BRAIN TUMOR DETECTION USING IMAGE SEGEMENTATION

A PROJECT REPORT

Submitted by

AAGAM SHAH [Reg No: RA1911027010015]
SASWATA LAHIRI[Reg No: RA1911027010001]

Under the Guidance of

Dr. P.RAJASEKAR
(Assistant Professor, Department of Data Science and Business Systems)

In partial fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY
COMPUTER SCIENCE ENGINEERING
with specialization in (BIG DATA ANALYTICS)

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603203

NOVEMBER2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this project report titled "**BRAIN TUMOR DETECTION USING IMAGE SEGEMENTATION"** is the bonafide work of **"AAGAM SHAH [Reg No: RA1911042010015] and SASWATA LAHIRI [Reg No: RA1911042010001]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

<<Signature of the Supervisor>>	<<Signature>>
Dr. P.Rajasekar **GUIDE** Assistant Professor Dept. of DSBS	Dr. M.Lakshmi **HEAD OF THE DEPARTMENT** Dept. of DSBS
Signature of Internal Examiner	Signature of External Examiner

ABSTRACT

The human brain is the major controller of the humanoid system. The abnormal growth and division of cells in the brain lead to a brain tumor, and the further growth of brain tumors leads to brain cancer. In the area of human health, Computer Vision plays a significant role, which reduces the human judgment that gives accurate results. CT scans, X-Ray, and MRI scans are the common imaging methods among magnetic resonance imaging (MRI) that are the most reliable and secure. MRI detects every minute objects. Our paper aims to focus on the use of different techniques for the discovery of brain cancer using brain MRI. In this study, we performed pre-processing using the bilateral filter (BF) for removal of the noises that are present in an MR image. This was followed by the binary thresholding and Convolution Neural Network (CNN) segmentation techniques for reliable detection of the tumor region. Training, testing, and validation datasets are used. Based on our machine, we will predict whether the subject has a brain tumor or not. The resultant outcomes will be examined through various performance examined metrics that include accuracy, sensitivity, and specificity. It is desired that the proposed work would exhibit a more exceptional performance over its counterparts.

KEYWORDS: Brain tumor, Magnetic resonance imaging, Adaptive Bilateral Filter, Convolution Neural Network.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department**, Dr M. Lakshmi** Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program coordinators **Dr.G.Vadivu,** Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor**, DR.T Karthick,** Assistant professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. T Veeramakali ,** Assistant Professor, Department of Data Science and Business Systems, for providing me with an opportunity to pursue my project under her mentorship. She provided me with the freedom and support to explore the research topics of my interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Data Science and Business Systems staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Aagam Shah

Saswata lahiri

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities.

The medical imaging processing refers to handling images by using the computer. This processing includes many types of techniques and operations such as image gaining, storage, presentation, and communication. This process pursues the disorder identification and management. This process creates a data bank of the regular structure and function of the organs to make it easy to recognize the anomalies. This process includes both organic and radiological imaging which used electromagnetic energies (X-rays and gamma), sonography, magnetic, scopes, and thermal and isotope imaging. There are many other technologies used to record information about the location and function of the body. Those techniques have many limitations compared to those modulates which produce images.

An image processing technique is the usage of a computer to manipulate the digital image. This technique has many benefits such as elasticity, adaptability, data storing, and communication. With the growth of different image resizing techniques, the images can be kept efficiently. This technique has many sets of rules to perform in the images synchronously. The 2D and 3D images can be processed in multiple dimensions.

The focus of this project is MR brain images tumor extraction and its representation in simpler form such that it is understandable by everyone. The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. The aim of this work is to define an algorithm that will result in extracted image of the tumor from the MR brain image.

1.1 GENERAL:

The brain tumor is one all the foremost common and, therefore, the deadliest brain diseases that have affected and ruined several lives in the world. Cancer is a disease in the brain in which cancer cells ascends in brain tissues. Conferring to a new study on cancer, more than one lakh people are diagnosed with brain tumors every year around the globe. Regardless of stable efforts to overcome the complications of brain tumors, figures show unpleasing results for tumor patients. To contest this, scholars are working on computer vision for a better understanding of the early stages of tumors and how to overcome using advanced treatment options.

Magnetic resonance (MR) imaging and computed tomography (CT) scans of the brain are the two most general tests to check the existence of a tumor and recognize its position for progressive treatment decisions. These two scans are still used extensively for their handiness, and the capability to yield high-definition images of pathological tissues is more. At present, there are several other conducts offered for tumors, which include surgery, therapies such as radiation therapy, and chemotherapy. The decision for which treatment relies on the many factors such as size, kind, and grade of the tumor present in the MR image. It’s conjointly chargeable for whether or not cancer has reached the other portions of the body.

Precise sighting of the kind of brain abnormality is enormously needed for treatment operations with a resolution to diminish diagnostic errors. The precision is often makeshift utilizing computer-aided diagnosis (CAD) systems. The essential plan of computer vision is to produce a reliable output, which is an associate estimation to assist medical doctors in image understanding and to lessen image reading time. These advancements increase the steadiness and correctness of medical diagnosis — however, segmenting an MR image of the tumor and its area itself a very problematic job. The occurrence of tumors in specific positions within the brain image without distinguishing picture intensities is an additional issue that makes a computerized detection of brain tumor and segmentation a problematic job

1.2 PURPOSE

Brain tumor is a life-threatening problem and hampers the normal functioning of the human body. For proper diagnosis and efficient treatment planning, it is necessary to detect the brain tumor in early stages. Digital image processing plays a vital role in analysis of medical images. Segmentation of brain tumor involves separation of abnormal brain tissues from normal tissues of brain. In the past, various researchers have proposed the semi and fully automatic methods for detection and segmentation of brain tumor. In this article, the different techniques available for segmentation have been presented. This PROJECT focuses on the work done by many researchers in the past to partially or fully automate the job of segmenting the brain tumor. The consolidated details of the reviewed literature have been tabulated. Simplicity and degree of human supervision decides the clinical acceptance of a particular segmentation technique.

1.2.1 SCOPE

Our aim is to develop an automated system for enhancement, segmentation and classification of brain tumors. The system can be used by neurosurgeons and healthcare specialists. The system incorporates image processing, pattern analysis, and computer vision techniques and is expected to improve the sensitivity, specificity, and efficiency of brain tumor screening. The primary goal of medical imaging projects is to extract meaningful and accurate information from these images with the least error possible. The proper combination and parameterization of the phases enables the development of adjunct tools that can help on the early diagnosis or the monitoring of the tumor identification and locations

1.2.2 NEED FOR BRAIN TUMOR DETECTION

Brain tumor occurs because of anomalous development of cells. It is one of the major reasons of death in adults around the globe. Millions of deaths can be prevented through early detection of brain tumor. Earlier brain tumor detection using Magnetic Resonance Imaging (MRI) may increase patient's survival rate. In MRI, tumor is shown more clearly that helps in the process of further treatment. This work aims to detect tumor at an early phase.

1.3 MOTIVATION

A brain tumor is defined as abnormal growth of cells within the brain or central spinal canal. Some tumors can be cancerous thus they need to be detected and cured in time. The exact cause of brain tumors is not clear and neither is exact set of symptoms defined, thus, people may be suffering from it without realizing the danger. Primary brain tumors can be either malignant (contain cancer cells) or benign (do not contain cancer cells). Brain tumor occurred when the cells were dividing and growing abnormally. It is appearing to be a solid mass when it diagnosed with diagnostic medical imaging techniques. There are two types of brain tumor which is primary brain tumor and metastatic brain tumor. Primary brain tumor is the condition when the tumor is formed in the brain and tended to stay there while the metastatic brain tumor is the tumor that is formed elsewhere in the body and spread through the brain. The symptom having of brain tumor depends on the location, size and type of the tumor. It occurs when the tumor compressing the surrounding cells and gives out pressure. Besides, it is also occurring when the tumor blocks the fluid that flows throughout the brain. The common symptoms are having headache, nausea and vomiting, and having problem in balancing and walking. Brain tumor can be detected by the diagnostic imaging modalities such as CT scan and MRI. Both of the modalities have advantages in detecting depending on the location type and the purpose of examination needed. In this paper, we prefer to use the MRI images because it is easy to examine and gives out accurate calcification and foreign mass location. The MRI is the most regularly utilized strategy for imaging brain tumors and the identification of its vicinity. The conventional technique for CT and MR image classification and detection of tumor cells remains largely supported for the human reviewing apart from different other methods. MR images are mainly used because there are non-destructive and non-ionizing. MR imaging offers high-definition pictures that are extensively utilized in discovering brain tumors. MRI has diverse schemes such as flair, T1-weighted, T2-weighted images. There are many image processing techniques such as pre-processing, segmentation of images, image improvements, feature extraction, and classifiers.

1.4 PROBLEM STATEMENT

Our study deals with automated brain tumor detection and classification. Normally the anatomy of the brain is analyzed by MRI scans or CT scans. The aim of the paper is tumor identification in brain MR images. The main reason for detection of brain tumors is to provide aid to clinical diagnosis. The aim is to provide an algorithm that guarantees the presence of a tumor by combining several procedures to provide a foolproof method of tumor detection in MR brain images. The methods utilized are filtering, erosion, dilation, threshold, and outlining of the tumor such as edge detection. The focus of this project is MR brain images tumor extraction and its representation in simpler form such that it is understandable by everyone.

1.5 RESEARCH OBJECTIVES

The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. The aim of this work is to define an algorithm that will result in extracted image of the tumor from the MR brain image. The resultant image will be able to provide information like size, dimension and position of the tumor, and its boundary provides us with information related to the tumor that can prove useful for various cases, which will provide a better base for the staff to decide the curing procedure. Finally, we detect whether the given MR brain image has tumor or not using Convolution Neural Network.

1.6 ORGANIZATION OF THE THESIS

In this document, chapter 2 consists about literature survey. The literature survey tells about the research done to work on the project. All the details about the papers, websites on which the research work is done in order to work on the project is provided in the literature survey. In chapter 3, we discuss about the various methodologies used in the project. In chapter 4, the details about experimental analysis is discussed. The experimental analysis includes sample code, result screenshots for a tested input image. In the next chapter we give the conclusion about the project and also provide information if the project can be implemented further or not. In the final chapter we provide all the references for this project.

CHAPTER 2

2. LITERATURE REVIEW

In Medical diagnosis, robustness and accuracy of the prediction algorithms are very important, because the result is crucial for treatment of patients. There are many popular classification and clustering algorithms used for prediction. The goal of clustering a medical image is to simplify the representation of an image into a meaningful image and make it easier to analyze. Several Clustering and Classification algorithms are aimed at enhancing the prediction accuracy of diagnosis process in detecting abnormalities. In the literature survey we provide a brief summary of the different methods that have been proposed for clustering over the period of 2002 to 2022.We have been though 25 papers each of which has a unique approach towards segmentation in some parameter or the other. The summaries of each of the papers are provided below.

- **Sivaramakrishnan And Dr. M. Karnan "A Novel Based Approach for Extraction Of Brain Tumor In MRI Images Using Soft Computing Techniques," International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2, Issue 4, April 2013.**

A Sivaramakrishnan et al. (2013) [1] projected an efficient and innovative discovery of the brain tumor vicinity from an image that turned into finished using the Fuzzy Capproach grouping algorithm and histogram equalization. The disintegration of images is achieved by the usage of principal factor evaluation is done to reduce the extent of the wavelet coefficient. The outcomes of the anticipated FCM clustering algorithm accurately withdrawn tumor area from the MR images.

- **Asra Aslam, Ekram Khan, M.M. Sufyan Beg, Improved Edge Detection Algorithm for Brain Tumor Segmentation, Procedia Computer Science, Volume 58,2015, Pp 430-437, ISSN 1877-0509**

M. M. Sufyan et al. [2] has presented a detection using enhanced edge technique for brain-tumor segmentation that mainly relied on Sobel feature detection. Their presented work associates the binary thresholding operation with the Sobel approach and excavates diverse extents using a secure contour process. After the completion of that process, cancer cells are extracted from the obtained picture using intensity values.

- **B.Sathya and R.Manavalan, Image Segmentation by Clustering Methods: Performance Analysis, International Journal of Computer Applications (0975 – 8887) Volume 29– No.11, September 2011**

Sathya et al. (2011) [3], provided a different clustering algorithm such as K-means, Improvised K-means, C-means, and improvised C-means algorithms. Their paper presented an experimental analysis for massive dat=asets consisting of unique photographs. They analyzed the discovered consequences using numerous parametric tests.

- **Devkota, B. & Alsadoon, Abeer & Prasad, P.W.C. & Singh, A.K. & Elchouemi, A. (2018). Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction. Procedia Computer Science. 125. 115-123. 10.1016/j.procs.2017.12.017.**

B. Devkota et al. [4] have proposed that a computer-aided detection (CAD) approach is used to spot abnormal tissues via Morphological operations. Amongst all different segmentation approaches existing, the morphological opening and closing operations are preferred since it takes less processing time with the utmost efficiency in withdrawing tumor areas with the least faults.

- **K. Sudharani, T. C. Sarma and K. Satya Rasad, "Intelligent Brain Tumor lesion classification and identification from MRI images using a K-NN technique," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 777-780. DOI: 10.1109/ICCICCT.2015.7475384**

K. Sudharani et al. [5] presented a K- nearest neighbor algorithm to the MR images to identify and confine the hysterically full-fledged part within the abnormal tissues. The proposed work is a sluggish methodology but produces exquisite effects. The accuracy relies upon the sample training phase.

- **Kaur, Jaskirat & Agrawal, Sunil & Renu, Vig. (2012). A Comparative Analysis of Thresholding and Edge Detection Segmentation Techniques. International Journal of Computer Applications.vol. 39.pp. 29-34. 10.5120/4898-7432. 8**

Jaskirat Kaur et al. (2012) [6] defined a few clustering procedures for the segmentation process and executed an assessment on distinctive styles for those techniques. Kaur represented a scheme to measure selected clustering techniques based on their steadiness in exceptional tenders. They also defined the diverse performance metric tests, such as sensitivity, specificity, and accuracy.

- **Li, Shutao, JT-Y. Kwok, IW-H. Tsang and Yaonan Wang. "Fusing images with different focuses using support vector machines." IEEE Transactions on neural networks 15, no. 6 (2004): 1555-1561.**

J.T. Kwok et al. [7] delivered wavelet-based photograph fusion to easily cognizance at the object with all focal lengths as several vision-related processing tasks can be carried out more effortlessly when wholly substances within the images are bright. In their work Kwok et al. investigated with different datasets, and results show that presented work is extra correct as it does not get suffering from evenness at different activity stages computations.

