

DETECTION OF ABNORMALITIES IN BRAIN USING MACHINE LEARNING IN MEDICAL IMAGE ANALYSIS

**Sathyabama Institute of Science and
Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in Computer
Science and Engineering

By

**DEEPIKA S
(Reg. No.38110121)
GAYATHRI S
(Reg. No.38110158)**



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI
SALAI,
CHENNAI – 600119, TAMILNADU**

MARCH

-

2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **DEEPIKA S (38110121), GAYATHRI S(38110158)** who carried out the project entitled “**Detection of abnormalities in Brain using machine learning in medical image analysis**” under my supervision from November 2021 to May 2022.

Internal Guide

Mr.A.Yovan Felix M.E

Head of the Department

Dr.L.Lakshmanan M.E., Ph.D.

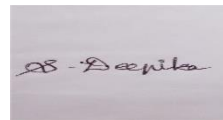
Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **DEEPIKA S, GAYATHRI S** hereby declare that the Project Report entitled **DETECTION OF ABNORMALITIES IN BRAIN USING MACHINE LEARNING IN MEDICAL IMAGE ANALYSIS** done by me under the guidance of **Mr.A.Yovan Felix** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

A small rectangular image showing a handwritten signature in black ink on a light-colored background. The signature appears to be "DS - Deepika".

DATE:

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **MR.A. Yovan felix M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

Automated defect detection in medical imaging has become the emergent field in several medical diagnostic applications. Automated detection of tumor in MRI is very crucial as it provides information about abnormal tissues which is necessary for planning treatment. The conventional method for defect detection in magnetic resonance brain images is human inspection. This method is impractical due to large amount of data. Hence, trusted and automatic classification schemes are essential to prevent the death rate of human. So, automated tumor detection methods are developed as it would save radiologist time and obtain a tested accuracy. The MRI brain tumor detection is complicated task due to complexity and variance of tumors. In this project, we propose the machine learning algorithms to overcome the drawbacks of traditional classifiers where tumor is detected in brain MRI using machine learning algorithms. Machine learning and image classifier can be used to efficiently detect cancer cells in brain through MRI.

INDEX

	TABLE OF CONTENTS	
CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	I
	LIST OF FIGURES	II
	LIST OF TABLES	III
	LIST OF ABBREVIATIONS	IV
1	INTRODUCTION	1
2	LITERARY REVIEW	3
3	AIM AND SCOPE OF THE PROJECT	5
	3.1 AIM OF THE PROJECT	5
	3.2 SCOPE OF THE PROJECT	5
	3.2.1 AUTOMATED MEDICAL IMAGING	6
	3.2.2 ADVANTAGES	6
	3.2.3 DISADVANTAGES	6
4	WORKING THEORY OF THE PROJECT	8
	4.1 ARTIFICIAL INTELLIGENCE	8
	4.2 BRAIN TUMOR	8
	4.3 BASIC OPERATION OF NEURAL NETWORK	9
	4.4 TRANSFER LEARNING	10
	4.5 ACTIVATION FUNCTION	11
	4.6 CONVOLUTION NEURAL NETWORK	11

5	IMPLEMENTATION AND METHODOLOGY	14
	5.1 SOFTWARE REQUIREMENT	14
	5.2 HARDWARE REQUIREMENT	14
	5.3 DATA SET	14
	5.4 PRE-PROCESSING TECHNIQUES	15
	5.4.1 DATA AGUMENTATION	15
	5.4.2 IMAGE PRE-PROCESSING	16
	5.4.3 SEGMENTATION	16
	5.4.4 FEATURE EXTRACTION	17
	5.4.5 MACHINE LEARNING TRAINING AND TESTING	17
6	RESULT AND DISCUSSION	20
7	CONCLUSION AND FUTURE WORK	24
	7.1 CONCLUSION	24
	7.2 FUTURE WORK	24
	REFERENCE	25
	APPENDIX	27
	A. SOURCE CODE	27
	B. SNAPSHOTS	35
	C.PUPLICATION WITH PLAGARISM REPORT	37

