure 2: Crop The Image Using The Normalization Technique

Figure 3 shows the images with no tumor after applying the normalization technique, whereas Figure 4 shows the images with present tumor after the normalization technique.

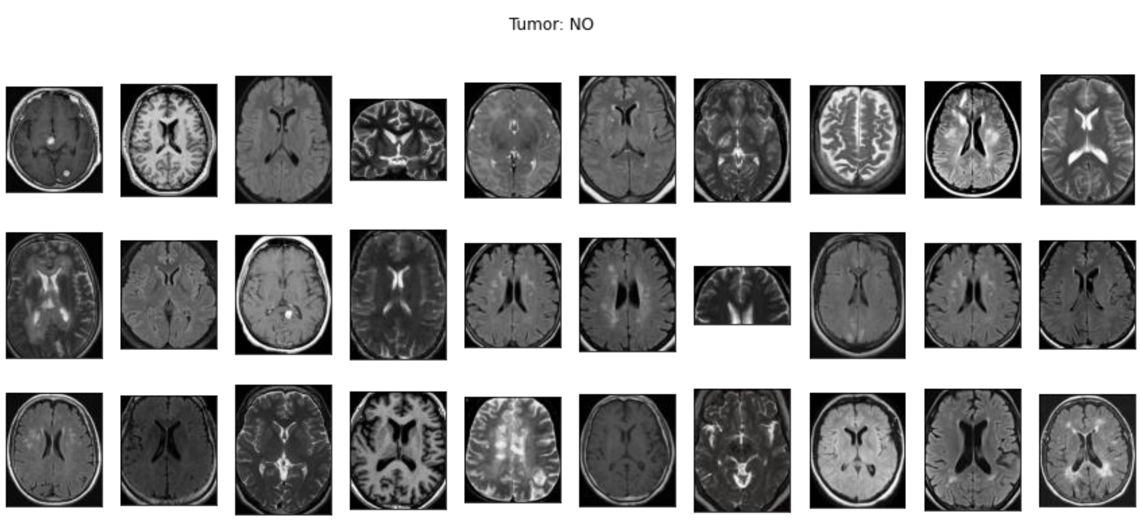


Figure 3: Image After Applying Normalization (Tumor Is Not Present)

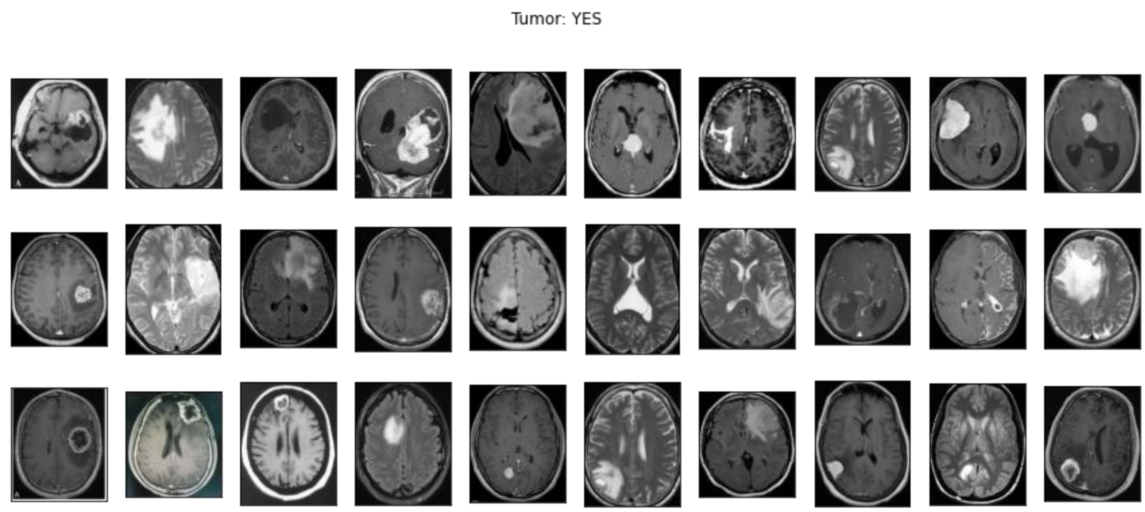


Figure 4: Image After Applying Normalization (Tumor Is Present)

2.2 Data Augmentation

Data augmentation [10] is a process of artificially increasing the amount of data by generating new data points from existing data. Data augmentation [10] can artificially enhance the amount of data. Data augmentation [10] includes making minor adjustments to the data or creating new data points for the deep learning models. Gray scaling, reflection, gaussian blur, histogram equalization, translation, and linear transformations like rotation (0–10 degrees), shifts (0–180 degrees), and flips (0–180 degrees) are the parts of data augmentation [10]. When limited training examples are available, data augmentation teaches the network the appropriate invariance and resilience qualities. This project uses grayscale scaling, reflection, cropping, and linear transformation techniques.

Figure 5 depicts one of the original images from the dataset before the Augmentation process, whereas Figure 6 represents minor adjustments done to the images after applying the techniques like transformation, cropping, and rotation.