- **M. Kumar and K. K. Mehta, "A Texture based Tumor detection and automatic Segmentation using Seeded Region Growing Method," International Journal of Computer Technology and Applications, ISSN: 2229-6093, Vol. 2, Issue 4, PP. 855-859 August 2011.**

Kumar and Mehta [8] proposed the texture-based technique in this paper. They highlighted the effects of segmentation if the tumor tissue edges aren't shrill. The performance of the proposed technology may get unwilling results due to those edges. The texture evaluation and seeded region approach turned into executed inside the MATLAB environment

- **Mahmoud, Dalia & Mohamed, Eltaher. (2012). Brain Tumor Detection Using Artificial Neural Networks. Journal of Science and Technology. 13. 31-39.**

Dalia Mahmoud et al. [9] presented a model using Artificial Neural Networks for tumor detection in brain images. They implemented a computerized recognition system for MR imaging the use of Artificial Neural Networks. That was observed that after the Elman community was used during the recognition system, the period time and the 9 accuracy level were high, in comparison with other ANNs systems. This neural community has a sigmoid characteristic which elevated the extent of accuracy of the tumor segmentation.

- **Marroquin J.L., Vemuri B.C., Botello S., Calderon F. (2002) An Accurate and Efficient Bayesian Method for Automatic Segmentation of Brain MRI. In: Heyden A., Sparr G., Nielsen M., Johansen P. (eds) Computer Vision — ECCV 2002. ECCV 2002. Lecture Notes in Computer Science, vol 2353.**
  Springer, Berlin, Heidelberg

L. Marroquin et al. [10] presented the automated 3d segmentation for brain MRI scans. Using a separate parametric model in preference to a single multiplicative magnificence will lessen the impact on the intensities of a grandeur. Brain atlas is hired to find nonrigid conversion to map the usual brain. This transformation is further used to segment the brain from nonbrain tissues, computing prior probabilities and finding automatic initialization and finally applying the MPM-MAP algorithm to find out optimal segmentation. Major findings from the study show that the MPM-MAP algorithm is comparatively robust than EM in terms of errors while estimating the posterior marginal. For optimal segmentation, the MPM-MAP algorithm involves only the solution of linear systems and is therefore computationally efficient.

- **Minz, Astina, and Chandrakant Mahobiya. "MR Image Classification Using Adaboost for Brain Tumor Type." 2017 IEEE 7th International Advance Computing Conference (IACC) (2017): 701-705.**

Astina minz et al. [11] implemented an operative automatic classification approach for brain image that projected the usage of the AdaBoost gadget mastering algorithm. The proposed system includes three main segments. Pre-processing has eradicated noises in the datasets and converted images into grayscale. Median filtering and thresholding segmentation are implemented in the pre-processed image.

- **Monica Subashini.M, Sarat Kumar Sahoo, "Brain MR Image Segmentation for TumorDetection using Artificial Neural Networks," International Journal of Engineering and Technology (IJET), Vol.5, No 2, Apr-May 2013.**

Monica Subashini and Sarat Kumar Sahoo [12] has suggested a technique for detecting the tumor commencing the brain MR images. They also worked on different techniques, which include pulse-coupled Neural Network and noise removal strategies for reinforcing the mind MRI images and backpropagation network for classifying the brain MRI images from tumor cells. They observed image enhancement and segmentation of the usage of their proposed technique, and the backpropagation network helps in the identification of a tumor in a brain MR image.

- **S. Li, J.T. Kwok, I.W Tsang, and Y. Wang, —Fusing Images with Different Focuses using Support Vector Machines, Proceedings of the IEEE transaction on Neural Networks, China, November 2007.**

Li et al. [13] report that edge detection, image segmentation, and matching are not easy to achieve in optical lenses that have long focal lengths. Previously, researchers have proposed many techniques for this mechanism, one of which is wavelet-based image fusion. The wavelet function can be improved by applying a discrete wavelet frame transform (DWFT) and a support vector machine (SVM). In this paper, the authors experimented with five sets of 256-level images. Experimental results show that this technique is efficient and more accurate as it does not get affected by consistency verification and activity level measurements. However, the paper is limited to only one task related to fusion, and dynamic ranges are not considered during the calculation.

- **Pan, Yuehao & Huang, Weimin & Lin, Zhiping & Zhu, Wanzheng & Zhou, Jiayin & Wong, Jocelyn & Ding, Zhongxiang. (2015). Brain tumor grading based on Neural Networks and Convolutional Neural Networks. Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. 2015. 699-702. 10.1109/EMBC.2015.7318458.**

Yuehao Pan et al., [14] has used brain MRI pix for getting useful statistics for classifying brain tumor. In their proposed method, they used Convolutional Neural Networks (CNN) algorithms for developing a brain tumor detection system. The performance of their CNN report is measured primarily based on sensitivity and specificity parameters, which have stepped forward when in comparison to the Artificial Neural Networks (ANN).

- **Abhista Bhandari & Jarrad Koppen & Marc Agzarian (2020). Convolution neural networks for brain tumor segmentation. Published a paper in SpringerOpen.2020**

Abhista Bhandari et al.,[15] has presented The introduction of quantitative image analysis has given rise to fields such as radiomics which have been used topredict clinical sequelae. One growing area of interest for analysis is brain tumours, in particular glioblastomamultiforme (GBM). Tumour segmentation is an important step in the pipeline in the analysis of this pathology.Manual segmentation is often inconsistent as it varies between observers. Automated segmentation has beenproposed to combat this issue. Methodologies such as convolutional neural networks (CNNs) which are machinelearning pipelines modelled on the biological process of neurons (called nodes) and synapses (connections) havebeen of interest in the literature. We investigate the role of CNNs to segment brain tumours by firstly taking aneducational look at CNNs and perform a literature search to determine an example pipeline for segmentation. Wethen investigate the future use of CNNs by exploring a novel field—radiomics. This examines quantitative featuresof brain tumours such as shape, texture, and signal intensity to predict clinical outcomes such as survival andresponse to therapy.

CHAPTER 3

3. PURPOSED METHODOLOGY

Classification is the best approaches for identification of images like any kind of medical imaging. All classification algorithms are based on the prediction of image, where one or more features and that each of these features belongs to one of several classes. An automatic and reliable classification method Convolutional Neural Network (CNN) will be used since it is robust in structure which helps in identifying every minute details.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance to various aspects/objects in the image and be able to differentiate one from the other. The preprocessing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNet have the ability to learn these filters/characteristics.

A ConvNet is able to successfully capture the spatial and temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

We input the following packages:

- Sequential is used to initialize the neural network.
- Convolution2D is used to make the convolutional network that deals with the images.
- Batch Normalization
- MaxPooling2D layer is used to add the pooling layers.
- Flatten is the function that converts the pooled feature map to a single column that is passed to the fully connected layer.
- Dense adds the fully connected layer to the neural network.
- Glob module is used to retrieve files/pathnames matching a specified pattern.

SEQUENTIAL:

- To initialize the neural network, we create an object of the Sequential class.

BATCH NORMALIZATION:

- Batch normalization applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.
- Importantly, batch normalization works differently during training and during inference.

CONVOLUTION:

- To add the convolution layer, we call the add function with the classifier object and pass in Convolution2D with parameters. The first argument feature detectors which is the number of feature detectors that we want to create. The second and third parameters are dimensions of the feature detector matrix.
- We used 256 feature detectors for CNNs.
- The next parameter is input shape which is the shape of the input image. The images will be converted into this shape during pre-processing. If the image is black and white it will be converted into a 2D array and if the image is coloured it will be converted into a 3D array. · In this case, we'll assume that we are working with coloured images. If you are not using a GPU it's advisable to use lower dimensions to reduce the computation time. The final parameter is the activation function. Classifying images is a nonlinear problem. So, we use the rectifier function to ensure that we don't have negative pixel values during computation. That's how we achieve non-linearity.

POOLING:

- The Pooling layer is responsible for reducing the spatial size of the convolved feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.
- There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Generally, we use max pooling
- In this step we reduce the size of the feature map. Generally, we create a pool size of 2x2 for max pooling. This enables us to reduce the size of the feature map while not losing important image information

FLATTENING:

- In this step, all the pooled feature maps are taken and put into a single vector for inputting it to the next layer.
- The Flatten function flattens all the feature maps into a single long column.

FULLY CONNECTION:

- The next step is to use the vector we obtained above as the input for the neural network by using the Dense function in Keras. The first parameter is output which is the number of nodes in the hidden layer. You can determine the most appropriate number through experimentation. The higher the number of dimensions the more computing resources you will need to fit the model. A common practice is to pick the number of nodes in powers of two.
- The next layer we have to add is the output layer. In this case, we'll use the sigmoid activation function since we expect a binary outcome. If we expected more than two outcomes, we would use the SoftMax function.
- The output here is 1 since we just expect the predicted probabilities of the classes.

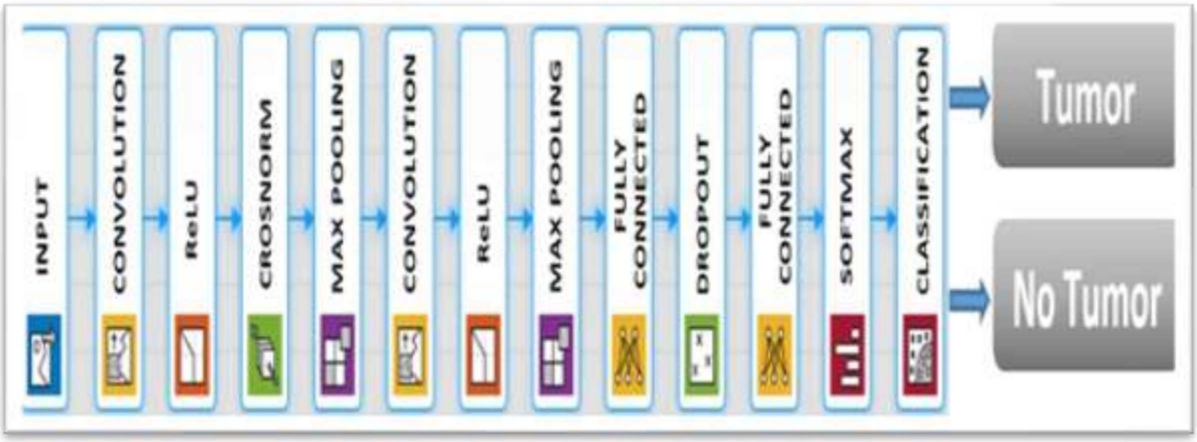


Figure 1 CNN ARCITECTURE

CHAPTER 4

4.1 MODULE/ EMPRICAL STUDY

This provides the architecture of the system that would be developed by our hands. It consists of six steps where the execution starts from taking an input image from the data set followed by the image pre-processing, image enhancement, Image segementation and the brain tumor classification using Convolutional Neural Network. Finally, the output is observed after all the abovementioned steps are completed. Each module is unique in its own way. Every step has its importance.

This architecture also includes a testing and training data set. The data set used is has been downloaded from Kaggle which consists of nearly 3060 images that are used to test and train the system.

4.1.1 MODULE 1: IMAGE PREPROCESSING AND IMAGE ENHANCEMENT:

4.1.1.1 IMAGE PREPROCESSING:

The Brain MRI image dataset has been downloaded from the Kaggle. The MRI dataset consists of around 3060 MRI images, including normal, benign, and malignant. These MRI images are taken as input to the primary step. The pre-processing is an essential and initial step in improving the quality of the brain MRI Image.

The critical steps in pre-processing are the reduction of impulsive noises and image resizing. In the initial phase, we convert the brain MRI image into its corresponding gray-scale image. The removal of unwanted noise is done using the adaptive bilateral filtering technique to remove the distorted noises that are present in the brain picture. This improves the diagnosis and also increase the classification accuracy rate.

4.1.1.2 IMAGE ACQUISITION FROM DATASET:

In image processing, image acquisition is done by retrieving an image from dataset for processing. It is the first step in the workflow sequence because, without an image no processing is possible. The image that is acquired is completely unprocessed. Here we process the image using the file path from the local device.

4.1.1.3 IMAGE ENHANCEMENT:

Image enhancement is a technique used to improve the image quality and perceptibility by using computer-aided software. This technique includes both objective and subjective enhancements. This technique includes points and local operations.

The local operations depend on the district input pixel values. Image enhancement has two types: spatial and transform domain techniques. The spatial techniques work directly on the pixel level, while the transform technique works on Fourier and later on the spatial technique. Edge detection is a segmentation technique that uses border recognition of strictly linked objects or regions. This technique identifies the discontinuity of the objects.

This technique is used mainly in image study and to recognize the parts of the image where a huge variation in intensity arises.

4.1.2 MODULE 2: MORPHOLOGICAL OPERATIONS:

Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. The Morphological techniques are also used with segmentation techniques.

The morphological action is normally performed on binary images. It processes the operations based on shape and it has a wide set of the image processing operation. Erosion and Dilation are two methods of morphological operations which used in this proposed work. We perform both Erosion and dilation operations used together.

Two main steps of the erosion and dilation morphological operation are opening and closing. The first step is the opening of the MRI binary image. The main work of opening operation is open up a gap which is present in between object and connect that to a small collection of pixels. After setting of the bridge, the erosion again restored with their actual size using dilation. If the binary image has been opened then the subsequent opened same structured elements have not affected on that image. After completing the opening operations next step is the closing operation. Based on the closing operation while keeping the original region sizes, the erosion and dilation can handle different hole in the image region. Dilation and Erosion are the basic morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries.

4.1.3 MODULE 3 : IMAGE AGUMENTATION:

Medical image segmentation is often constrained by the availability of labelled training data. 'Data augmentation' helps to prevent memorisation of training data and helps the network's performance on data from outside the training set. As such, it is vital in building robust deep learning pipelines. Augmentation in medical imaging typically involves applying small transformations to images during training to create variety. However, it is also possible to use linear combinations of training images and labels to augment the dataset using the recently-proposed 'mixup' algorithm. Here, we apply this algorithm for use in medical imaging segmentation. We show that it increases performance in segmentation tasks, and also offer a theoretical suggestion for the efficacy of this technique

'Data augmentation' is used to artificially increase the size of the training dataset. In medical imaging, this is typically done with transformations that are applied to both the images and labels equally, creating warped versions of the training data. Augmentation methods commonly employ transformations such as rotations, reflections, and elastic deformations, which produce training images that closely resemble one particular training example. While the intuitive motivation behind augmentation strategies is appealing, a recently-proposed technique, 'mixup' works by training on linear combinations of existing training data: the training labels are also linear combinations of the ground-truth labels. Although images generated in this manner are noticeably different than training images (looking like two images super-posed), this augmentation technique has been shown to improve performance on a variety of machine-learning tasks.

4.1.4 MODULE 4: BRAIN TUMOR DETECTION USING CONVOLUTIONAL NEURAL NETWORK:

CNN is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification.

CNN uses **max pooling** to replace output with a max summary to reduce data size and processing time. This allows you to determine features that produce the highest impact and reduces the risk of overfitting.

Max pooling takes two **hyperparameters**: stride and size. The stride will determine the skip of value pools while the size will determine how big the value pools in every skip.

Activation Function (ReLU and Sigmoid)

After each convolutional and max pooling operation, we can apply Rectified Linear Unit (ReLU). The ReLU function mimics our neuron activations on a “big enough stimulus” to introduce nonlinearity for values x>0 and returns 0 if it does not meet the condition. This method has been effective to solve diminishing gradients. Weights that are very small will remain as 0 after the ReLU activation function.