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1	ARCHITECTURE DIAGRAM OF OUR MODEL	13
2	MRI SCAN IMAGE OF BRAIN	15
3	IMAGE AGUMENTATION	16
4	MODEL USING CNN ARCHITETURE	18
5	FLOWCHART OF PROPOSED PROJECT	19
6	TEST ACCURACY RESNET 50	20
7	TEST ACCURACY VGG16	21
8	TEST ACCURACY INCEPTION V3	22

LIST OF TABLES

S.NO	TITLE	PAGE NO
1	COMPARISON OF THREE MODELS	22

LIST OF ABBREVIATIONS

AI - Artificial Intelligence

ML – Machine Learning

MRI – Magnetic Resonance Imaging

CNN – Convolutional Neural Network

GM – Grey Matter

WM – White Matter

CSF – Cerebro Spinal Fluid

SVM – Support Vector Machine

KNN – K-Nearest Neighbour

RF – Random Forest

NN – Neural network

PCA – Principal Component Analysis

IDM – Inverse Difference Moment

GLCM – Gray Level Co-Occurrence Matrix

CHAPTER – 1

INTRODUCTION

1.1 Introduction

Brain tumor is one of the most rigorous diseases in the medical science. An effective and efficient analysis is always a key concern for the radiologist in the premature phase of tumor growth. Histological grading, based on a stereotactic biopsy test, is the gold standard and the convention for detecting the grade of a brain tumor. The biopsy procedure requires the neurosurgeon to drill a small hole into the skull from which the tissue is collected. There are many risk factors involving the biopsy test, including bleeding from the tumor and brain causing infection, seizures, severe migraine, stroke, coma and even death. But the main concern with the stereotactic biopsy is that it is not 100% accurate which may result in a serious diagnostic error followed by a wrong clinical management of the disease.

Tumor biopsy being challenging for brain tumor patients, non-invasive imaging techniques like Magnetic Resonance Imaging (MRI) have been extensively employed in diagnosing brain tumors. Therefore, development of systems for the detection and prediction of the grade of tumors based on MRI data has become necessary. But at first sight of the imaging modality like in Magnetic Resonance Imaging (MRI), the proper visualization of the tumor cells and its differentiation with its nearby soft tissues is somewhat difficult task which may be due to the presence of low illumination in imaging modalities or its large presence of data or several complexity and variance of tumors-like unstructured shape, viable size and unpredictable locations of the tumor.

Automated defect detection in medical imaging using machine learning has become the emergent field in several medical diagnostic applications. Its application in the detection of brain tumor in MRI is very crucial as it provides information about abnormal tissues which is necessary for planning treatment. Studies in the recent literature have also reported that automatic computerized detection and diagnosis of

the disease, based on medical image analysis, could be a good alternative as it would save radiologist time and also obtain a tested accuracy. Furthermore, if computer algorithms can provide robust and quantitative measurements of tumor depiction, these automated measurements will greatly aid in the clinical management of brain tumors by freeing physicians from the burden of the manual depiction of tumors

The machine learning based approaches like Deep ConvNets in radiology and other medical science fields plays an important role to diagnose the disease in much simpler way as never done before and hence providing a feasible alternative to surgical biopsy for brain tumors. In this project, we attempted at detecting and classifying the brain tumor and comparing the results of binary and multi class classification of brain tumor with and without Transfer Learning (use of pre-trained Keras models like VGG16, ResNet50 and Inception v3) using Convolutional Neural Network (CNN) architecture.

CHAPTER -2

LITERARY REVIEW

2.1 LITERARY REVIEW

Krizhevsky et al. 2012 achieved state-of-the-art results in image classification based on transfer learning solutions upon training a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, he achieved top-1 and top-5 error rates of 37.5% and 17.0% which was considerably better than the previous state-of-the-art. He also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. The neural network, which had 60 million parameters and 650,000 neurons, consisted of five convolutional layers, some of which were followed by max-pooling layers, and three fully-connected layers with a final 1000-way SoftMax. To make training faster, he used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers he employed a recently-developed regularization method called —dropoutll that proved to be very effective.

Simonyan& Zisserman 2014 they investigated the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. These findings were the basis of their ImageNet Challenge 2014 submission, where their team secured the first and the second places in the localization and classification tracks respectively. Their main contribution was a thorough evaluation of networks of increasing depth using architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers after training smaller versions of VGG with less weight layers.