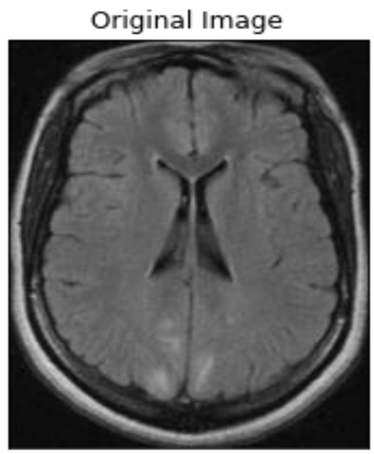


Figure 5: Original Image Before Augmentation Process

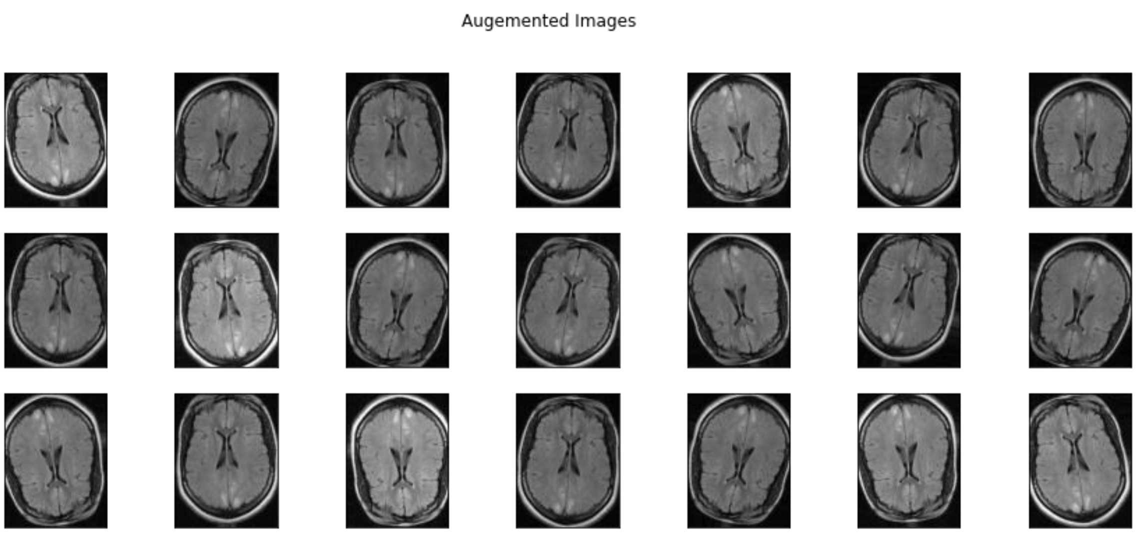
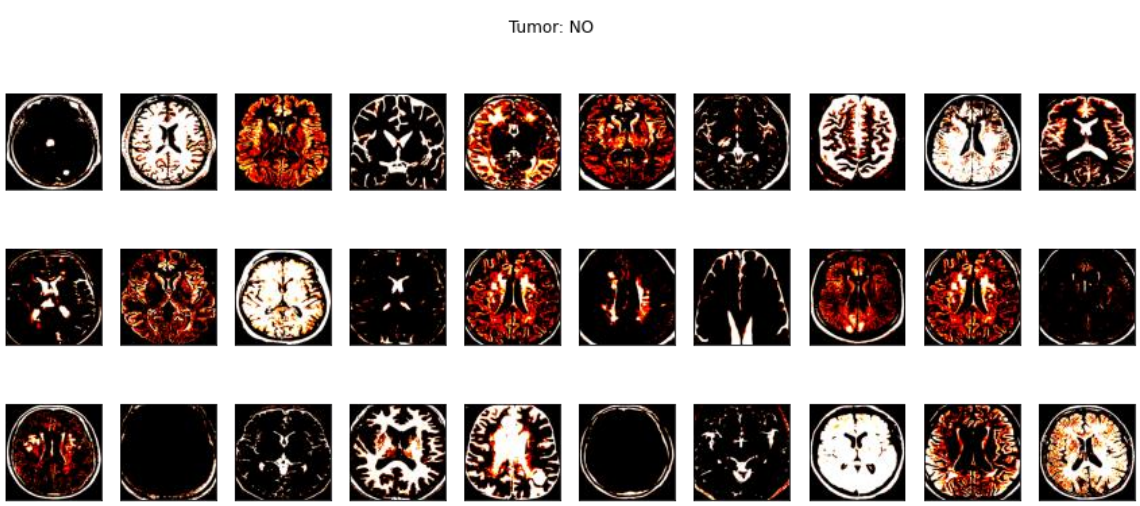


Figure 6: Augmented Images

2.3 Image Pre-Processing

Image preprocessing [13] is a significant aspect of any image-based application. Pre-processing prepares the image for higher-level processes such as segmentation and feature extraction during the preprocessing stage. Secondly, it removes the dates marked on the images, unnecessary marks, and other minute details from the image, which will affect tumor detection. Finally, the image quality is improved, and the noise will be removed

The image pre-processing stage [13] aims to improve the image's quality by suppressing unintentional distortions or enhancing some image features crucial for subsequent processing. First, we rescaled the photos and removed the noise from the pictures, and at last, we applied the Binary Thresholding [14] technique for this project. Figure 7 depicts the brain images which doesn't contain the tumors after using the binary thresholding technique. The first set of images are transformed into the RGB variant, and the second set are presented in the dark variant.



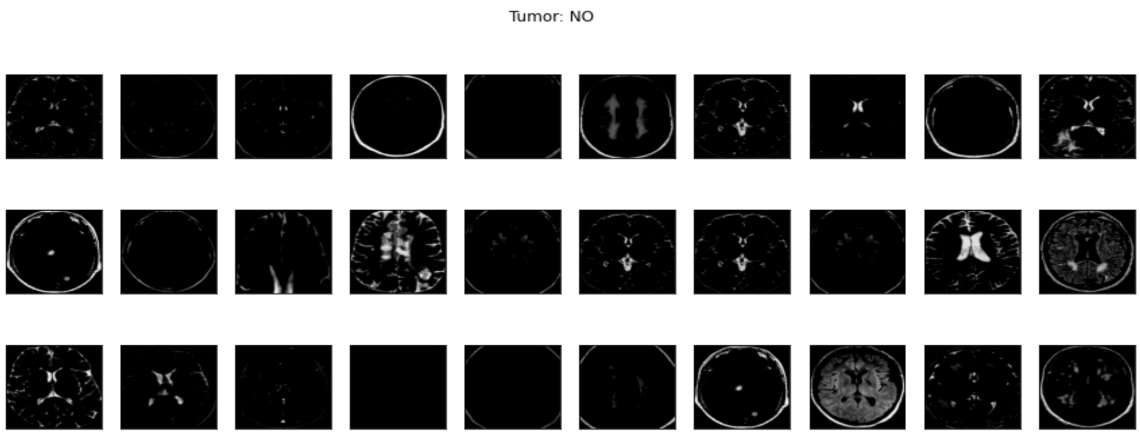
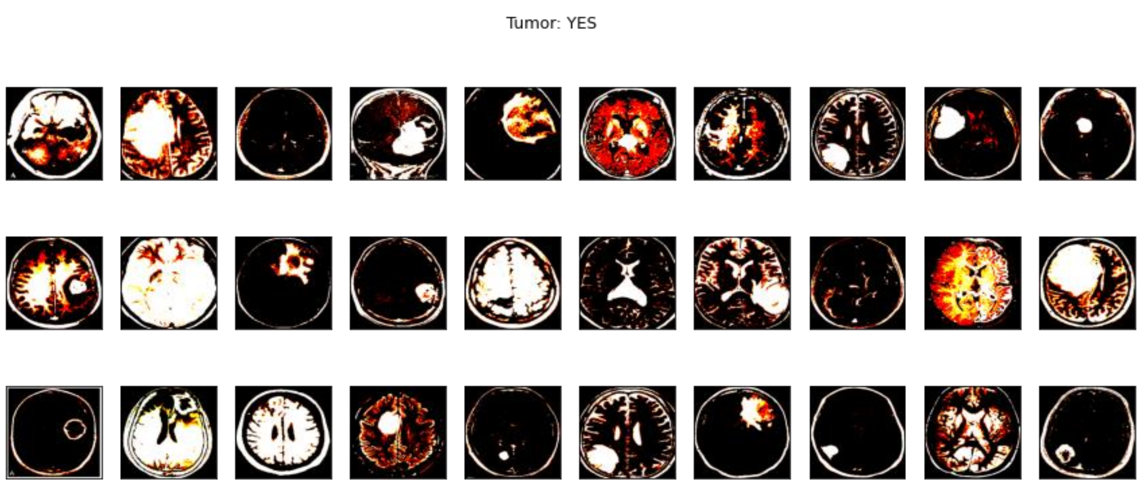