Finally, we will serve the convolutional and max pooling feature map outputs with Fully Connected Layer (FCL). We flatten the feature outputs to column vector and feed-forward it to FCL. We wrap our features with **softmax** activation function which assign decimal probabilities for each possible label which add up to 1.0. Every node in the previous layer is connected to the last layer and represents which distinct label to output.

4.2 EXPERIMENTAL ANALYSIS AND RESULTS

4.2.1 SYSTEM CONFIGURATIONS

4.2.1.1 SOFTWARE REQUIREMENTS

- Windows: Python 3.6.2 or above, PIP and NumPy 1.13.1

PYTHON

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant Whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

PIP

It is the package management system used to install and manage software packages written in Python.

NUMPY

NumPy is a general-purpose array-processing package. It provide a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones

PANDAS

Provides highly optimized performance with back-end source code is purely written in C or Python. We can analyze data in pandas with

Pandas is the most popular python library that is used for data analysis.

1. Series

2. Data frames

TENSORFLOW

Tensor flow is a free and open-source software library for dataflow an Differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

KERAS

Running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or Plaid ML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. Keras is an open-source neural-network library written in Python.

OPENCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by willow garage then Itseez (which was later acquired by Intel). The library is cross platform and free for use under the open source BSD license. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers. It promotes Open Vision Capsules. which is a portable format, compatible with all other formats.

SEABORN

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

GLOB

The glob module is a useful part of the Python Standard Library Glob (short for global) is used to return all file paths that match a specific pattern. We can use Glob to search for a specific file pattern, or perhaps more usefully, search for files where the filename matches a certain pattern by using wildcard characters.

SK LEARN

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

4.2.1.2 HARDWARE CONFIGURATIONS:

1) Processor: AMD Ryzen 5 5500u or above
2) 64 bit, 2.5 GHz minimum per core
3) RAM 8GB
4) Hard disk: 10 GB of available space or more.
5) Operating System: Windows

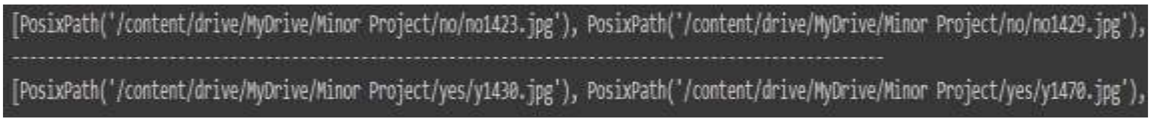
4.2.2 SMAPLE CODE AND RESULTS:

Importing libraries

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from warnings import filterwarnings

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, roc_auc_score, roc_curve

from tensorflow.keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization,MaxPooling2D

from keras import models

from keras import layers

import tensorflow as tf

import os

import os.path

from pathlib import Path

import cv2

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from keras.utils.np_utils import to_categorical

from sklearn.model_selection import train_test_split

from keras import regularizers

from keras.optimizers import RMSprop,Adam

import glob

from PIL import Image

# IGNORING UNNECESSARRY WARNINGS

filterwarnings("ignore",category=DeprecationWarning)
filterwarnings("ignore", category=FutureWarning)

filterwarnings("ignore", category=UserWarning)
```

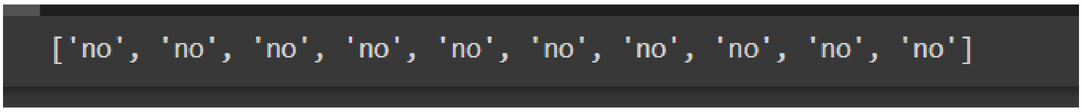
PATH PROCESS

```
No_Data_Path = Path("/content/drive/MyDrive/Minor Project/no")

Yes_Data_Path = Path("/content/drive/MyDrive/Minor Project/yes")

No_JPG_Path = list(No_Data_Path.glob(r"*.jpg"))

Yes_JPG_Path = list(Yes_Data_Path.glob(r"*.jpg"))

print(No_JPG_Path[0:5])

print("-----"*20)

print(Yes_JPG_Path[0:5])
```



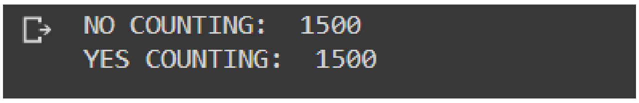
```
[PosixPath('/content/drive/MyDrive/Minor Project/no/no1423.jpg'), PosixPath('/content/drive/MyDrive/Minor Project/no/no1429.jpg'),
----------------------------------------------------------------------------------------------------
[PosixPath('/content/drive/MyDrive/Minor Project/yes/y1430.jpg'), PosixPath('/content/drive/MyDrive/Minor Project/yes/y1470.jpg'),
```

```
JPG_Labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1],Yes_No_List))

print(JPG_Labels[0:10])
```



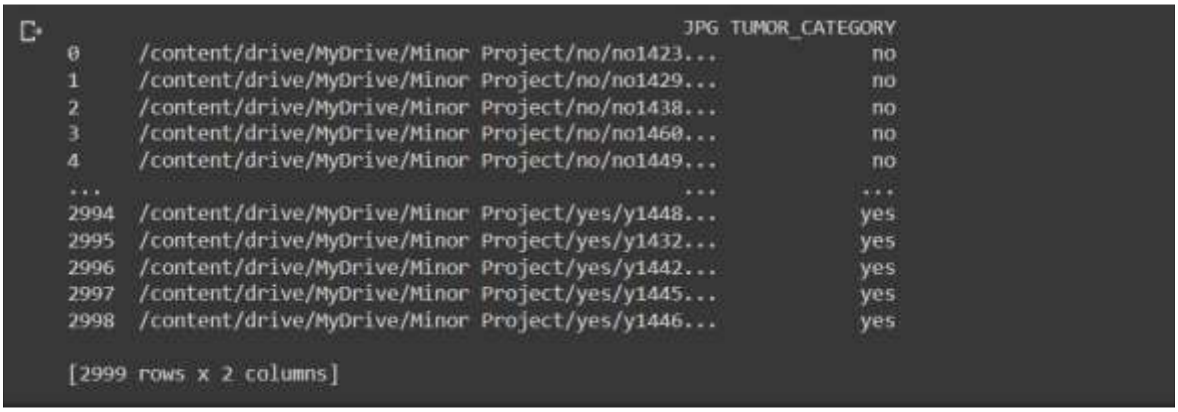
```
['no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no']
```

```
print("NO COUNTING: ", JPG_Labels.count("no"))
print("YES COUNTING: ", JPG_Labels.count("yes"))
```



```
NO COUNTING:  1500
YES COUNTING:  1500
```

```
print(Main_Train_Data.head(-1))
```



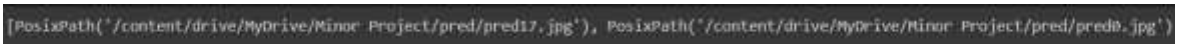
```
                                                JPG TUMOR_CATEGORY
0     /content/drive/MyDrive/Minor Project/no/no1423...            no
1     /content/drive/MyDrive/Minor Project/no/no1429...            no
2     /content/drive/MyDrive/Minor Project/no/no1438...            no
3     /content/drive/MyDrive/Minor Project/no/no1460...            no
4     /content/drive/MyDrive/Minor Project/no/no1449...            no
...                                                 ...           ...
2994  /content/drive/MyDrive/Minor Project/yes/y1448...           yes
2995  /content/drive/MyDrive/Minor Project/yes/y1432...           yes
2996  /content/drive/MyDrive/Minor Project/yes/y1442...           yes
2997  /content/drive/MyDrive/Minor Project/yes/y1445...           yes
2998  /content/drive/MyDrive/Minor Project/yes/y1446...           yes

[2999 rows x 2 columns]
```

TEST

Prediction_Path = Path("/content/drive/MyDrive/Minor Project/pred")

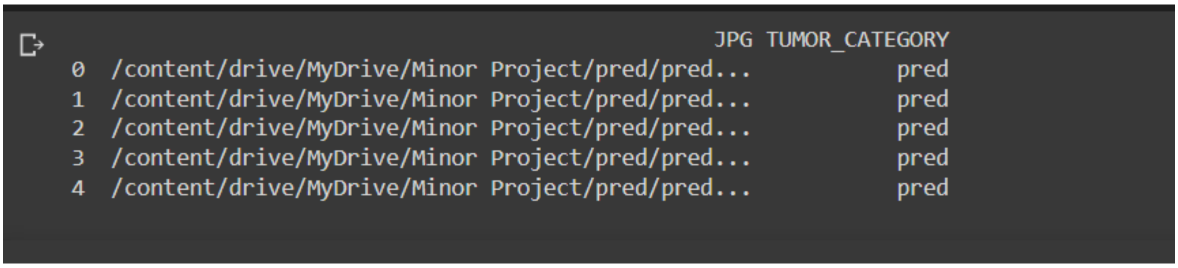
Test_JPG_Path = list(Prediction_Path.glob(r"*.jpg"))

print(Test_JPG_Path[0:2])



```
[PosixPath('/content/drive/MyDrive/Minor Project/pred/pred17.jpg'), PosixPath('/content/drive/MyDrive/Minor Project/pred/pred0.jpg')
```

print(Test_Data.head())

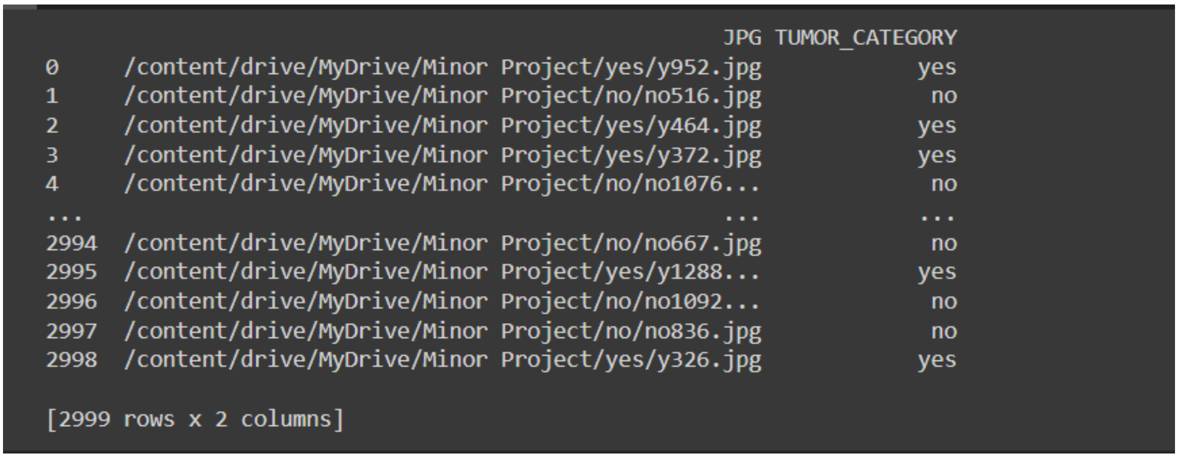


```
                                           JPG TUMOR_CATEGORY
0  /content/drive/MyDrive/Minor Project/pred/pred...           pred
1  /content/drive/MyDrive/Minor Project/pred/pred...           pred
2  /content/drive/MyDrive/Minor Project/pred/pred...           pred
3  /content/drive/MyDrive/Minor Project/pred/pred...           pred
4  /content/drive/MyDrive/Minor Project/pred/pred...           pred
```

SHUFFLING

Main_Train_Data = Main_Train_Data.sample(frac=1).reset_index(drop=True)

print(Main_Train_Data.head(-1))



```
                                                JPG TUMOR_CATEGORY
0     /content/drive/MyDrive/Minor Project/yes/y952.jpg            yes
1     /content/drive/MyDrive/Minor Project/no/no516.jpg             no
2     /content/drive/MyDrive/Minor Project/yes/y464.jpg            yes
3     /content/drive/MyDrive/Minor Project/yes/y372.jpg            yes
4     /content/drive/MyDrive/Minor Project/no/no1076...             no
...                                                 ...            ...
2994  /content/drive/MyDrive/Minor Project/no/no667.jpg             no
2995  /content/drive/MyDrive/Minor Project/yes/y1288...            yes
2996  /content/drive/MyDrive/Minor Project/no/no1092...             no
2997  /content/drive/MyDrive/Minor Project/no/no836.jpg             no
2998  /content/drive/MyDrive/Minor Project/yes/y326.jpg            yes

[2999 rows x 2 columns]
```

VISUALIZATION

figure = plt.figure(figsize=(10,10))

plt.imshow(plt.imread(Main_Train_Data["JPG"][10]))

plt.title(Main_Train_Data["TUMOR_CATEGORY"][10])

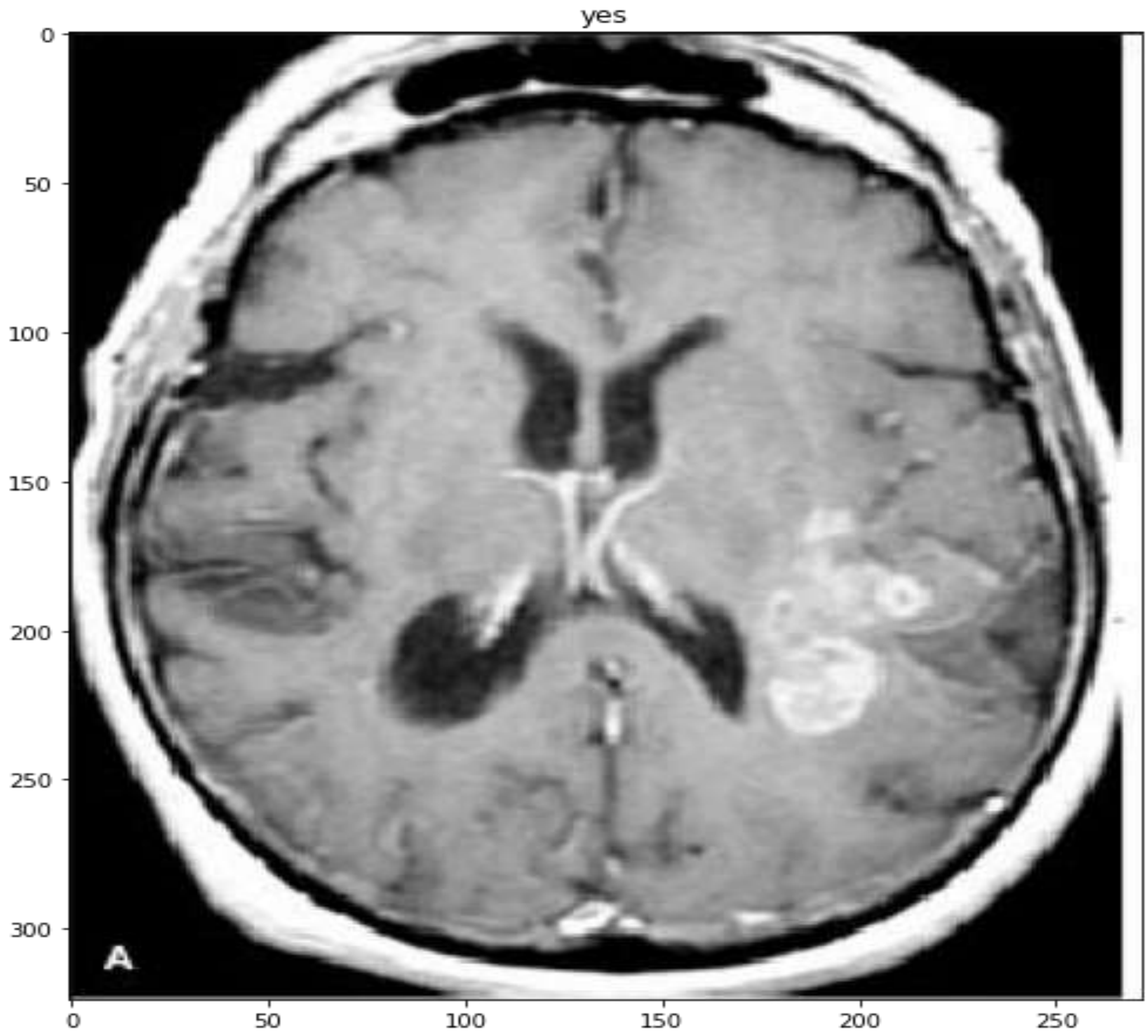


Figure 2 TUMOR VIZUALIZATION(YES)