Pan & Yang 2010'ssurvey focused on categorizing and reviewing the current progress on transfer learning for classification, regression and clustering problems. In this

survey, they discussed the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multitask learning and sample selection bias, as well as co-variate shift. They also explored some potential future issues in transfer learning research. In this survey article, they reviewed several current trends of transfer learning

Szegedy et al. 2015 proposed a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. His results seem to yield solid evidence that approximating the expected optimal sparse structure by readily available dense building blocks is a viable method for improving neural networks for computer vision.

He et al., 2015 introduced the ResNet, which utilizes —skip connections and batch normalization. He presented a residual learning framework to ease the training of networks that are substantially deeper than those used previously. He explicitly reformulated the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. He provided comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset he evaluated residual nets with a depth of up to 152 layers—8×deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. He also presented analysis on CIFAR-10 with 100 and 1000 layers.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 AIM OF THE PROJECT

- ✓ Machine learning techniques are used for recognizing patterns that can be applied to medical images and predict the abnormalities in brain.
- ✓ Analyzing the various performance metrics of the machine learning algorithm.
- ✓ Extract useful information from the medical images to help the doctor.

3.2 SCOPE OF THE PROJECT

- The AI-enabled technologies are plying a bigger role in analysing such images with accuracy. Actually, these medical images are used to train the machines through computer vision and then a capable AI machine scan and analyse the images with possible malady detected.
- Scanning such images through machines is not possible and not even reliable if it is not well-trained to detect the illness accurately. So, a huge quantity of medical images with similar issues or health problems symptoms are used to train the AI model.
- Human eye or radiologist can easily detect the problems as they have learned such things during their studies and medical training. Similarly, machine will not able to find the issue in a medical image annotation, unless you train them by labelling or highlighting the issue and show it to machine learning algorithms that can perceive the specific patterns and learn to give the right prediction.

3.2.1 Automated Medical Imaging with Accuracy

Once a successful AI model is developed, it can diagnosis the different types of diseases in short span of time with better accuracy compare to radiologist. The AI-enabled machines also help to annotate the images at large scale and train more such models to develop and integrate the automated medical imaging analysis process at hospitals and medical centres helping doctors as well as patients get faster and more affordable treatments and medical care.

Getting the such annotated medical images as a healthcare training data is not difficult but getting the accurate data sets become a challenging task for AI developers to build a right model that can predict with accuracy. Cogito is one of the leaning companies providing, medical imaging data with best level of accuracy at most competitive pricing. It is also providing machine learning validation healthcare data sets dentistry and other sub-fields of medical treatments.

3.2.2 Advantages

1. It is considered as the best ml technique for image classification due to high accuracy.
2. Image pre-processing required is much less compared to other algorithms.
3. It is used over feed forward neural networks as it can be trained better in case of complex images to have higher accuracies.
4. It reduces images to a form which is easier to process without losing features which are critical for a good prediction by applying relevant filters and reusability of weights
5. It can automatically learn to perform any task just by going through the training data i.e., there no need for prior knowledge
6. There is no need for specialized hand-crafted image features like that in case of SVM, Random Forest etc.

3.2.3 Disadvantages

1. It requires a large training data.
2. It requires appropriate model.

3. It is time consuming.
4. It is a tedious and exhaustive procedure.
5. While convolutional networks have already existed for a long time, their success was limited due to the size of the considered network

CHAPTER 4

WORKING THEORY OF OUR PROJECT

4.1 Artificial Intelligence

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems enabling it to even mimic human behavior. Its applications lie in fields of Computer Vision, Natural Language Processing, Robotics, Speech Recognition, etc. Advantages of using AI are improved customer experience, accelerate speed to market, develop sophisticated products, enable cost optimization, enhance employee productivity and improve operational efficiency.