Figure 7: Binary Thresholding Is Applied, And The Tumor Is Not Present

Figure 8 depicts the brain images which contain the tumors after using the binary thresholding technique. The first set of images are transformed into the RGB variant, and the second set are presented in the dark variant.



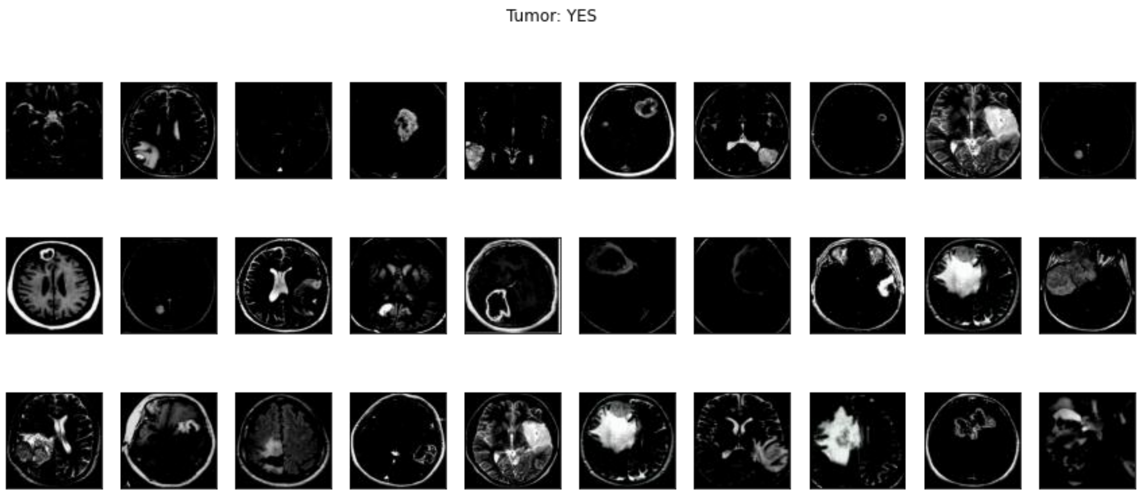


Figure 8: Binary Thresholding Is Applied, And The Tumor Is Present

2.4 Image Segmentation

Image segmentation [10] is based on locating comparable objects inside an image and grouping them by calculating the photos that share the most similarities or dissimilarities. Generally, it separates components from the remainder of the images so they can be observed as objects.

There are five types of image segmentation techniques, including thresholding, edge-based segmentation, region-based segmentation, watershed segmentation, cluster-based segmentation, and neural network segmentation. Alireza et al. [44] and Yu Jin [45] also published general image segmentation surveys focusing mainly on image segmentation with MRI images. After carefully reviewing the literature, the watershed algorithm [6][14] was selected to be used in this project as it offers lower computational time and provides closed contours of the brain images.

2.5 Feature Extraction

Feature extraction [4] is a type of dimensionality reduction technique where many image pixels are represented so that crucial parts of the image are captured effectively. Feature extraction aims to find out the anomalies. As to classify the photos using a classifier that requires the mentioned characteristics to be trained, we must extract specific features from the images such as entropy, Root Mean Square (RMS), Smoothness, Skewness, Symmetry, Kurtosis, Mean, Texture, Variance, Centroid, Central Tendency, Inverse Difference Moment (IDM), Correlation, Energy, Homogeneity, Dissimilarity, Contrast, Shade, Prominence, Eccentricity, etc.

Anne's [47] paper surveyed feature extraction classes such as statistical, structural, model, and graph-based approaches. From the survey and other papers [48], the statistical approach shows promising results for the brain images. One of the best methods of the statistical approach is Gray Level Co-occurrence Matrix (GLCM) [47]. GLCM is a statistical technique for analyzing texture that considers the spatial arrangement of pixels. GLCM shows better performance where surfaces are separated easily [47].

Moreover, when processing images, the GLCM-based approach offers an acceptable performance in terms of processing time and complexity [48]. We will leverage GLCM as the feature extraction method in this project.

2.6 Training and Testing

The aim is to split the data set into training and testing groups. Whenever we are slicing the data set into training and testing, we need to take care of the following conditions:

- The data collection must be big enough to train a good model with high accuracy.
- Choose a test set that shares no attributes with the training set.

Both characteristics mentioned above are met for this project, and we aim to develop a model that applies well to the new data. In this project, we split the dataset into 80%, 10 %, and 10% for training, testing, and validation, respectively. This slicing has been carried out after reviewing the literature [13-17]. We will also leverage VGG16, ResNet50, and Inception v3 as training models in our project. After the model has been trained, we will verify and fine-tune the parameters before testing the model on the test dataset.

CHAPTER THREE

DEEP LEARNING

Deep learning [21][5] is a subfield of machine learning where the algorithms are guided like the structure of the human brain and are based on the foundation of neural networks (NN). NN takes in data, educates itself to identify patterns, and then forecasts the results for comparable fresh data collection. One of the main applications of deep learning models is to leverage them for classification problems. In general, classification models are classified into two categories: discriminative models based on classical learning, such as support vector machines (SVM) and random forests (RF), and the second category is neural network-based classifiers [42]. Our focus in this project will be on CNN models, which are variant of deep learning-based networks. Convolutional neural networks handle complex data for face identification, picture classification, and other image-related tasks. CNN is a unique 3D structure with a unique NN extracts vital features and then utilizes those features to categorize the image. The convolutional layer serves as the foundation of a CNN architecture. Figure 9 replicates the Single Layer CNN architecture; the CNN consists of the other layers; the first one is the input layer which will take the raw pixel value of the input image as input. The second layer is the convolutional layer; the convolutional layer learns visual characteristics from tiny input data squares, preserving the link between pixels. The performance of operations like edge detection, blur, and sharpening is feasible by applying filters to the convolution of a picture. The third layer is the activation layer which will produce a single output based on the weighted sum of inputs. The fourth layer is the pooling layer; when the images are enormous, the pooling layer will reduce the number of parameters. Spatial pooling can be of different types, such as max pooling, which will take the most prominent element; average pooling, which will take the average of the elements present in the feature map and last one is sum pooling which will take the summation of the elements. The output matrix from the pooling layer is converted into a vector at the final convolutional layer using fully linked layers. The main advantage of CNN models is their high accuracy, as well as the minimal level of image pre-processing required for the training data. There are some drawbacks for the CNN models which includes the need for a large training dataset.