```
figure = plt.figure(figsize=(10,10))
plt.imshow(plt.imread(Main_Train_Data["JPG"][2997]))
plt.title(Main_Train_Data["TUMOR_CATEGORY"][2997])
```

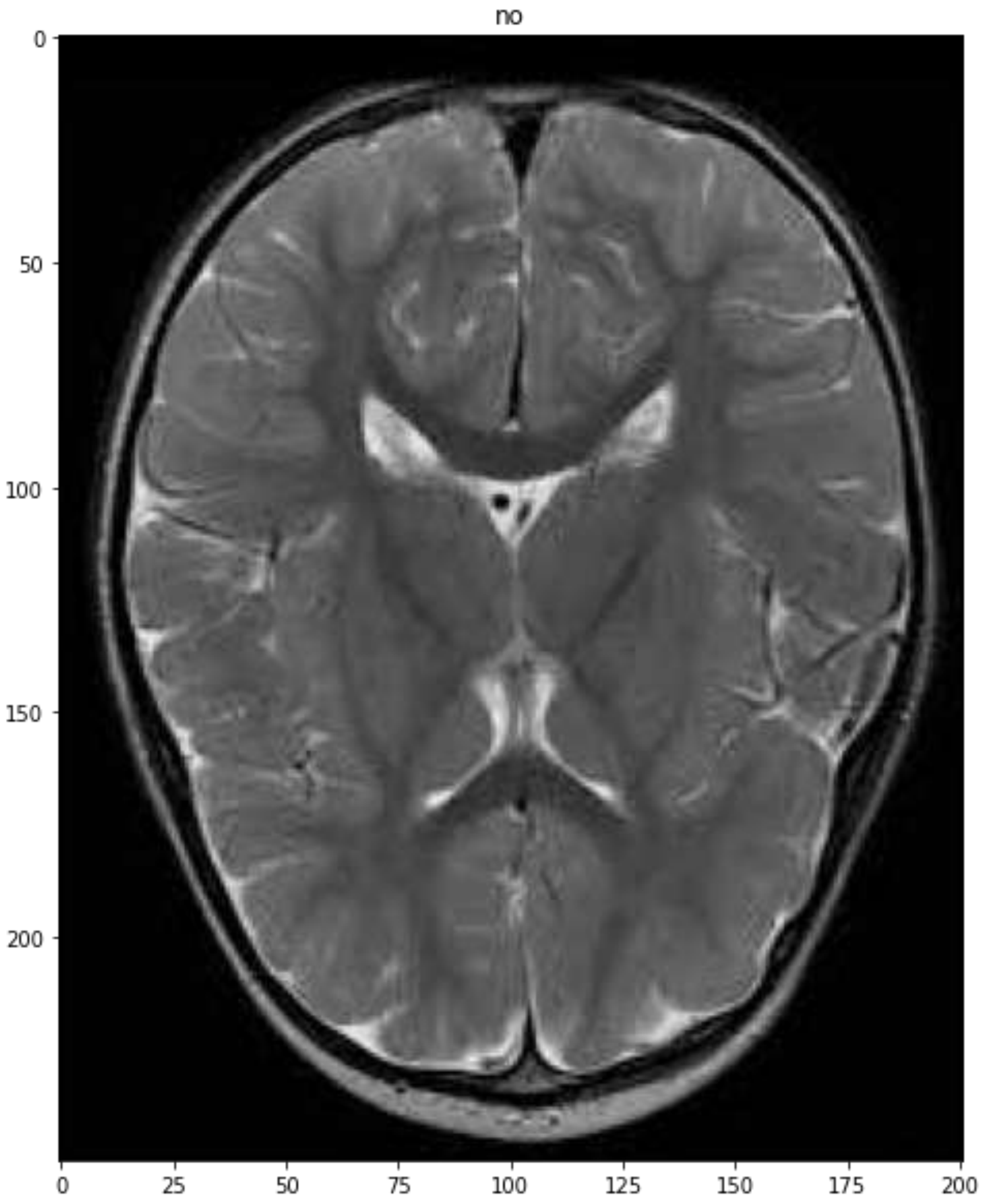


Figure 3 TUMOR VISUALIZATION(NO)

```
fig, axes = plt.subplots(nrows=5, ncols=5, figsize=(10, 10),
                        subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(Main_Train_Data["JPG"][i]))
    ax.set_title(Main_Train_Data["TUMOR_CATEGORY"][i])
plt.tight_layout()
plt.show()
```

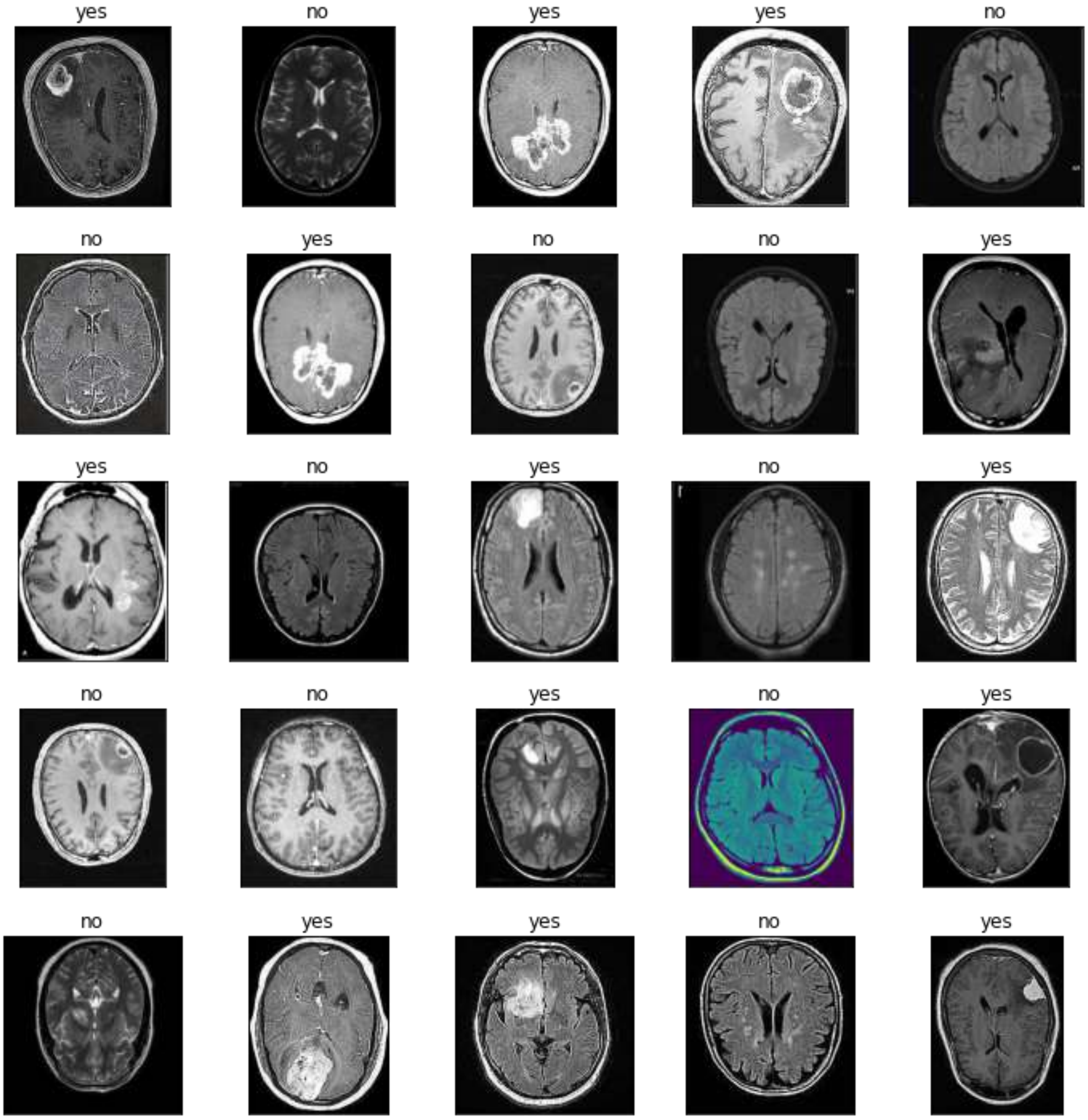


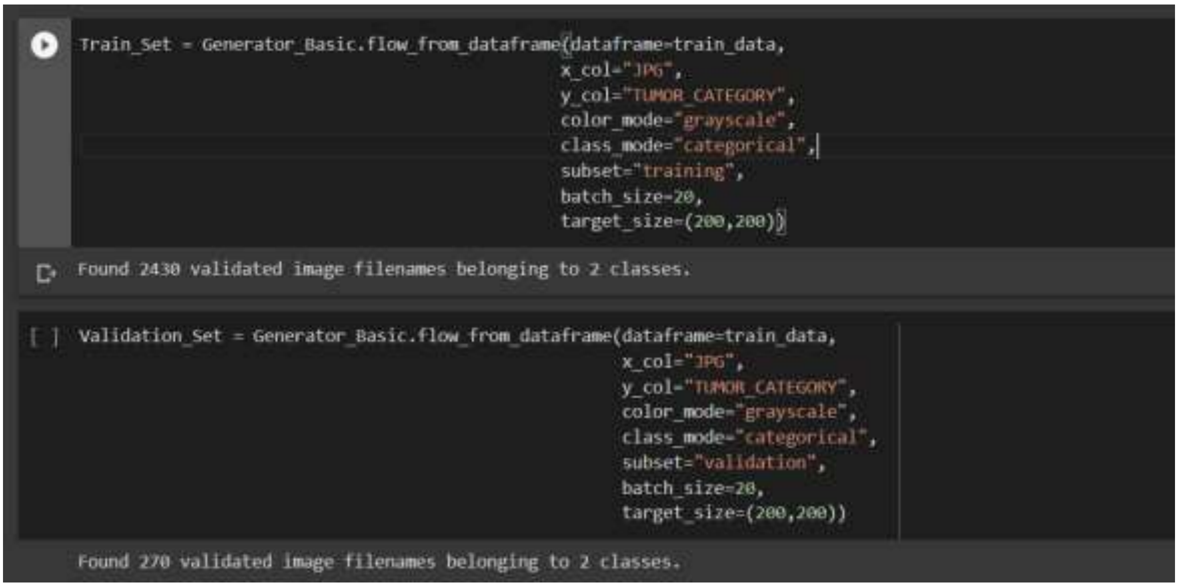
Figure 4 TUMOR IMAGES

MODEL TRAINING DATA

```
train_data,test_data = train_test_split(Main_Train_Data,train_size=0.9,random_state=42)
```

IMAGE DATA GENERATOR WITHOUT AGUMENTATION

```
Generator_Basic = ImageDataGenerator(rescale=1./255,
                                     validation_split=0.1)
```

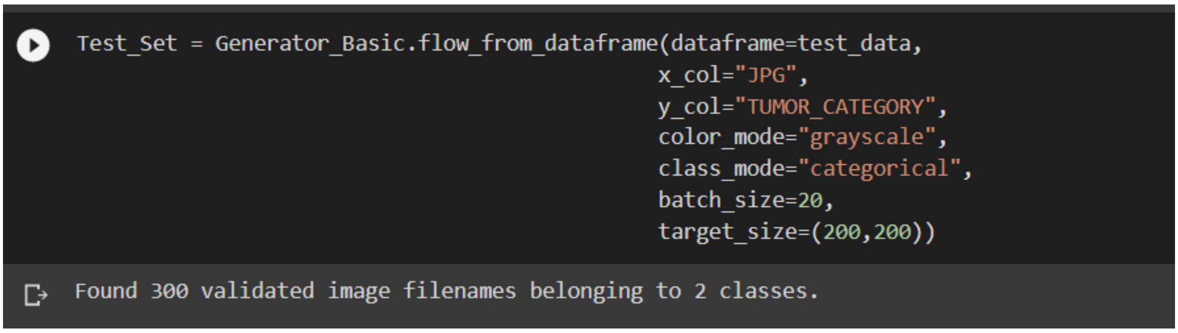


```
Train_Set = Generator_Basic.flow_from_dataframe(dataframe=train_data,
                                                x_col="JPG",
                                                y_col="TUMOR_CATEGORY",
                                                color_mode="grayscale",
                                                class_mode="categorical",
                                                subset="training",
                                                batch_size=20,
                                                target_size=(200,200))
```

```
Found 2430 validated image filenames belonging to 2 classes.
```

```
Validation_Set = Generator_Basic.flow_from_dataframe(dataframe=train_data,
                                                     x_col="JPG",
                                                     y_col="TUMOR_CATEGORY",
                                                     color_mode="grayscale",
                                                     class_mode="categorical",
                                                     subset="validation",
                                                     batch_size=20,
                                                     target_size=(200,200))
```

```
Found 270 validated image filenames belonging to 2 classes.
```



```
Test_Set = Generator_Basic.flow_from_dataframe(dataframe=test_data,
                                               x_col="JPG",
                                               y_col="TUMOR_CATEGORY",
                                               color_mode="grayscale",
                                               class_mode="categorical",
                                               batch_size=20,
                                               target_size=(200,200))
```