Machine Learning (ML) is a subset of AI which is programmed to think on its own, perform social interaction, learn new information from the provided data and adapt as well as improve with experience. Although training time via Deep Learning (DL) methods is more than Machine Learning methods, it is compensated by higher accuracy in the former case. Also, DL being automatic, large domain knowledge is not required for obtaining desired results unlike in ML. Fig: A diagram showing the sub-classes of Artificial Intelligence

4.2 Brain tumor

In medical science, an anomalous and uncontrollable cell growth inside the brain is recognized as tumor. Human brain is the most receptive part of the body. It controls muscle movements and interpretation of sensory information like sight, sound, touch, taste, pain, etc. The human brain consists of Grey Matter (GM), White Matter (WM) and Cerebrospinal Fluid (CSF) and on the basis of factors like quantification of tissues, location of abnormalities, malfunctions & pathologies and diagnostic radiology, a presence of tumor is identified. A tumor in the brain can affect such sensory information and muscle movements or even results in more dangerous situation which includes death.

Depending upon the place of commencing, tumor can be categorized into primary tumors and secondary tumors. If the tumor is originated inside the skull, then the tumor is known as primary brain tumor otherwise if the tumor 's initiation place is somewhere else in the body and moved towards the brain, is called the secondary tumors.

Process of brain tumor segmentation can be manual selection of ROI, Semi-automatic and fully-automatic. Popular machine learning algorithms for classification of brain tumor are Artificial Neural Network, Convolutional Neural Network, k-Nearest Neighbor (kNN), Decision Tree, Support Vector Machine (SVM), Naïve Bayes and Random Field (RF). Here, we are using Convolutional Neural Network (CNN) for the detection and classification of the brain tumor.

4.3 Basic Operation of Neural Networks

Neural Networks (NN) form the base of deep learning, a subfield of machine learning where the algorithms are inspired by the structure of the human brain. NN take in data, train themselves to recognize the patterns in this data and then predict the outputs for a new set of similar data. NN are made up of layers of neurons. These neurons are the core processing units of the network.

First, we have the input layer which receives the input; the output layer predicts our final output. In between, exist the hidden layers which perform most of the computations required by our network. Our brain tumor images are composed of 128 by 128 pixels which make up for 16,384 pixels.

Each pixel is fed as input to each neuron of the first layer. Neurons of one layer are connected to neurons of the next layer through channels. Each of these channels is assigned a numerical value known as 'weight'.

The inputs are multiplied to the corresponding weight and their sum is sent as input to the neurons in the hidden layer. Each of these neurons is associated with a numerical value called the 'bias' which is then added to the input sum.

This value is then passed through a threshold function called the 'activation function'. The result of the activation function determines if the particular neuron will get activated or not. An activated neuron transmits data to the neurons of the next layer over the channels.

In this manner the data is propagated through the network this is called 'forward

propagation'. In the output layer the neuron with the highest value fires and determines the output. The values are basically a probable.

The predicted output is compared against the actual output to realize the 'error' in prediction. The magnitude of the error gives an indication of the direction and magnitude of change to reduce the error. This information is then transferred backward through our network. This is known as 'back propagation'.

Now based on this information the weights are adjusted. This cycle of forward propagation and back propagation is iteratively performed with multiple inputs. This process continues until our weights are assigned such that the network can predict the type of tumor correctly in most of the cases. This brings our training process to an end. NN may take hours or even months to train but time is a reasonable trade-off when compared to its scope. Several experiments show that after pre-processing MRI images, neural network classification algorithm was the best more specifically CNN (Convolutional Neural Network) as compared to Support Vector Machine (SVM), Random Forest Field.

4.4 Transfer Learning

A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution. In such cases, knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data labelling efforts. In recent years, transfer learning has emerged as a new learning framework to address this problem.

Transfer learning allows neural networks using significantly less data. With transfer learning, we are in effect transferring the knowledge that a model has learned from a previous task, to our current one. The idea is that the two tasks are not totally disjoint, as such we can leverage whatever network parameters that model has learned through its extensive training, without having to do that training ourselves. Transfer

learning has been consistently proven to boost model accuracy and reduce required training time, less data, less time, more accuracy. Transfer learning is classified to three different settings: inductive transfer learning, transductive transfer learning and unsupervised transfer learning. Most previous works focused on the settings. Furthermore, each of the approaches to transfer learning can be classified into four contexts based on —what to transfer in learning. They include the instance-transfer approach, the feature-representation-transfer approach, the parameter transfer approach and the relational knowledge-transfer approach, respectively. The smaller networks converged & were then used as initializations for the larger, deeper networks- This process is called pre-training. While making logical sense, pre-training is a very time consuming, tedious task, requiring an entire network to be trained before it can serve as an initialization for a deeper network