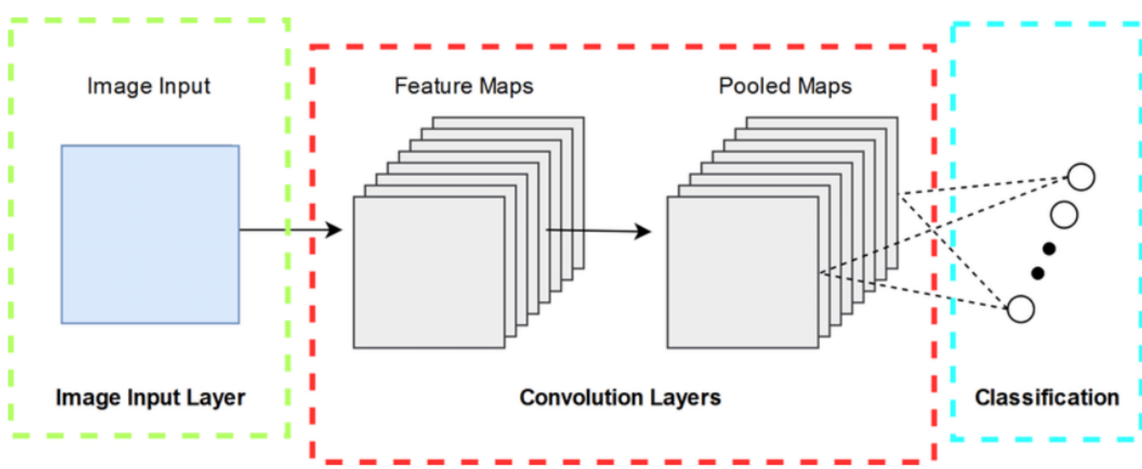


Figure 9: Single Layer CNN

3.2 Transfer Learning

Transfer learning [20] is the process of transferring as much information as possible from one task of the model to the current task. With transfer learning, a computer can use its understanding of one activity to better generalize about another. Transfer learning is primarily used in computer vision, and natural language processing tasks like sentiment analysis require massive computational power. Transfer learning [20] offers several advantages, but its key benefits include reducing training time, improving neural network performance (in most circumstances), and not requiring much training data.

3.2.1 VGG-16

VGG 16 is a model which consists of a 16-layer CNN model. VGG-16 is still considered one of the best and most effective models for detection and classification. Figure 10 represents a standard VGG 16 architecture. With a 3 * 3 kernel size, ConvNet layers is the main focus of the VGG 16 model architecture. The value of this model is that it can be downloaded for free from the internet and used in systems and apps. It distinguishes itself from other developed models by being straightforward. The minimum input picture size that this CNN model supports is 224 * 224 pixels with three channels. Optimization algorithms are employed in neural networks to assess whether a neuron needs to be activated by calculating the weighted total of the input. Artificial neural network input gains nonlinearity from the input layer and activation function, enabling it to learn and carry out challenging tasks.

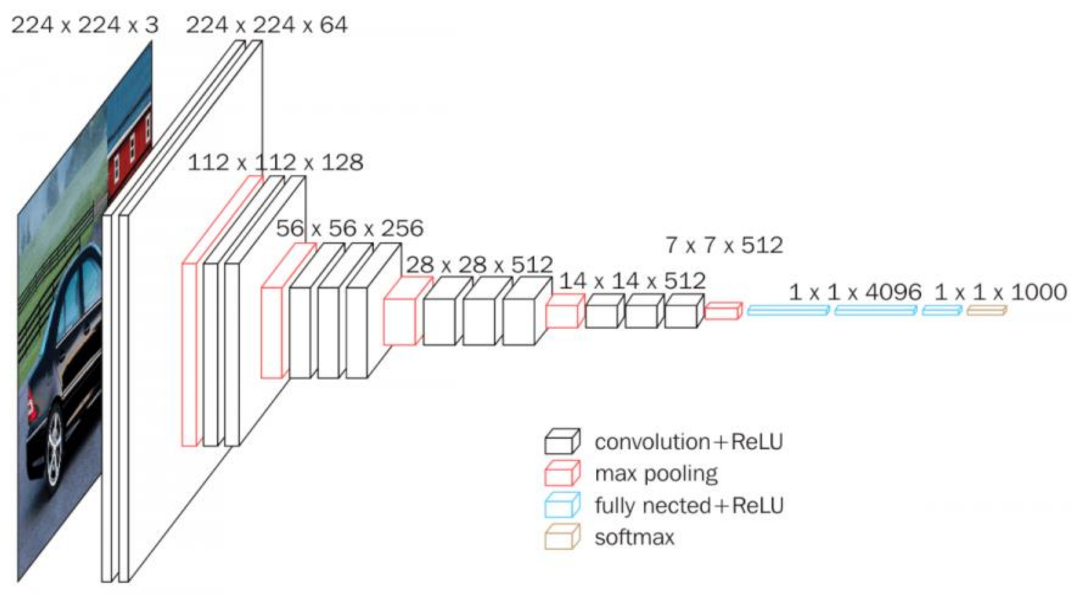


Figure 10: VGG-16 Architecture

3.2.2 Resnet-50

Resnet50 is a 50-layer CNN model; out of 50 layers, 48 are Convolutional layers, one is the max pool layer, and one is the intermediate pool layer. The primitive version was Resnet 34, which consists of only 34 layers. Each convolutional network in the primary network featured 3*3 filters and was based on the VGG neural network (VGG-16 and VGG-19).

Figure 11 represents the Resnet50 architecture where (a) represents the stem block, (b) represents Stage-1 Block 1, (c) represents Stage-1 Block 2, and (d) is fully connected block. The ResNet architecture follows two basic design ideas. First, regardless of the output feature map's size, each layer has the same filters. Second, it contains twice as many filters to preserve each layer's temporal complexity if the feature map's size is reduced by half.