```
Found 300 validated image filenames belonging to 2 classes.
```

CNN MODEL

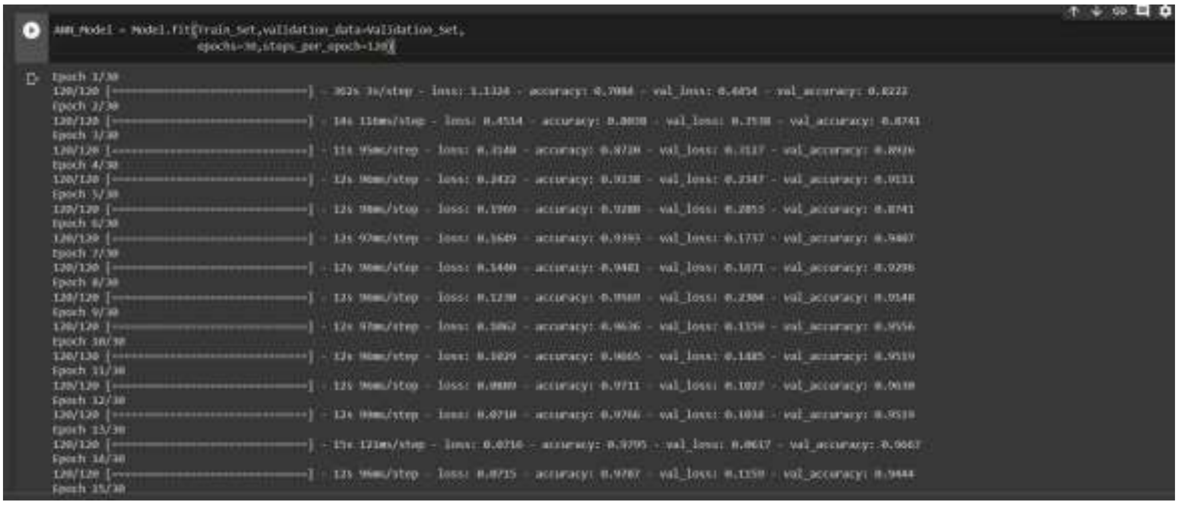
```
Model = Sequential()


Model.add(Conv2D(32,(5,5),activation="relu",input_shape=(200,200,1)))
Model.add(MaxPool2D((2,2)))
Model.add(Dropout(0.2))
#
Model.add(Conv2D(64,(3,3),activation="relu"))
Model.add(MaxPool2D((2,2)))
Model.add(Dropout(0.2))
#
Model.add(Conv2D(128,(3,3),activation="relu"))
Model.add(MaxPool2D((2,2)))
Model.add(Dropout(0.2))
#
Model.add(Conv2D(256,(3,3),activation="relu"))
Model.add(MaxPool2D((2,2)))
Model.add(Dropout(0.2))
#
Model.add(Flatten())
Model.add(Dropout(0.5))
Model.add(Dense(512,activation="relu"))
Model.add(Dense(2,activation="softmax"))
```

MODEL COMPILE

```
Model.compile(optimizer=RMSprop(lr=0.001),loss="categorical_crossentropy",metrics=["
accuracy"])
```

MODEL FITTING

```
ANN_Model = Model.fit(Train_Set,validation_data=Validation_Set,
                    epochs=30,steps_per_epoch=120)
```

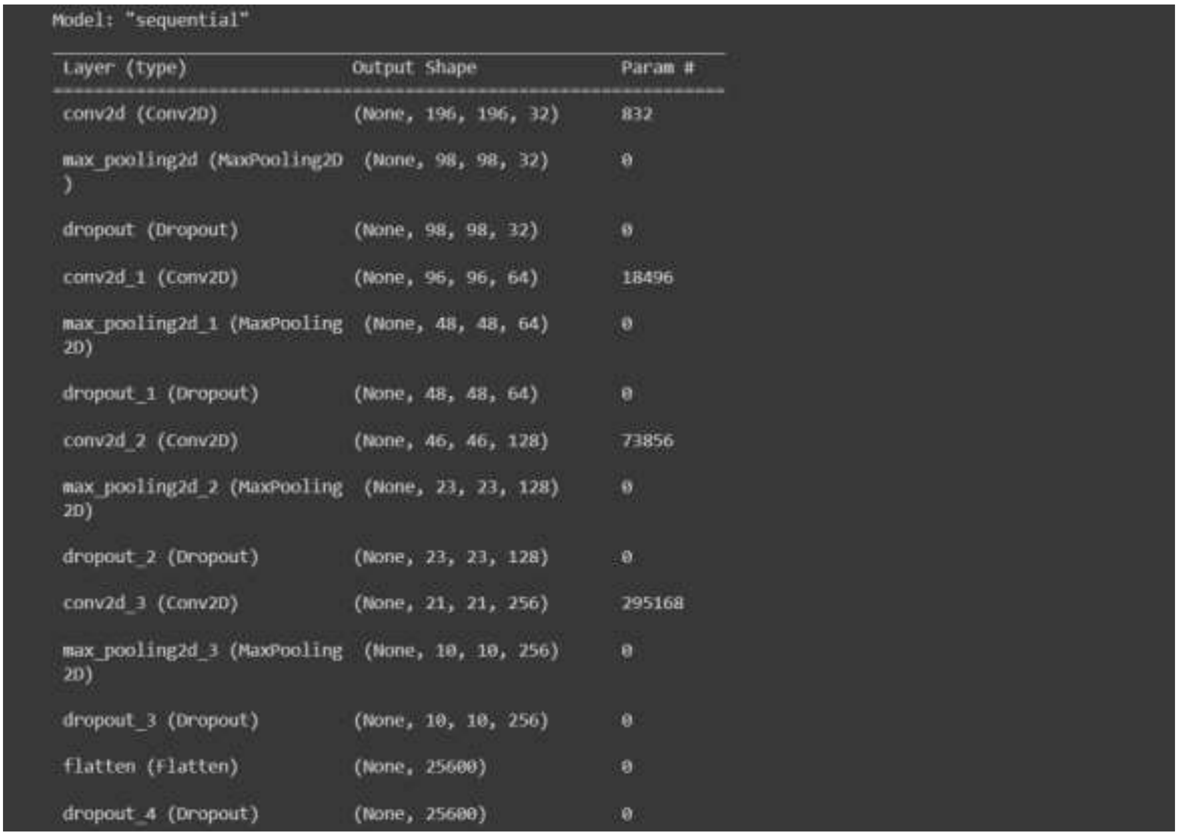


```
ANN_Model = Model.fit(Train_Set,validation_data=Validation_Set,
                      epochs=30,steps_per_epoch=120)
```

```
Epoch 1/30
120/120 [==============================] - 362s 3s/step - loss: 1.1324 - accuracy: 0.7084 - val_loss: 0.4854 - val_accuracy: 0.8222
Epoch 2/30
120/120 [==============================] - 14s 116ms/step - loss: 0.4514 - accuracy: 0.8838 - val_loss: 0.3538 - val_accuracy: 0.8741
Epoch 3/30
120/120 [==============================] - 11s 95ms/step - loss: 0.3140 - accuracy: 0.8720 - val_loss: 0.3117 - val_accuracy: 0.8926
Epoch 4/30
120/120 [==============================] - 12s 98ms/step - loss: 0.2422 - accuracy: 0.9138 - val_loss: 0.2347 - val_accuracy: 0.9111
Epoch 5/30
120/120 [==============================] - 12s 98ms/step - loss: 0.1969 - accuracy: 0.9288 - val_loss: 0.2855 - val_accuracy: 0.8741
Epoch 6/30
120/120 [==============================] - 12s 97ms/step - loss: 0.1649 - accuracy: 0.9393 - val_loss: 0.1737 - val_accuracy: 0.9407
Epoch 7/30
120/120 [==============================] - 12s 98ms/step - loss: 0.1440 - accuracy: 0.9481 - val_loss: 0.1871 - val_accuracy: 0.9296
Epoch 8/30
120/120 [==============================] - 12s 98ms/step - loss: 0.1238 - accuracy: 0.9560 - val_loss: 0.2304 - val_accuracy: 0.9148
Epoch 9/30
120/120 [==============================] - 12s 97ms/step - loss: 0.1062 - accuracy: 0.9636 - val_loss: 0.1359 - val_accuracy: 0.9556
Epoch 10/30
120/120 [==============================] - 12s 96ms/step - loss: 0.1029 - accuracy: 0.9665 - val_loss: 0.1485 - val_accuracy: 0.9519
Epoch 11/30
120/120 [==============================] - 12s 96ms/step - loss: 0.0889 - accuracy: 0.9711 - val_loss: 0.1027 - val_accuracy: 0.9630
Epoch 12/30
120/120 [==============================] - 12s 99ms/step - loss: 0.0710 - accuracy: 0.9766 - val_loss: 0.1034 - val_accuracy: 0.9519
Epoch 13/30
120/120 [==============================] - 15s 121ms/step - loss: 0.0716 - accuracy: 0.9795 - val_loss: 0.0617 - val_accuracy: 0.9667
Epoch 14/30
120/120 [==============================] - 12s 96ms/step - loss: 0.0715 - accuracy: 0.9787 - val_loss: 0.1150 - val_accuracy: 0.9444
Epoch 15/30
```

MODEL SUMMARY

```
print(Model.summary())
```



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 196, 196, 32)      832

 max_pooling2d (MaxPooling2D  (None, 98, 98, 32)       0
 )

 dropout (Dropout)           (None, 98, 98, 32)        0

 conv2d_1 (Conv2D)           (None, 96, 96, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 48, 48, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 48, 48, 64)        0

 conv2d_2 (Conv2D)           (None, 46, 46, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 23, 23, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 23, 23, 128)       0

 conv2d_3 (Conv2D)           (None, 21, 21, 256)       295168

 max_pooling2d_3 (MaxPooling  (None, 10, 10, 256)      0
 2D)

 dropout_3 (Dropout)         (None, 10, 10, 256)       0

 flatten (Flatten)           (None, 25600)             0

 dropout_4 (Dropout)         (None, 25600)             0
```

CHECKING MODEL WITH GRAPHIC

```
HistoryDict = ANN_Model.history
val_losses = HistoryDict["val_loss"]
val_acc = HistoryDict["val_accuracy"]
acc = HistoryDict["accuracy"]
losses = HistoryDict["loss"]
epochs = range(1,len(val_losses)+1)
plt.plot(ANN_Model.history["accuracy"],label ="ACCURACY")
plt.plot(ANN_Model.history["val_accuracy"],label = "ACCURACY VALIDATION")
plt.ylabel("ACCURACY")
plt.xlabel("EPOCH")
plt.legend()
plt.show()
```

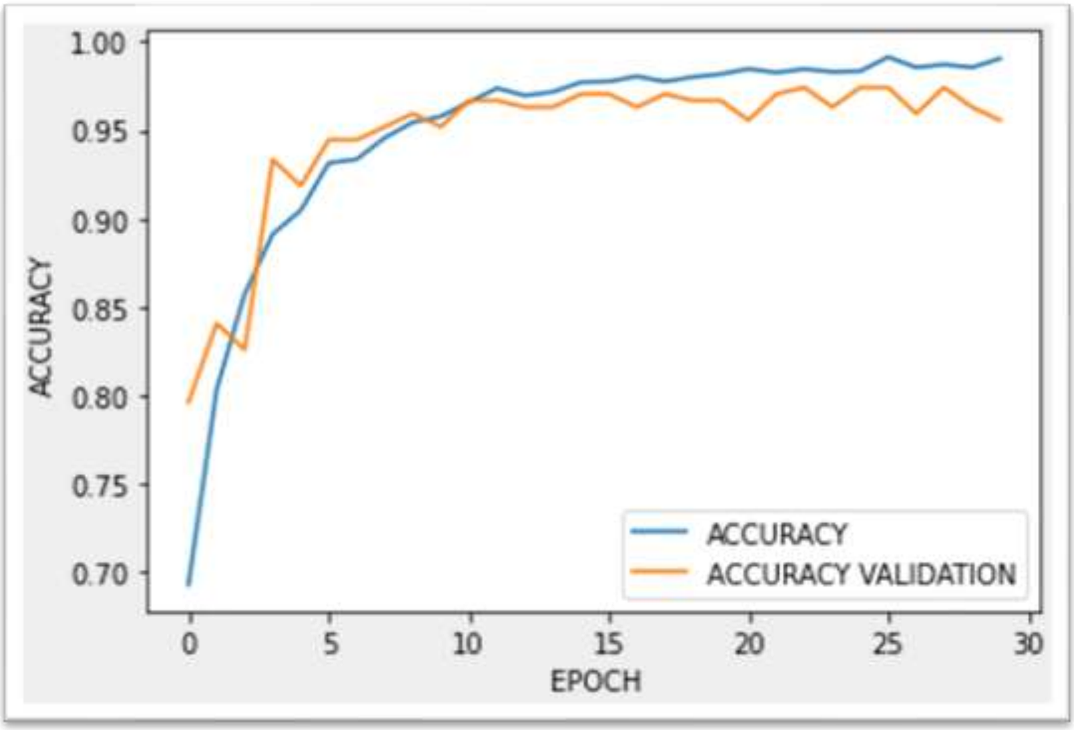


Figure 5 ACCURACY VS EPOCHS

```
plt.plot(epochs,losses,"k-",label="LOSS")
plt.plot(epochs,val_losses,"ro",label="LOSS VALIDATION")
plt.title("LOSS & LOSS VAL")
plt.xlabel("EPOCH")
plt.ylabel("LOSS & LOSS VAL")
plt.legend()
plt.show()
```

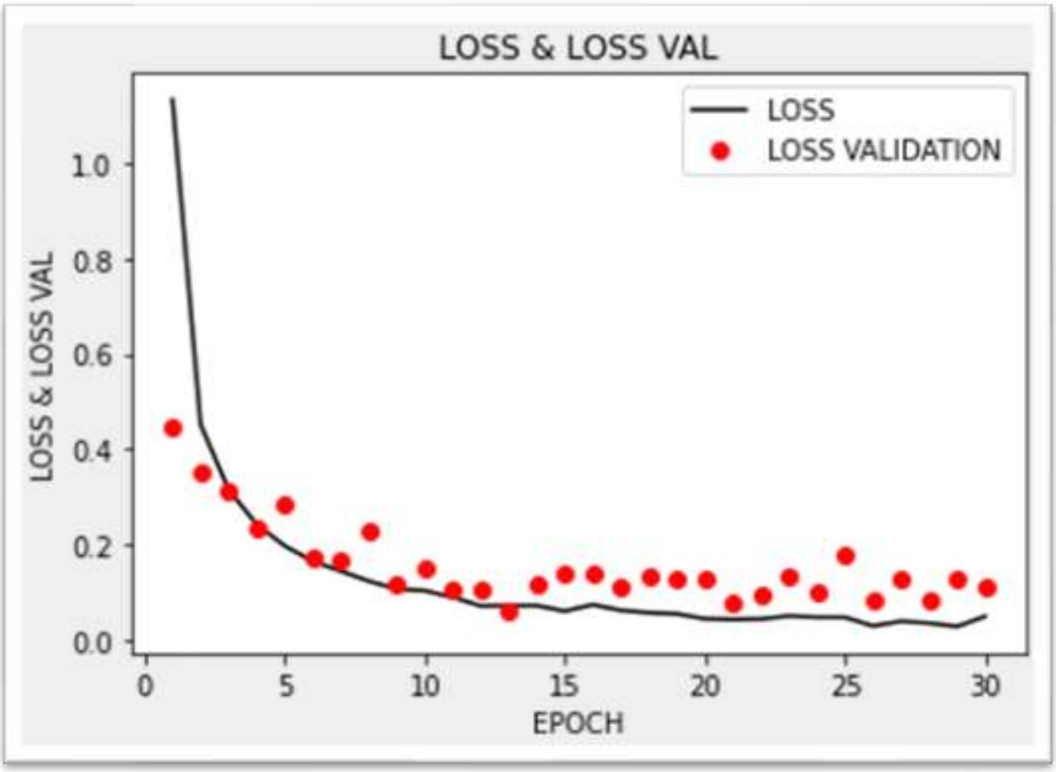


Figure 6 LOSS & LOSS VAL VS EPOCH

```
plt.plot(epochs,acc,"k-",label="ACCURACY")
plt.plot(epochs,val_acc,"ro",label="ACCURACY VALIDATION")
plt.title("ACCURACY & ACCURACY VAL")
plt.xlabel("EPOCH")
plt.ylabel("ACCURACY & ACCURACY VAL")
plt.legend()
plt.show()
```