4.5 Activation Function

Sigmoid function ranges from 0 to 1 and is used to predict probability as an output in case of binary classification while SoftMax function is used for multi-class classification. tanh function ranges from -1 to 1 and is considered better than sigmoid in binary classification using feed forward algorithm. ReLU (Rectified Linear Unit) ranges from 0 to infinity and Leaky ReLU (better version of ReLU) ranges- from -infinity to +infinity. ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0, x)$. ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real-world data would want our ConvNet to learn would be non-negative linear values. There are other nonlinear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two. Stride is the number of pixels that would move over the input matrix one at a time. Sometimes filter does not fit perfectly fit the input image. We have two options: either pad the picture with zeros (zero-padding) so that it fits or drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

4.6 Convolutional Neural Network

Classifier models can be basically divided into two categories respectively which

are generative models based on hand-crafted features and discriminative models based on traditional learning such as support vector machine (SVM), Random Forest (RF) and Convolutional Neural Network (CNN). One difficulty with methods based on hand-crafted features is that they often require the computation of a large number of features in order to be accurate when used with many traditional machine learning techniques. This can make them slow to compute and expensive memory-wise. More efficient techniques employ lower numbers of features, using dimensionality reduction like PCA (Principle Component Analysis) or feature selection methods, but the reduction in the number of features is often at the cost of reduced accuracy. Brain tumor segmentation employ discriminative models because unlike generative modelling approaches, these approaches exploit little prior knowledge on the brain's anatomy and instead rely mostly on the extraction of [a large number of] low level image features, directly modelling the relationship between these features and the label of a given voxel. In our project, we have used the Convolutional Neural Network architecture for Brain tumor Detection and Classification. Convolutional neural network processes closely knitted data used for image classification, image processing, face detection etc. It is a specialized 3D structure with specialized NN analyzing RGB layers of an image. Unlike others, it analyses one image at a time, identifies and extracts important features and uses them to classify the image. Convolutional Neural Networks (ConvNets) automatically learns mid-level and high-level representations or abstractions from the input training data. The main building block used to construct a CNN architecture is the convolutional layer. It also consists of several other layers, some of which are described as bellow:

- Input Layer-It takes in the raw pixel value of input image
- Convolutional Layer- It is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel to generate a feature map Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.
- Activation Layer-It produces a single output based on the weighted sum of inputs
- Pooling Layer-Pooling layers section would reduce the number of parameters when

the images are too large. Spatial pooling (also called subsampling or down sampling) reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- o Max Pooling – taking the largest element in the feature map
- o Average Pooling - taking the average of elements in the feature map
- o Sum Pooling – taking the sum of all elements in the feature map

Fully Connected Layer-The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network. the feature map matrix will be converted as column vector ($x_1, x_2, x_3, \dots$). With the fully connected layers, we combined these features together to create a model. For classifying input image into various classes based on training set.

- Dropout Layer-It prevents nodes in a network from co-adapting to each other.

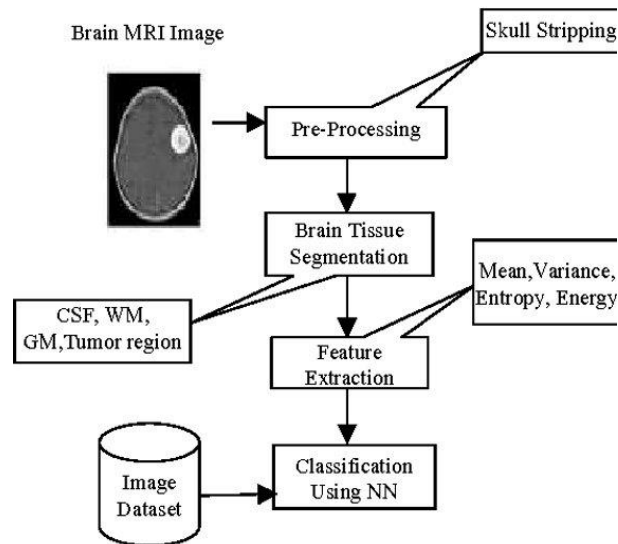


Figure 4.1: Architectural Diagram of our model

CHAPTER 5

IMPLEMENTATION & METHODOLOGY

5.1 Software Requirements:

Python 3 - We have used Python which is a statistical mathematical programming language like R instead of MATLAB due to the following reasons:

1. Python code is more compact and readable than MATLAB
2. The python data structure is superior to MATLAB
3. It is an open source and also provides more graphic packages and data sets Keras (with TensorFlow backend 2.3.0 version) - Keras is a neural network API consisting of TensorFlow, CNTk, Theano etc.