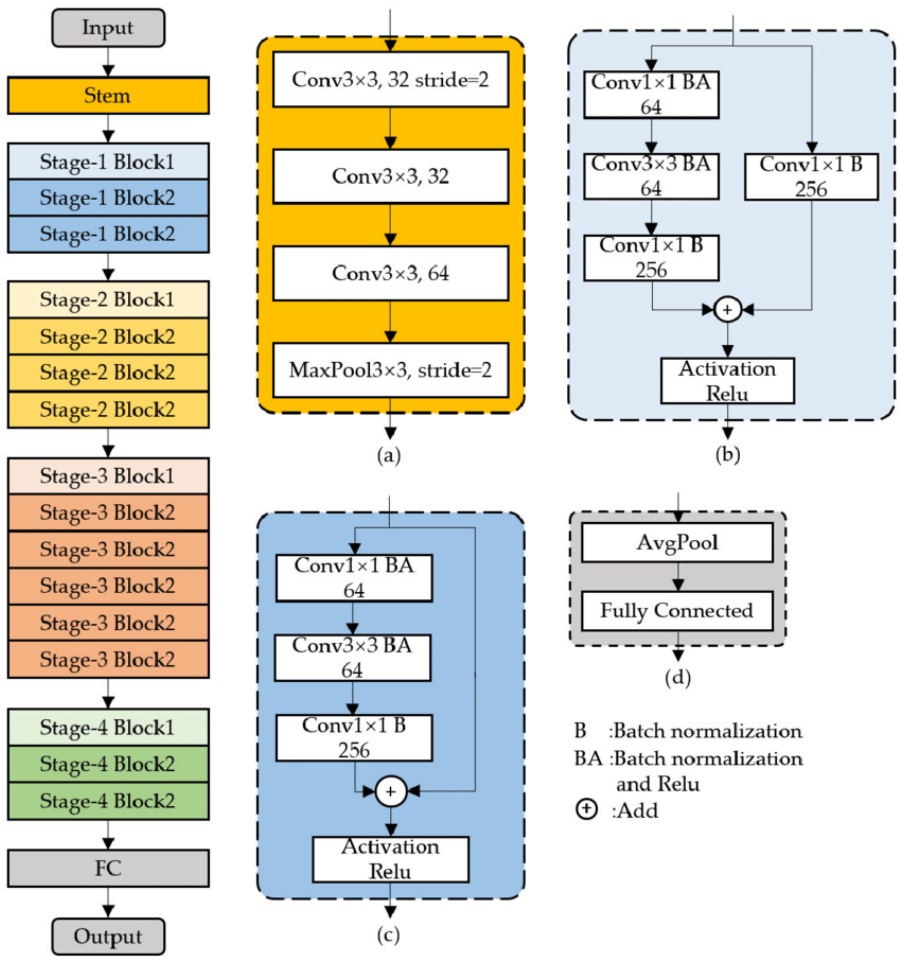


Figure 11: Resnet 50 Architecture

3.2.3 Inception V3

Inception was introduced and developed by Google Net in 2014, trained on the ImageNet database. Inception was a pre-trained CNN model of 22 layers, including 5M parameters with a kernel (filter) dimension of 1×1, 3×3, and 5×5 to capture handcrafted features at different scales, including the max-pooling layer. Filters with 1x1 kernels conserve computation time while having a less adverse impact on network performance. Google upgraded the Inception network later in 2015 to Inception-v3, where Conv layers are rescaled to decrease with hyperparameters. Inception-v3 is a 48-layer deep neural network; the inception-v3 network requires an input image of dimension 299×299. Figure 12 shows the basic architecture of the Inceptionv3 model. The model comprises symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, dropouts, and fully connected layers. The model uses the batch norm, which is also applied to the activation inputs. SoftMax is used to calculate the output probabilities.

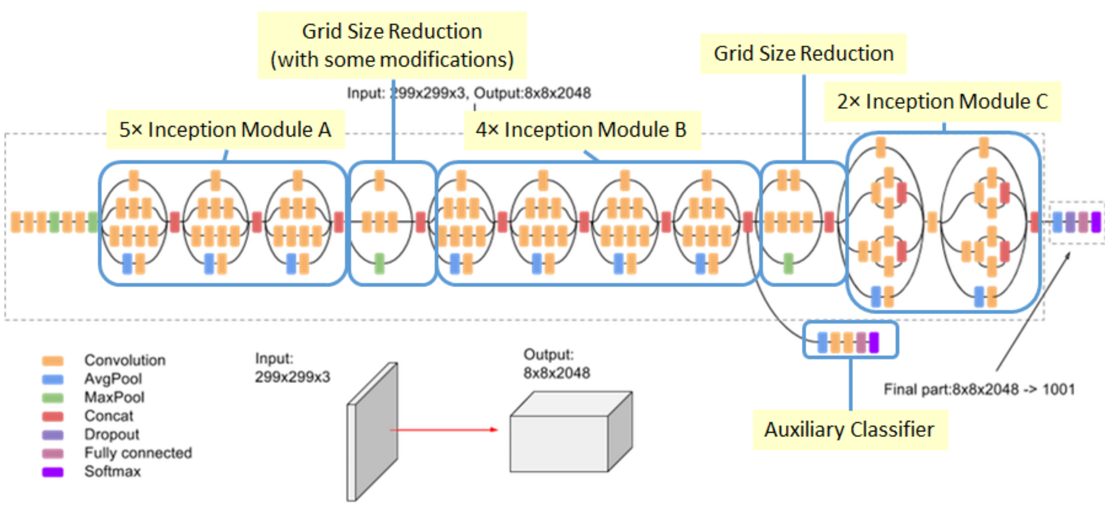


Figure 12: Inception V3 Architecture

3.3 Hyperparameter Tuning

Hyperparameter tuning determines a set of optimum hyperparameter values for a machine learning algorithm and then applies this adjusted algorithm to any data collection. Hyperparameter adjustment is essential as they govern a machine learning model's accuracy. Different hyperparameters can be configured for every machine-learning model. During training, a model starts with random parameter values and modifies them as needed. We define different parameters for tuning the deep learning-based models, such as the batch size; the network's architecture (number of neurons and number of layers), dropout rate, loss model, etc. There are promising methods to tune the parameters, such as Manual Search, Grid Search, Bayes Optimization, etc. In this project, we will leverage Bayesian optimization [[2]] as the tuning method.