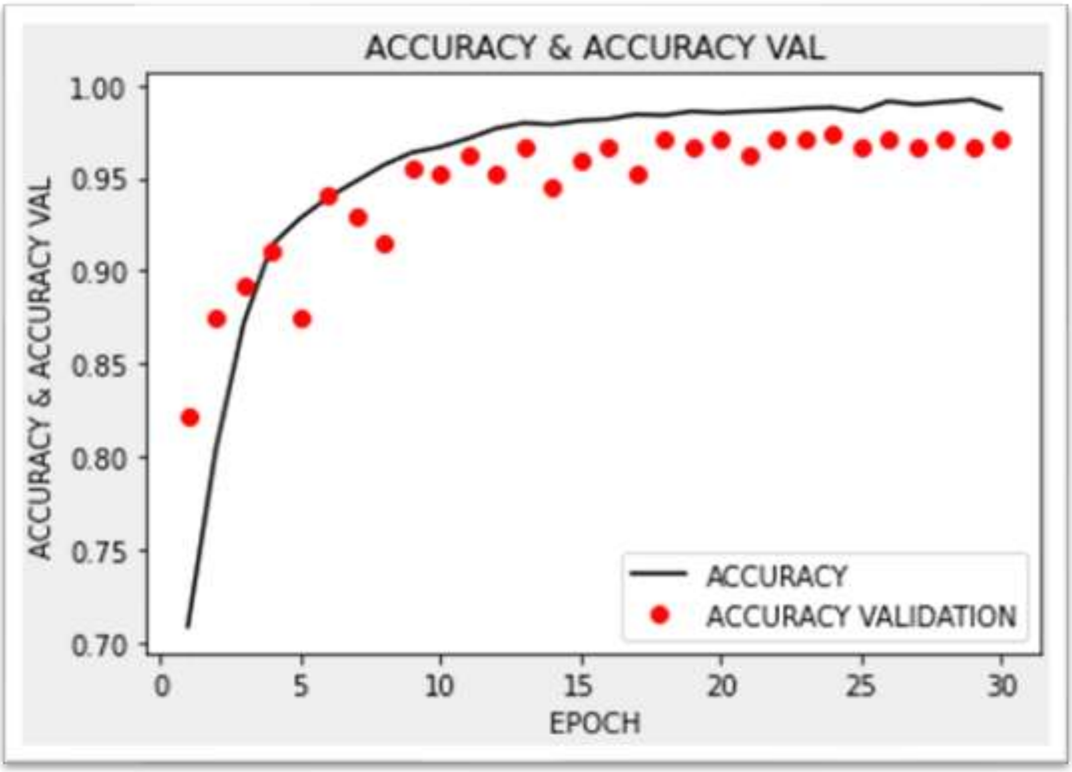


Figure 7 ACCURACY & ACCURACY VAL VS EPOCH

```
Dict_Summary = pd.DataFrame(ANN_Model.history)
Dict_Summary.plot()
plt.xlabel("EPOCH")
plt.ylabel("ACC,ACC VAL,LOSS & LOSS VAL")
```

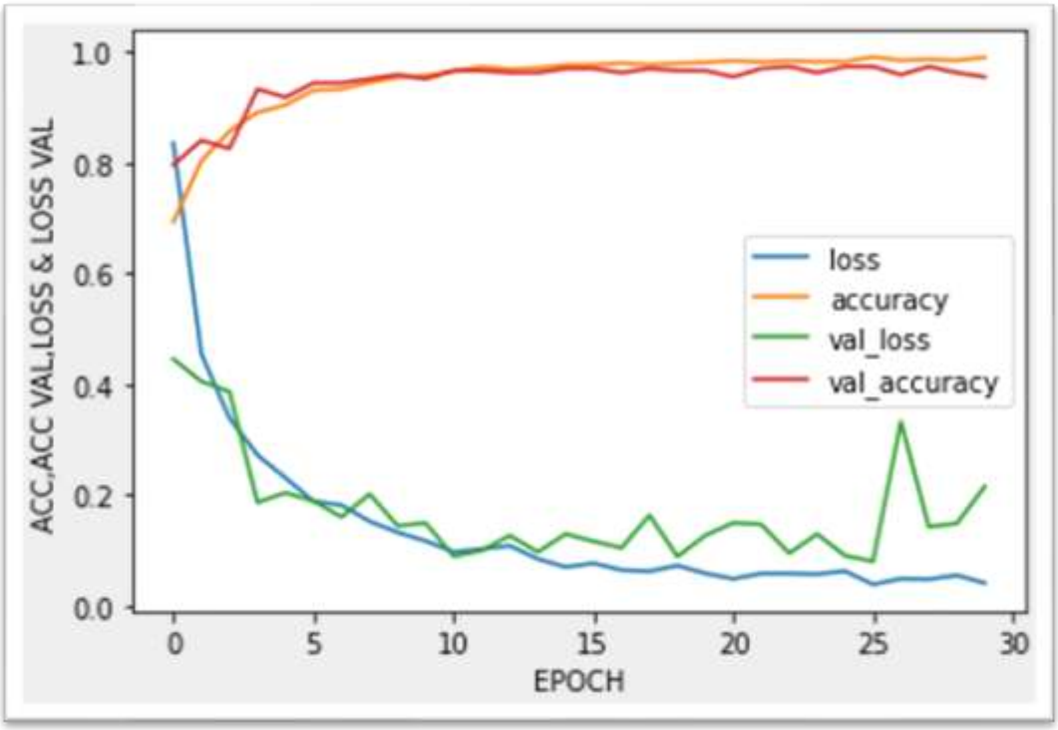
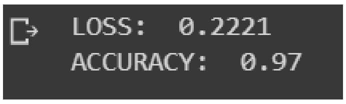


Figure 8 ACC,ACC VAL,LOSS & LOSS VAL VS EPOCS

```
Model_Results = Model.evaluate(Test_Set,verbose=False)
print("LOSS:  " + "%.4f" % Model_Results[0])
print("ACCURACY:  " + "%.2f" % Model_Results[1])
```



```
LOSS:  0.2221
ACCURACY:  0.97
```

PREDICTION

```
fig, axes = plt.subplots(nrows=5,
                         ncols=5,
                         figsize=(20, 20),
                        subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(Main_Data_Prediction["JPG"].loc[i]))
    ax.set_title(f"PREDICTION:{Last_Prediction[i]}")
plt.tight_layout()
plt.show()
```

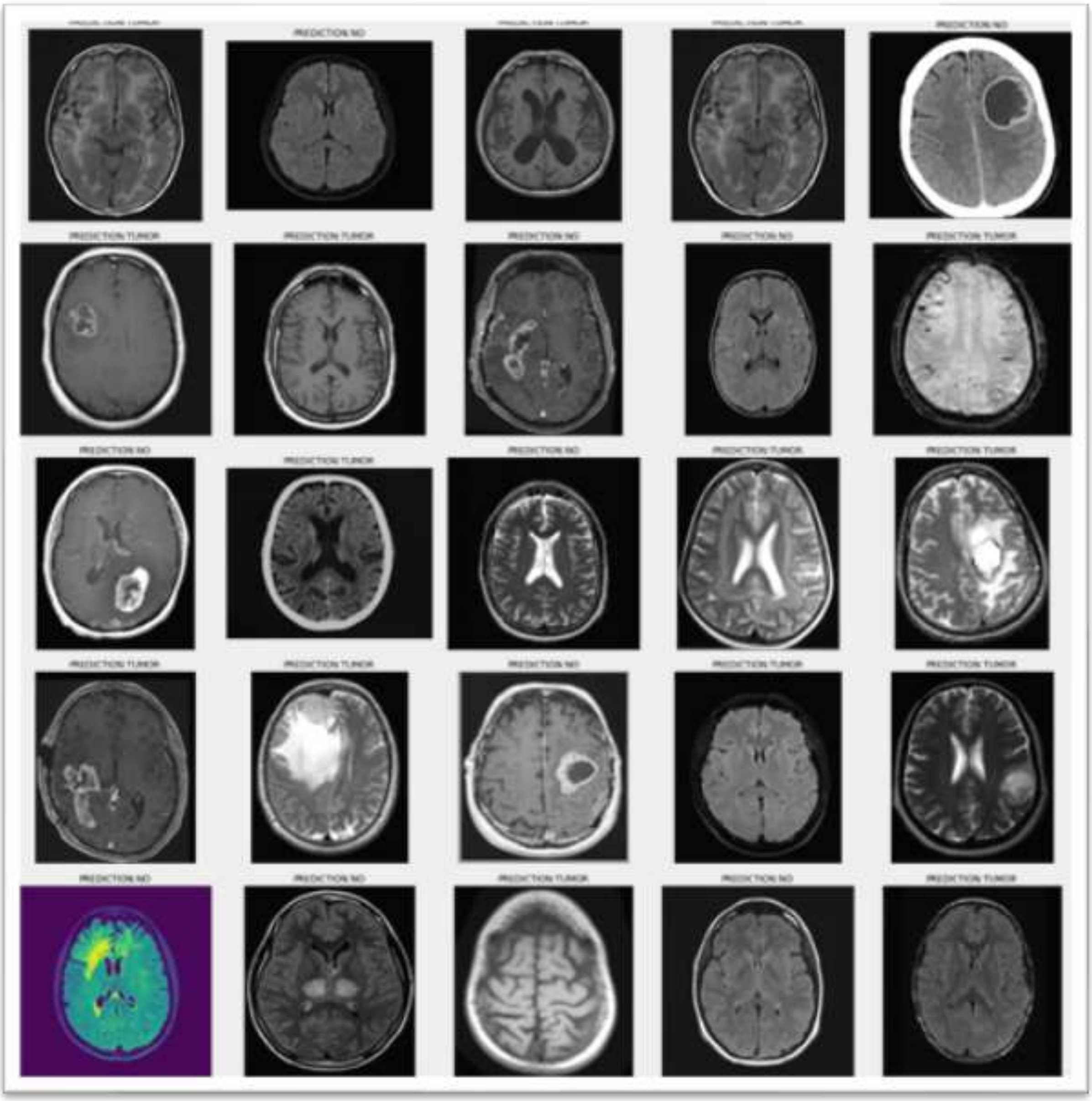
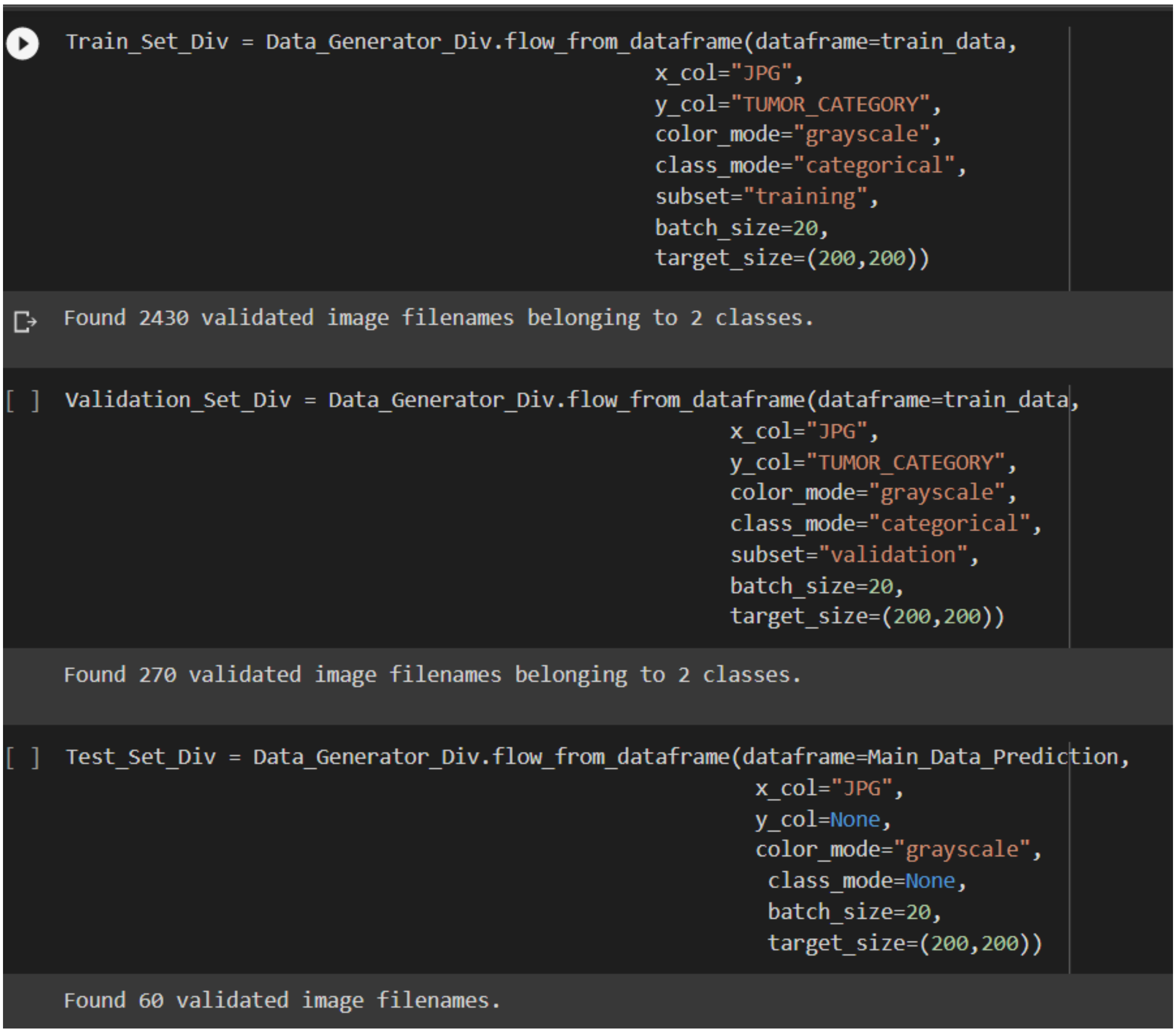