Python packages like NumPy, Matplotlib, Pandas for mathematical computation and plotting graphs, SimpleITK for reading the images which were in .mha format and Mahotas for feature extraction of GLCM Kaggle was used to obtain the online dataset.

GitHub and Stackoverflow was used for reference in case of programming syntax errors. OpenCV (Open-Source Computer Vision) is a library of programming functions aimed at real time computer vision i.e., used for image processing and any operations relating to image like reading and writing images, modifying image quality, removing noise by using Gaussian Blur, performing binary thresholding on images, converting the original image consisting of pixel values into an array, changing the image from RGB to grayscale etc.

It is free to use, simple to learn and supports C++, Java, C, Python.

5.2 Hardware Requirements:

Processor: Intel® Core™ i3-2350M

CPU @ 2.30GHz

Installed memory (RAM):4.00GB

System Type: 64-bit Operating System

5.3 DATASET:

The Multimodal Brain Tumor Segmentation (BraTS) MICCAI [7] has constantly been concentrated upon the assessment of modern-day techniques for the distribution of intracranial tumor in magnetic resonance imaging (MRI) [15] scans. All the imaging datasets were segmented manually, via way of means of one to four, following the identical glossary protocol, and their comments have been permitted via way of means of skilled neuro-radiologists. Therefore, we get one of the datasets from Kaggle which contains MRI scans of the brain. The dataset is divided into three folders.

- 1) yes- This folder contains the 1500 MRI scan images that have a tumor
- 2) no- This folder contains the 1500 MRI scan images that do not have a tumor.
- 3) pred- This folder contains 60 unlabeled MRI scan images for testing purposes. This is a dataset aimed at assisting people in the development of machine learning models for the detection of brain cancers.

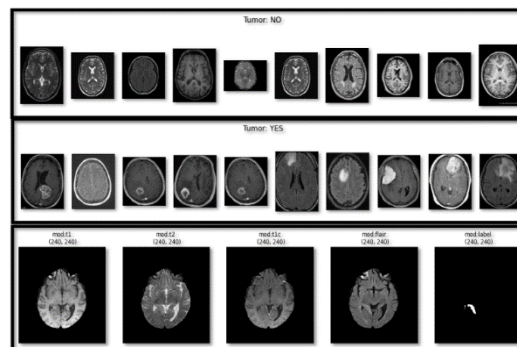


Figure 5.1: MRI scan Images of Brain

5.4 PRE-PROCESSING TECHNIQUES

5.4.1 Data Augmentation

Data augmentation consists of Grey Scaling(RGB/BW to ranges of grey),Reflection(vertical/horizontal flip),Gaussian Blur(reduces image noise),Histogram equalization(increases global contrast),Rotation(may not preserve image size),Translation(moving the image along x or y axis), linear transformation such as random rotation (0-10 degrees), horizontal and vertical shifts, and horizontal and

vertical flips augmentation is done to teach the network desired invariance and robustness properties, when only few training samples are available.

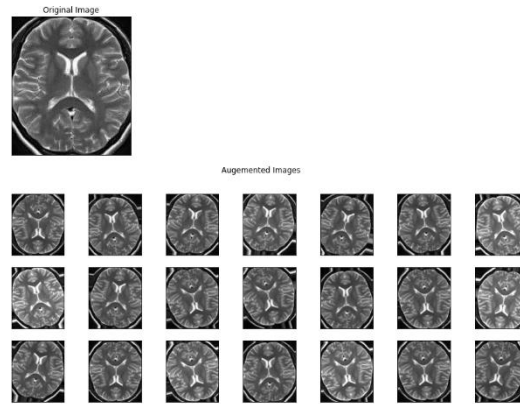


Figure 5.2: Image Augmentation

5.4.2 Image Pre-Processing

Our pre-processing includes rescaling, noise removal to enhance the image, applying Binary Thresholding and morphological operations like erosion and dilation, contour forming (edge-based methodology). In the first step of pre-processing, the memory space of the image is reduced by scaling the gray-level of the pixels in the range 0-255. We used Gaussian blur filter for noise removal as it is known to give better results than Median filter since the outline of brain is not segmented as tumor here.