Bayesian Optimization consists of three core components; the first one is choosing the search space, the second is the objective function, and the third is the acquisition function. The objective function simply takes in a set of hyperparameters and outputs a score that indicates how well a set of hyperparameters performs on the validation set. The acquisition function estimates the objective function, which can be used to direct future sampling. The acquisition function is the one by which the current posterior is used to select the sample from the search space and the chosen point is those with the optimum value of acquisition function. The main difference between the acquisition and objective technique is that the acquisition function will guide how the parameter space should be explored. However, the objective function will move towards the optimal parameter values.

Bayesian Optimization can be summarized as follows; firstly, select the sample by optimizing the acquisition function. Secondly, evaluate the sample with the objective function. Thirdly, update the hyperparameter values using the acquisition function.

CHAPTER FOUR

EVALUATION

4.1 Dataset

The dataset we used for the brain tumor detection problem is available on Kaggle [22]. This dataset contains almost 3700 JPEG files from BRATS 2015, of which 3300 images have the tumor. We need to take care of the following conditions while slicing the data set into training and testing:

- Data set should be large enough to yield meaningful results.
- Don’t pick the test set with different traits than the training set.

The slicing has been done after considering the above two conditions as well as referring to the numerous research papers [13-17]. We have divided the data into three sets, including 80%, 10 %, and 10% for training, testing, and validation sets, respectively. [17]

4.2 System Configuration

Hardware Requirement

- Memory: 4 GB RAM (minimum)
- NVIDIA GeForce GTX 970/AMD Radeon RX480/ Apple M1 Chip
- CPU: Intel Core i5 or above
- OS: Linux, Windows, Mac OS.

Software and Language Requirement

Google Colab

Google Colab is a cloud-based platform which provides vital GPU resources for Machine-Learning projects. We have used the google colab for this project to perform the computations.

Python

Python is a widely used programming language in which offers numerous benefits such as platform independence, flexibility, wide community, and rich library. Libraries used in this project include NumPy, TensorFlow, cv2, Keras, PyTorch, plotly, shutil, itertools, imutils, matplotlib.

4.3 Evaluation Metrics

Evaluation metrics are used to gauge how well a statistical or machine-learning model performs. We will use a confusion matrix to summarize the model's overall performance.

Confusion Matrix:

The confusion matrix gives a more insightful picture of a predictive model's performance, showing which classes are forecasted correctly or erroneously. Figure 13 shows how the four-categorization metrics (i.e., TP, FP, FN, and TN) are produced and how our predicted values compare to the actual prediction values.

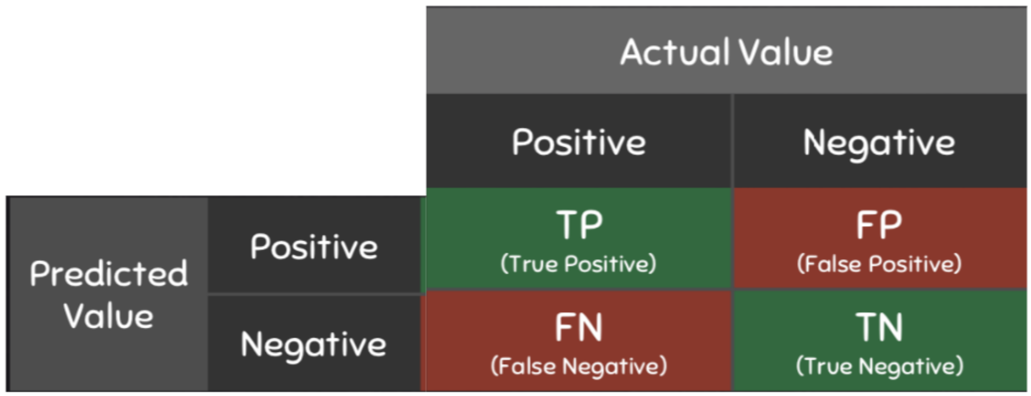


Figure 13: Confusion Matrix Overview

- Accuracy: Accuracy is the number of correct predictions divided by the total number of predictions.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision is defined as the number of true positives divided by the summation of true and false positives.

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

- Recall or Sensitivity is the data's ratio of true positives to total (actual) positives.

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

- F_1 Score is the harmonic mean of precision and recall.

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

4.4 Simulation Result

Figure 14 and Figure 15 present the training and validation accuracy of VGG-16 and the training and validation loss of VGG16, respectively. The training accuracy graph indicates that the Area Under Curve (AUC) is not constant and becomes increasingly non-linear as the number of epochs are increased. The validation accuracy of AUC in VGG16 achieved 96.00%, and the training accuracy is 97.5%.