Figure 9 PREDICTION WITHOUT AGUMENTATION

IMAGE DATA GENERATOR WITH AGUMENTATION

```
Data_Generator_Div = ImageDataGenerator(rescale=1./255,brightness_range=[0.3,0.9],
                        rotation_range=30,
                        width_shift_range=0.2,
                        height_shift_range=0.2,
                        shear_range=0.2,
                        horizontal_flip=True,
                        fill_mode="nearest",
                        validation_split=0.1)
```



```
Train_Set_Div = Data_Generator_Div.flow_from_dataframe(dataframe=train_data,
                                                       x_col="JPG",
                                                       y_col="TUMOR_CATEGORY",
                                                       color_mode="grayscale",
                                                       class_mode="categorical",
                                                       subset="training",
                                                       batch_size=20,
                                                       target_size=(200,200))
```

```
Found 2430 validated image filenames belonging to 2 classes.
```

```
[ ] Validation_Set_Div = Data_Generator_Div.flow_from_dataframe(dataframe=train_data,
                                                              x_col="JPG",
                                                              y_col="TUMOR_CATEGORY",
                                                              color_mode="grayscale",
                                                              class_mode="categorical",
                                                              subset="validation",
                                                              batch_size=20,
                                                              target_size=(200,200))
```

```
Found 270 validated image filenames belonging to 2 classes.
```

```
[ ] Test_Set_Div = Data_Generator_Div.flow_from_dataframe(dataframe=Main_Data_Prediction,
                                                          x_col="JPG",
                                                          y_col=None,
                                                          color_mode="grayscale",
                                                           class_mode=None,
                                                           batch_size=20,
                                                           target_size=(200,200))
```

```
Found 60 validated image filenames.
```

CNN MODEL WITH AGUMENTATION

Model_Two = Sequential()

Model_Two.add(Conv2D(32,(5,5),activation="relu",input_shape=(200,200,1)))

Model_Two.add(MaxPool2D((2,2)))

Model_Two.add(Conv2D(64,(3,3),activation="relu"))

Model_Two.add(MaxPool2D((2,2)))

Model_Two.add(Conv2D(128,(3,3),activation="relu"))

Model_Two.add(MaxPool2D((2,2)))

Model_Two.add(Conv2D(128,(3,3),activation="relu"))

Model_Two.add(MaxPool2D((2,2)))

Model_Two.add(Conv2D(256,(3,3),activation="relu"))

Model_Two.add(MaxPool2D((2,2)))

Model_Two.add(Flatten())

Model_Two.add(Dropout(0.5))

Model_Two.add(Dense(512,activation="relu"))

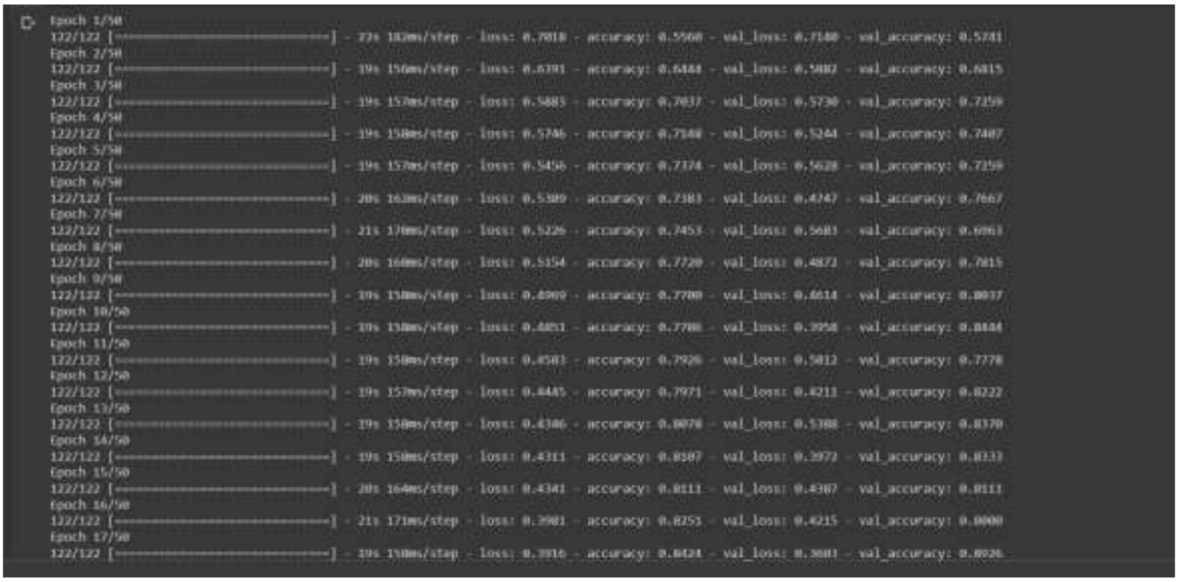
Model_Two.add(Dense(2,activation="softmax"))

MODEL COMPILE

Model_Two.compile(optimizer=RMSprop(lr=0.001),loss="categorical_crossentropy",metrics=["accuracy"])

MODEL FITTING

```
ANN_Model_Two = Model_Two.fit(Train_Set_Div,
                    validation_data=Validation_Set_Div,
                    batch_size=20,
               epochs=50)
```



```
Epoch 1/50
122/122 [==============================] - 23s 182ms/step - loss: 0.7018 - accuracy: 0.5568 - val_loss: 0.7140 - val_accuracy: 0.5741
Epoch 2/50
122/122 [==============================] - 19s 156ms/step - loss: 0.6391 - accuracy: 0.6444 - val_loss: 0.5882 - val_accuracy: 0.6815
Epoch 3/50
122/122 [==============================] - 19s 157ms/step - loss: 0.5883 - accuracy: 0.7037 - val_loss: 0.5730 - val_accuracy: 0.7259
Epoch 4/50
122/122 [==============================] - 19s 158ms/step - loss: 0.5746 - accuracy: 0.7148 - val_loss: 0.5244 - val_accuracy: 0.7407
Epoch 5/50
122/122 [==============================] - 19s 157ms/step - loss: 0.5456 - accuracy: 0.7374 - val_loss: 0.5628 - val_accuracy: 0.7259
Epoch 6/50
122/122 [==============================] - 20s 162ms/step - loss: 0.5309 - accuracy: 0.7383 - val_loss: 0.4747 - val_accuracy: 0.7667
Epoch 7/50
122/122 [==============================] - 21s 170ms/step - loss: 0.5226 - accuracy: 0.7453 - val_loss: 0.5681 - val_accuracy: 0.6963
Epoch 8/50
122/122 [==============================] - 20s 160ms/step - loss: 0.5154 - accuracy: 0.7720 - val_loss: 0.4872 - val_accuracy: 0.7815
Epoch 9/50
122/122 [==============================] - 19s 158ms/step - loss: 0.4969 - accuracy: 0.7700 - val_loss: 0.4614 - val_accuracy: 0.8037
Epoch 10/50
122/122 [==============================] - 19s 158ms/step - loss: 0.4851 - accuracy: 0.7706 - val_loss: 0.3958 - val_accuracy: 0.8444
Epoch 11/50
122/122 [==============================] - 19s 158ms/step - loss: 0.4583 - accuracy: 0.7926 - val_loss: 0.5012 - val_accuracy: 0.7778
Epoch 12/50
122/122 [==============================] - 19s 157ms/step - loss: 0.4445 - accuracy: 0.7971 - val_loss: 0.4211 - val_accuracy: 0.8222
Epoch 13/50
122/122 [==============================] - 19s 158ms/step - loss: 0.4306 - accuracy: 0.8078 - val_loss: 0.5388 - val_accuracy: 0.8370
Epoch 14/50
122/122 [==============================] - 19s 158ms/step - loss: 0.4311 - accuracy: 0.8107 - val_loss: 0.3972 - val_accuracy: 0.8333
Epoch 15/50
122/122 [==============================] - 20s 164ms/step - loss: 0.4341 - accuracy: 0.8111 - val_loss: 0.4307 - val_accuracy: 0.8111
Epoch 16/50
122/122 [==============================] - 21s 171ms/step - loss: 0.3981 - accuracy: 0.8251 - val_loss: 0.4215 - val_accuracy: 0.8000
Epoch 17/50
122/122 [==============================] - 19s 158ms/step - loss: 0.3916 - accuracy: 0.8424 - val_loss: 0.3681 - val_accuracy: 0.8926
```

CHECKING MODEL WITH GRAPHIC

```
HistoryDict_Two = ANN_Model_Two.history

val_losses_Two = HistoryDict_Two["val_loss"]
val_acc_Two = HistoryDict_Two["val_accuracy"]
acc_Two = HistoryDict_Two["accuracy"]
losses_Two = HistoryDict_Two["loss"]
epochs_Two = range(1,len(val_losses_Two)+1)
plt.plot(ANN_Model_Two.history["accuracy"],label="ACCURACY")
plt.plot(ANN_Model_Two.history["val_accuracy"],label="ACCURACY VALIDATION")
plt.ylabel("ACCURACY")
plt.xlabel("EPOCH")
plt.legend()
plt.show()
```

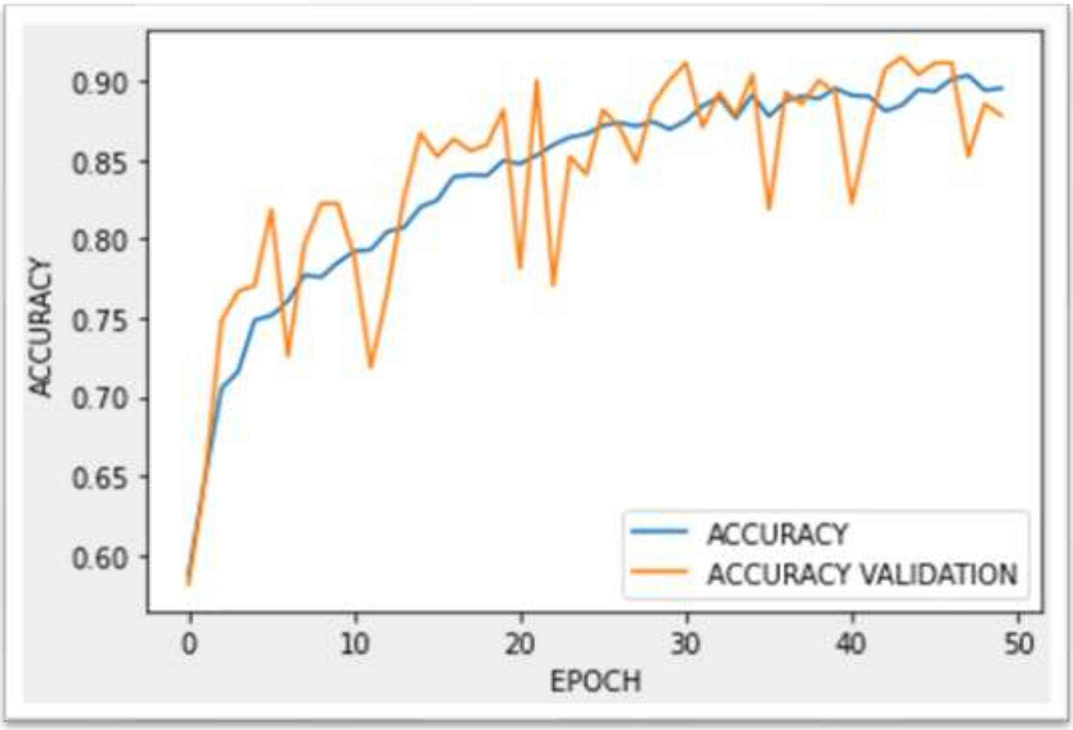


Figure 10 ACCURACY VS EPOCH

```
plt.plot(epochs_Two,losses_Two,"k-",label="LOSS")
plt.plot(epochs_Two,val_losses_Two,"r-.",label="LOSS VALIDATION")
plt.title("LOSS & LOSS VAL")
plt.xlabel("EPOCH")
plt.ylabel("LOSS & LOSS VAL")
plt.legend()
plt.show()
```

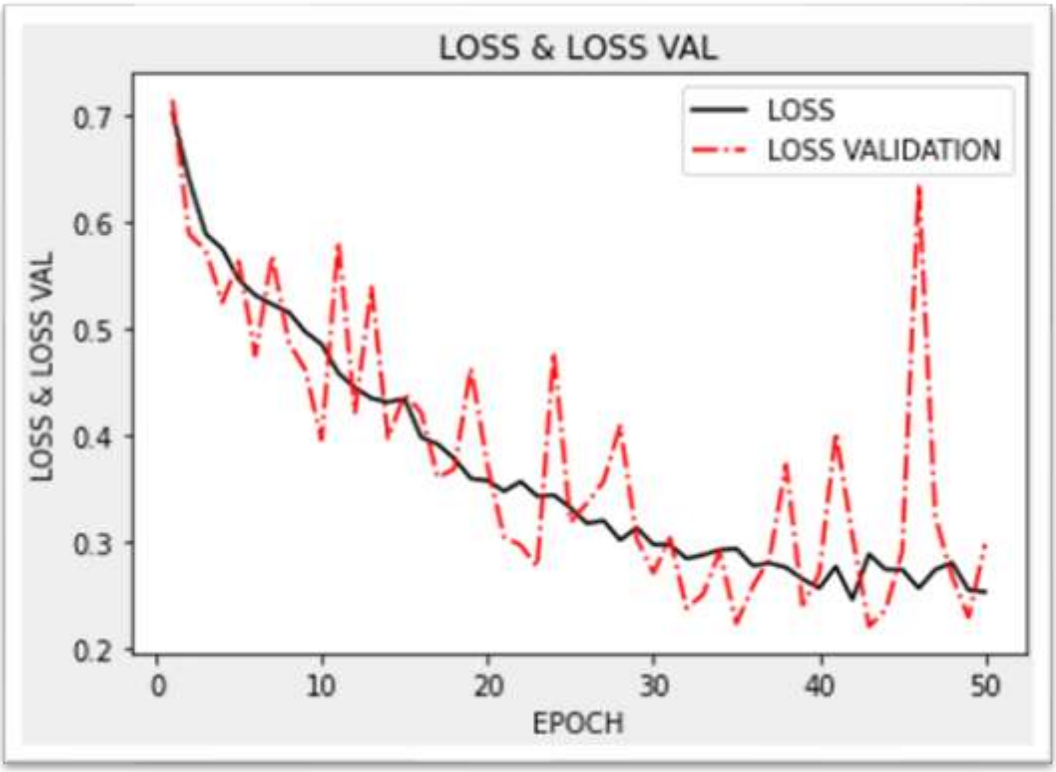


Figure 11 LOSS & LOSS VAL VS EPOCH

```
plt.plot(epochs_Two,acc_Two,"k-",label="ACCURACY")
plt.plot(epochs_Two,val_acc_Two,"r-.",label="ACCURACY VALIDATION")
plt.title("ACCURACY & ACCURACY VAL")
plt.xlabel("EPOCH")
plt.ylabel("ACCURACY & ACCURACY VAL")
plt.legend()
plt.show()
```