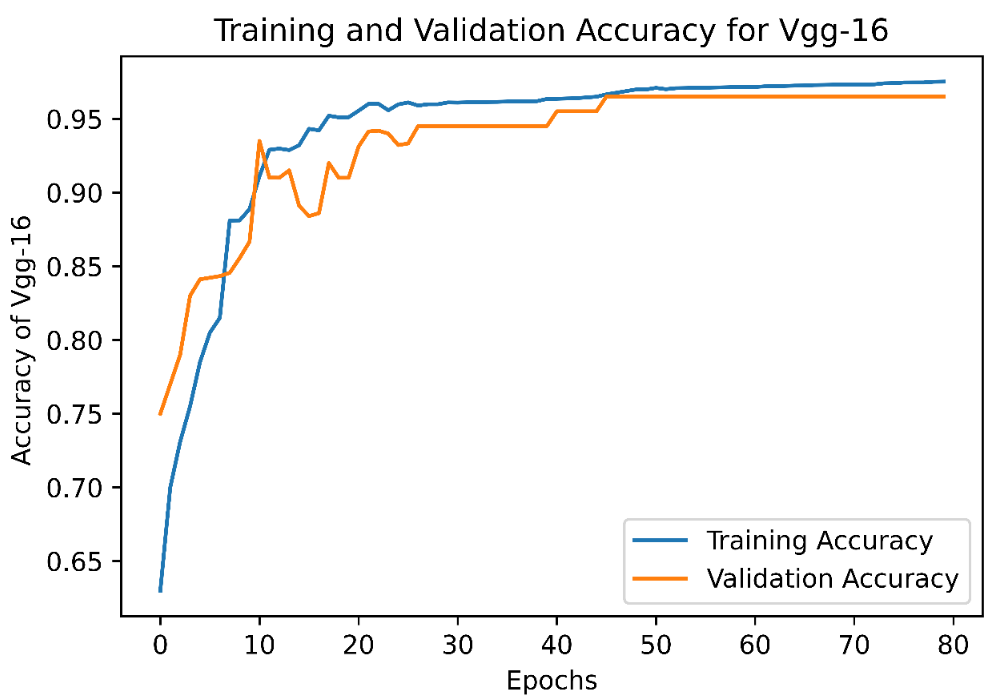


Figure 14: Training And Validation Accuracy Of VGG16

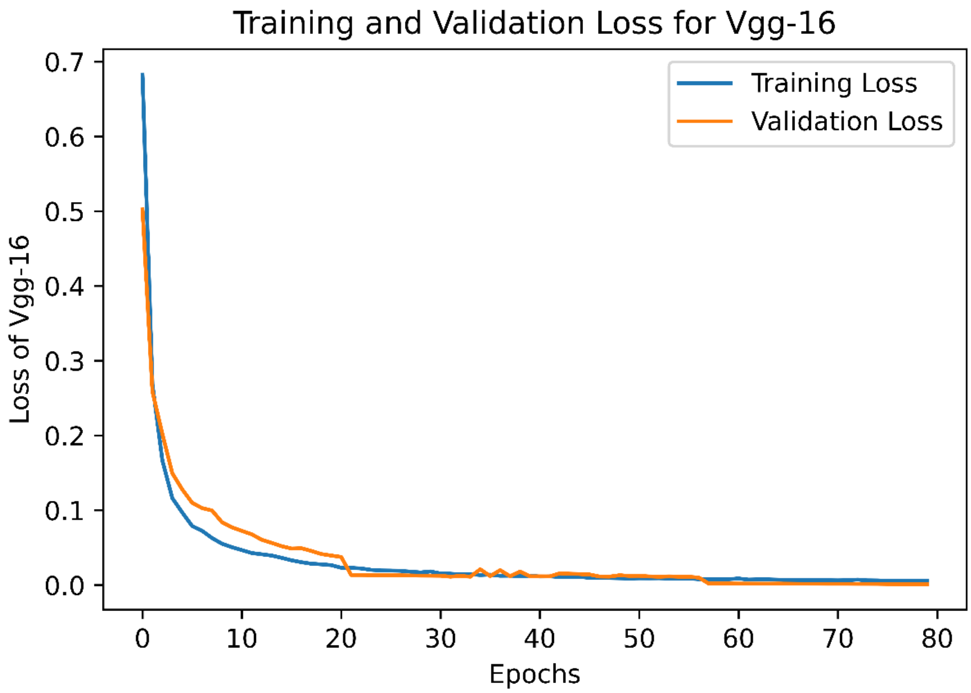


Figure 15: Training And Validation Loss Of VGG-16

Similarly, once the epochs are increasing, AUC for the training loss and validation loss are decreased. Moreover, after 50 epochs the validation loss and training loss converge to the final value. This shows that the model has been trained properly.

Table 1 illustrates how each model performed before applying the hyperparameter tuning approach. Reviewing the results indicate that we achieved 96.0% accuracy via VGG16, 94% via Resnet 50, and 90.7% via Inception V3 models. We have also presented the results using different evaluation metrics such as precision, recall, and F1 score.

Table 1: Comparison Of The Models (Before Hyperparameter Tuning)

	VGG 16	Resnet50	Inception V3
Training Accuracy	0.96	0.94	0.907
Test Accuracy	0.93	0.90	0.89
Precision	0.9	0.8	0.9
Recall	0.95	0.8	0.9
F_1 Score	0.9	0.82	0.9

Table 2 shows the accuracy of different models using the best set of hyperparameters such as learning rate, number of dense layers, convolutional filters, type of activation function, and type of optimizer obtained by the Bayesian Optimization technique. Interestingly, after applying the hyperparameter tuning, the accuracy of VGG-16 model was improved to 97.5%. This clearly shows how implementing a systematic tuning technique can further improve the initial results for the tumor detection problem.

Table 2: Comparison Of The Models (After Hyperparameter Tuning)

	VGG-16	Resnet50	Inception V3
Accuracy	97.5%	95%	91.5%
Learning Rate	0.00146	0.00146	0.00142
Convolutional Filters	5	4	10
Dense Layer	1	2	1
Drop-Out Rate	0.2	0.3	0.5

Activation	Relu	Relu, swish, and SoftMax	Relu and SoftMax
Optimizer	Adam	RMSProp	Adamax

CHAPTER FIVE

CONCLUSION

In this project, we used three pre-trained CNN models, i.e., VGG-16, Resnet 50, and Inception V3, for tumor detection problem. We leveraged the Kaggle dataset taken from BRATS 2015 for training the models. The simulation results indicate that applying a systematic hyperparameter tuning technique, i.e., Bayesian optimization, can improve the accuracy results even further for the transfer learning models. We obtained the optimal values for different set of hyperparameters such as learning rate, number of dense layers, convolutional filters, type of activation function, and type of optimizer. Moreover, VGG-16 model provided the highest detection accuracy, i.e., 97.5%, after applying the hyperparameter tuning. Results also indicated that using transfer learning models can be promising solution for brain tumor detection problem.

REFERENCES

1. Anitha, V.; Murugavalli, S. Brain tumor classification using two-tier classifier with adaptive segmentation technique. IET Comput.Vis. 2016, 10, 9–11.
2. M.Subramanian, N Prasad, Janakiramaiah B, Mohan B,Satishkumar V , ”Hyperparameter Optimization for Transfer Learning of VGG-16 for Disease Identification in Corn Leaves using Bayesian Optimization”
3. WY Lee, SM Park, and KB Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," *Optics (Stuttg).*, vol. 172, no. July, pp. 359–367, 2018. https://doi.org/10.1016/j.ijleo.2018.07.044
4. A. Yang, X. Yang, W. Wu, H. Liu, and Y. Zhuansun, "Research on feature extraction of tumor image based on convolutional neural network," *IEEE Access*, vol. 7, pp. 24204–24213, 2019. https://doi.org/10.1109/ACCESS.2019.2897131
5. N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib, "A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor," *IEEE Access*, vol. 8, pp. 55135–55144, 2020. https://doi.org/10.1109/ACCESS.2020.2978629
6. MA Khan *et al.*, "Brain tumor detection and classification: A framework of marker-based watershed algorithm and multilevel priority features selection,"

Microsc. Res. Tech., Vol. 82, no. 6, pp. 909–922, 2019. https://doi.org/10.1002/jemt.23238

7. Mansi Lather Parvinder Singh Dr. “Investigating Brain Tumor Segmentation and Detection Techniques”
8. P Gokila Brindha, M Kavinraj, P Manivasakam, and P Prasanth, “Brain Tumor Detection using MRI Images”
9. Marina Chane “AI for medical imaging: Data Augmentation & Image pre-processing” https://www.imaios.com/en/Company/blog/AI-for-medical-imaging-data-augmentation
10. Yuting Lu, Tao Zhou, and Shengwen Fu,” An Improved Watershed Segmentation Algorithm” https://iopscience.iop.org/article/10.1088/1757-899X/677/4/042028
11. Yingbo Liang,”Water Shed Algorithm for Medical Image Segmentation” https://www.worldscientific.com/doi/10.1142/S0218001419540193
12. Ayesha Y., Li Qiang, Charles Okanda, Mohammed J., Halima Bello, “Brain Tumor Analysis using Deep Learning and VGG-16”
13. N.Preethi, Aravind Vasudevan,“ Analysis of MRI Images to discover Brain Tumor using CNN and VGG-16” https://link.springer.com/chapter/10.1007/978-981-19-2541-2_32
14. Vaibhav Singh, Sarthak Sharma, Neetu Garg, “Brain Tumor Prediction using VGG-16” https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119752134.ch9

15. M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville and Y. Bengio, et al. Brain tumor segmentation with deep neural networks, Medical Image Analysis 35 (2017), 18-31.
16. C.Hari Prasad, S.Shafiulla Basha, P. Sandhya, K.Harish, “Brain Tumor Segmentation using RESNET 50” https://www.jetir.org/view?paper=JETIR2204638
17. Javeria A., Muhammad Anjum, Muhammad Sharif, Saima Jaaben, “Brain Tumor Detection using Inception v3” https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9023211/
18. Niklas Donges , http://builtin.com/data-science/transfer-learning
19. Sinno Pan and Qiang Yang, https://www.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf
20. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; M.I.T. Press: Cambridge, MA, USA, 2016.
21. Bangalore Yogananda, C.G.; Das, Y.; Wagner, B.C.; Nalawade, S.S.; Reddy, D.; Holcomb, J.; Pinho, M.C.; Fei, B.; Madhuranthakam, A.J.; Maldjian, J.A. Disparity Autoencoders for Multi-class Brain Tumor Segmentation. In International MICCAI Brainlesion Workshop; Springer: Cham, Switzerland, 2022; pp. 116–124.
22. Urban, G.; Bendszus, M.; Hamprecht, F.; Kleesiek, J. Multi-modal brain tumor segmentation using deep convolutional neural networks. In Proceedings of the BRATS-MICCAI 2014, Boston, MA, USA, 14 September 2014.

23. Al-Araj, R.S.A.; Abed, S.K.; Al-Ghoul, A.N.; Abu-Naser, S.S. Classification of Animal Species Using Neural Network. Int. J. Acad. Eng. Res. **2020**, 4, 23–31.
24. Sharma,P., Shukla, A.P. A Review on Brain Tumor Segmentation and Classification for MRI Images. In Proceedings of the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 30–31 December 2021
25. Rao, C.S.; Karunakara, K. A comprehensive review on brain tumor segmentation and classification of MRI images. Multimed. Tools Appl. 2021, 80, 17611–17643.
26. Magadza, T.; Viriri, S. Deep Learning for Brain Tumor Segmentation: A Survey of State-of-the-Art. J. Imaging 2021, 7, 19
27. Tiwari, A.; Srivastava, S.; Pant, M. Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019. Pattern Recognit. Lett. 2020, 131, 244–260
28. Kumari, N.; Saxena, S. Review of Brain Tumor Segmentation and Classification. In Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, 1–3 March 2018.
29. Pohle, R.; Toennies, K.D. Segmentation of medical images using adaptive region growing. In Medical Imaging 2001: Image Processing; Sonka, M.; Hanson, K.M., Eds.; SPIE: Bellingham, WA, USA,2001

30. Hooda, H.; Verma, O.P.; Singhal, T. Brain tumor segmentation: A performance analysis using K-Means, Fuzzy C-Means and Region growing algorithm. In Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 8–10 May 2014.
31. Bal, A.; Banerjee, M.; Sharma, P.; Maitra, M. Brain Tumor Segmentation on MR Image Using K-Means and Fuzzy-Possibilistic Clustering. In Proceedings of the 2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), Kolkata, India, 4–5 April 2018.
32. Ucuzal H, Yaşar Ş, Çolak C (2019) Classification of brain tumor types by deep learning with convolutional neural network on magnetic resonance images using a developed web-based interface. In: 3rd international symposium on multidisciplinary studies and innovative technologies (ISMSIT).
33. Kermi, A., Andjouh, K., & Zidane, F. (2018). Fully automated brain tumour segmentation system in 3D-MRI using symmetry analysis of brain and level sets. IET Image Processing, 12(11), 1964-1971
34. .Anitha, V., & Murugavalli, S. (2016). Brain tumour classification using two-tier classifier with adaptive segmentation technique. IET computer vision, 10(1), 9-17

35. .Sallemi, L., Njeh, I., & Lehericy, S. (2015). Towards a computer aided prognosis for brain glioblastomas tumor growth estimation. IEEE transactions on nanobioscience, 14(7), 727-733

36. Machhale, K., Nandpuru, H. B., Kapur, V., & Kosta, L. (2015, May). MRI brain cancer classification using hybrid classifier (SVM-KNN). In 2015 International Conference on Industrial Instrumentation and Control (ICIC) (pp. 60-65). IEEE.

37. Fausto Milletari Seyed-Ahmad Ahmadi Christine Kroll Annika Plate Verena Rozanski Juliana Maiostre Johannes Levin Olaf Dietrich Birgit Ertl-Wagner Kai Bötzel, Nassir Navab 2016 Houghin MRI and ultrasound Elsevier Inc

38. R.Tamilarasi, S. Gopinathan 2021,VGG-16, Inception and ResNet Architecture for Brain Image Detection

39. Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Rad, A. E., Rehman, A., & Uddin, M. (2014). Medical image segmentation methods, algorithms, and applications. IETE Technical Review, 31(3), 199-213

40. Zhang, Yu Jin. (2001). A survey on evaluation methods for image segmentation. Pattern recognition 1335-1346.

41. Clarke, L. P., et al. "MRII segmentation: methods and applications." Magnetic resonance imaging 13.3 (1995): 343- 368.

42. Anne, “Texture Feature Extraction Methods:A Survey”

43. M. Mehri, M., Héroux, P. Gomez-Krämer, and R. Mullot, ‘‘Texture feature benchmarking and evaluation for historical document image analysis,’’ Int. J. Document Anal. Recognit., vol. 20, no. 1, pp. 1–35, Mar. 2017.