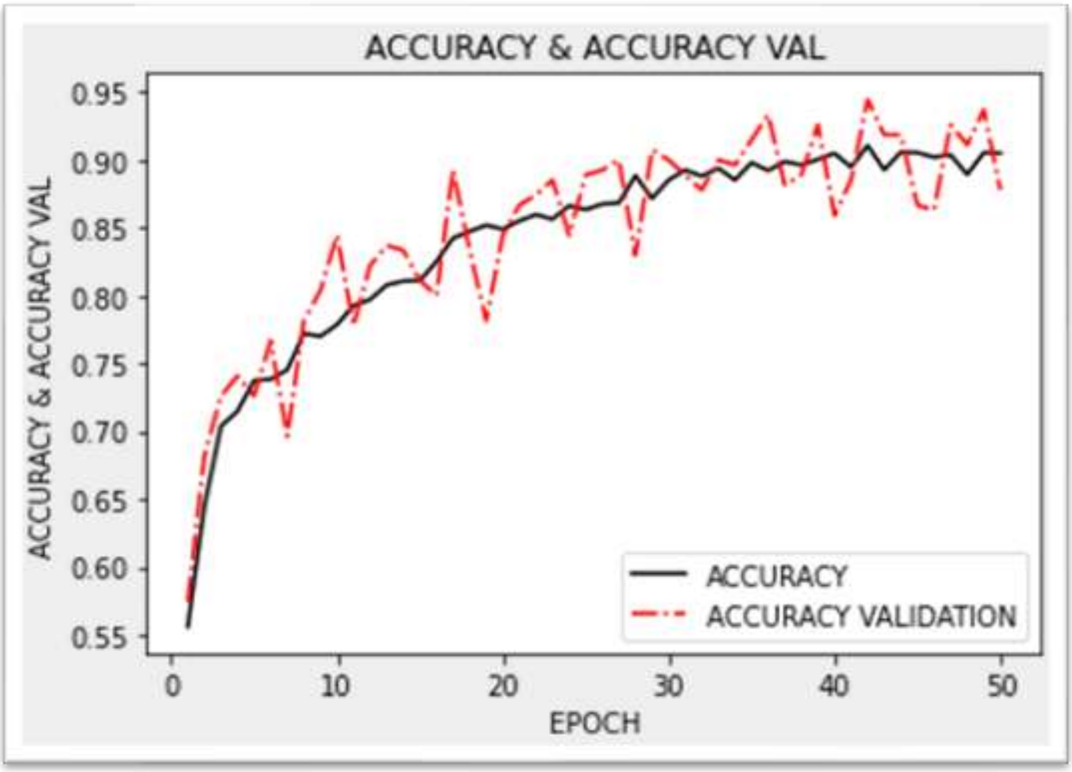


Figure 12 ACCURACY & ACCURACY VAL VS EPOCH

Dict_Summary_Two = pd.DataFrame(ANN_Model_Two.history)

Dict_Summary_Two.plot()

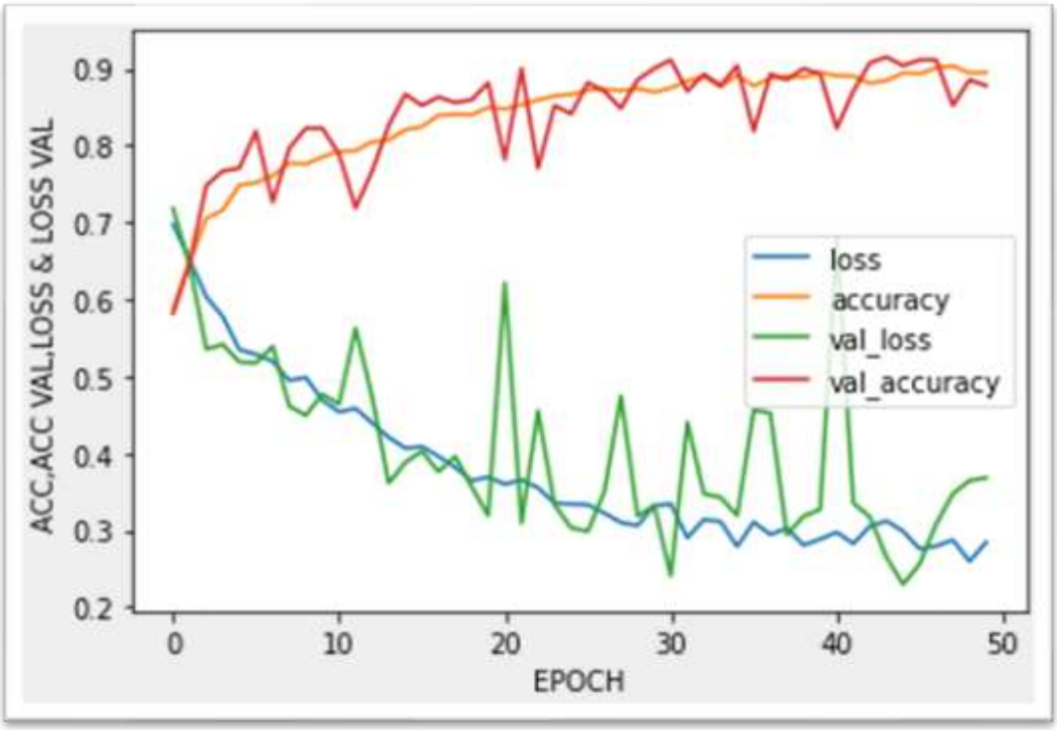
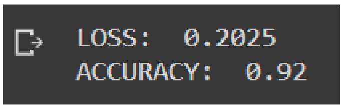


Figure 13 ACC, ACC VAL, LOSS & LOSS VAL VS EPOCH

PREDICTION SCORE ON DIVIDED DATA

```
Model_Results_Two = Model_Two.evaluate(Test_Set,verbose=False)
print("LOSS:  " + "%.4f" % Model_Results_Two[0])
print("ACCURACY:  " + "%.2f" % Model_Results_Two[1])
```



```
LOSS:  0.2025
ACCURACY:  0.92
```

PREDICTION PROCESS

```
fig, axes = plt.subplots(nrows=5,
                         ncols=5,
                         figsize=(20, 20),
                        subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(Main_Data_Prediction["JPG"].loc[i]))
    ax.set_title(f"PREDICTION:{Last_Prediction_Two[i]}")
plt.tight_layout()
plt.show()
```

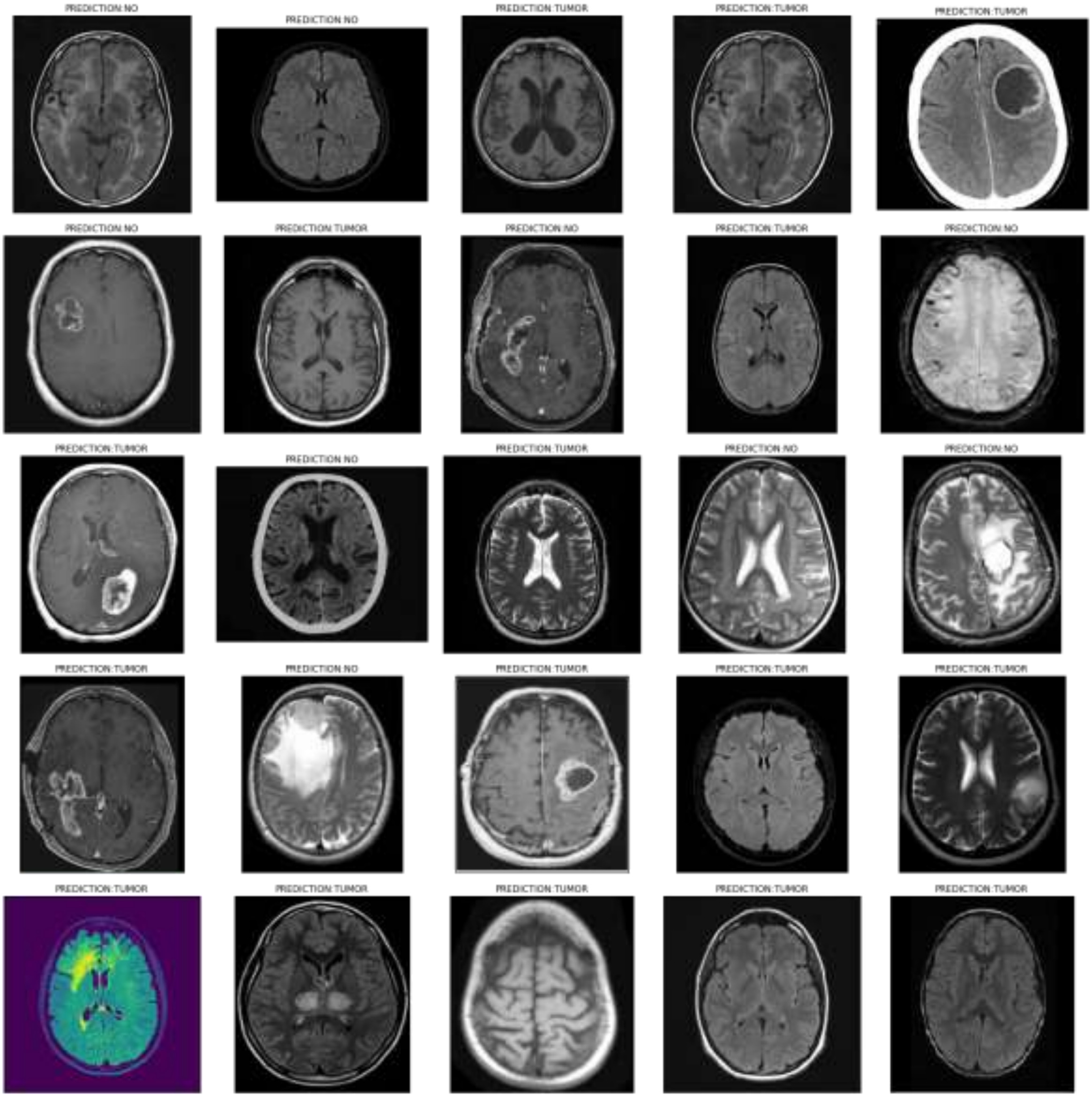


Figure 14 PREDICTION WITH AGUMENTATION

CHAPTER 5

CONCLUSION

We proposed a computerized method for the segementation and identification of a brain tumor using the Convolutional Neural Network. The input MR images are read from the google drive using the file path and converted into grayscale images. These images are pre-processed using adaptive bilateral filtering technique for the elimination of noises that are present inside the original image. With the help of image augmentation to the data, and Convolutional Neural Network, which helps in figuring out the tumor region in the MR images. The proposed model had obtained an accuracy of 92% and yields promising results without any reeors and much less computational time.

CHAPTER 6

FUTURE ENHANCEMENT

It is observed on extermination that the proposed approach needs a vast training set for better accurate results; in the field of medical image processing, the gathering of medical data is a tedious job, and, in few cases, the datasets might not be available. In all such cases, the proposed algorithm must be robust enough for accurate recognition of tumor regions from MR Images. The proposed approach can be further improvised through in cooperating weakly trained algorithms that can identify the abnormalities with a minimum training data and also self-learning algorithms would aid in enhancing the accuracy of the algorithm and reduce the computational time

CHAPTER 7

REFERENCES

[1] A. Sivaramakrishnan And Dr.M.Karnan "A Novel Based Approach For Extraction Of Brain Tumor In MRI Images Using Soft Computing Techniques," International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2, Issue 4, April 2013.

[2] Asra Aslam, Ekram Khan, M.M. Sufyan Beg, Improved Edge Detection Algorithm for Brain Tumor Segmentation, Procedia Computer Science, Volume 58,2015, Pp 430-437, ISSN 1877-0509.

[3] B.Sathya and R.Manavalan, Image Segmentation by Clustering Methods: Performance Analysis, International Journal of Computer Applications (0975 – 8887) Volume 29– No.11, September 2011.

[4] Devkota, B. & Alsadoon, Abeer & Prasad, P.W.C. & Singh, A.K. & Elchouemi, A.. (2018). Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction. Procedia Computer Science. 125. 115- 123. 10.1016/j.procs.2017.12.017.

[5] K. Sudharani, T. C. Sarma and K. Satya Rasad, "Intelligent Brain Tumor lesion classification and identification from MRI images using k-NN technique," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 777-780. DOI: 10.1109/ICCICCT.2015.7475384

[6] Kaur, Jaskirat & Agrawal, Sunil & Renu, Vig. (2012). A Comparative Analysis of Thresholding and Edge Detection Segmentation Techniques. International Journal of Computer Applications.vol. 39.pp. 29-34. 10.5120/4898-7432.

[7] Li, Shutao, JT-Y. Kwok, IW-H. Tsang and Yaonan Wang. "Fusing images with different focuses using support vector machines." IEEE Transactions on neural networks 15, no. 6 (2004): 1555-1561.

[8] M. Kumar and K. K. Mehta, "A Texture based Tumor detection and automatic Segmentation using Seeded Region Growing Method," International Journal of 49 Computer Technology and Applications, ISSN: 2229-6093, Vol. 2, Issue 4, PP. 855- 859 August 2011.

[9] Mahmoud, Dalia & Mohamed, Eltaher. (2012). Brain Tumor Detection Using Artificial Neural Networks. Journal of Science and Technology. 13. 31-39.

[10] Marroquin J.L., Vemuri B.C., Botello S., Calderon F. (2002) An Accurate and Efficient Bayesian Method for Automatic Segmentation of Brain MRI. In: Heyden A., Sparr G., Nielsen M., Johansen P. (eds) Computer Vision — ECCV 2002. ECCV 2002. Lecture Notes in Computer Science, vol 2353. Springer, Berlin, Heidelberg.

[11] Minz, Astina, and Chandrakant Mahobiya. "MR Image Classification Using Adaboost for Brain Tumor Type." 2017 IEEE 7th International Advance Computing Conference (IACC) (2017): 701-705.

[12] Monica Subashini.M, Sarat Kumar Sahoo, "Brain MR Image Segmentation for TumorDetection using Artificial Neural Networks," International Journal of Engineering and Technology (IJET), Vol.5, No 2, Apr-May 2013.

[13] S. Li, J.T. Kwok, I.W Tsang, and Y. Wang, ―Fusing Images with Different Focuses using Support Vector Machines, Proceedings of the IEEE transaction on Neural Networks, China, November 2007.

[14] H. Yu and J.L. Fan, ―Three-level Image Segmentation Based on Maximum Fuzzy Partition Entropy of 2-D Histogram and Quantum Genetic Algorithm, Advanced Intelligent Computing Theories, and Applications. With Aspects of Artificial Intelligence. Lecture Notes in Computer Science, Berlin, Heidelberg 2008.

[15] P.S. Mukambika, K Uma Rani, "Segmentation and Classification of MRI Brain Tumor," International Research Journal of Engineering and Technology (IRJET), Vol.4, Issue 7, 2017, pp. 683 – 688, ISSN: 2395-0056

[16] Pan, Yuehao & Huang, Weimin & Lin, Zhiping & Zhu, Wanzheng & Zhou, Jiayin & Wong, Jocelyn & Ding, Zhongxiang. (2015). Brain tumor grading based on Neural Networks and Convolutional Neural Networks. Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine 50 and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. 2015. 699-702. 10.1109/EMBC.2015.7318458.

PAPER PUBLICATION STATUS

Publication process not yet started