5.4.3 Segmentation

Brain tumor segmentation involves the process of separating the tumor tissues (Region of Interest – ROI) from normal brain tissues and solid brain tumor with the help of MRI images or other imaging modalities. Its mechanism is based on identifying similar type of subjects inside an image and forms a group of such by either finding the similarity measure between the objects and group the objects having most similarity or finding the dissimilarity measure among the objects and separate the most dissimilar objects in the space.

Segmentation algorithms can be of two type which are bi-clusters (2 sub-parts) or

multi-clustered (more than 2 sub-parts) algorithms. Segmentation can be done by using-Edge Detection, Region Growing, Watershed, Clustering via FCM, Spatial Clustering, Split and Merge Segmentation and Neural Network via MLP(ANN+DWT). In order to identify the tumor region from the brain image, Binary Thresholding can be used (via Region Growing method), which converts a gray scale image to binary image based on the selected threshold values. The problems associated with such approach are that binary image results in loss of texture and the threshold value comes out to be different for different images. Hence, we are looking for a more advanced segmentation algorithm, the watershed algorithm by using Otsu Binarization.

5.4.4 Feature Extraction

Feature Extraction is the mathematical statistical procedure that extracts the quantitative parameter of resolution changes/abnormalities that are not visible to the naked eye. Examples of such features are Entropy, RMS, Smoothness, Skewness, Symmetry, Kurtosis, Mean, Texture, Variance, Centroid, Central Tendency, IDM (Inverse Difference Moment), Correlation, Energy, Homogeneity, Dissimilarity, Contrast, Shade, Prominence, Eccentricity, Perimeter, Area and many more. Feature Extraction is identifying abnormalities. We need to extract some features from images as we need to do classification of the images using a classifier which needs these features to get trained on. We chose to extract GLCM (texture-based features). Gray Level Co-occurrence Matrix (GLCM) features are based on probability density function and frequency of occurrence of similar pixels. GLCM is a statistical method of examining texture that considers the spatial relationship of pixels.

5.4.5 Machine Learning Training and Testing

Models for image classification with weights on ImageNet are Xception, VGG16, VGG19, ResNet, ResNet2, ResNet 50, Inception v2, Inception v3, Mobile Net, MobileNet v2, DenseNet, AlexNet, GoogleNet, NasNet etc. For the implementation of Transfer Learning in our project, we have chosen VGG16, ResNet50 and Inception v3 as our samples. After training the model, we need to validate and fine-tune the parameters and finally test the model on unknown samples where the data undergoes

feature extraction on the basis of which the model can predict the class by matching corresponding labels. To achieve this, we can either split our dataset in the ratio of - 60/20/20 or 70/20/10. We have used the former one. For a given training dataset, back-propagation learning may proceed in one of the following two basic ways: o Pattern/Sequential/Incremental mode where the whole sequence of forward and backward computation is performed resulting in weight adjustment for each pattern. It again starts from the first pattern till errors are minimized, within acceptable levels. It is done online, requires less local storage, faster method and is less likely to be trapped in local minima. Batch mode where the weight upgradation is done after all the N training sets or epochs are presented. After presentation of the full set, weights are upgraded and then again, the whole batch/set is presented iteratively till the minimum acceptable error is arrived at by comparing the target and actual outputs. Training stops when a given number of epochs elapse or when the error reaches an acceptable level or when the error stops improving, we have used this mode during our Machine Learning training by taking the value of N as 30. In supervised network, the network learns by comparing the network output with the correct answer. The network receives feedback about the errors by matching the corresponding labels and weights in different layers and adjusts its weights to minimize the error. It is also known as learning through teacher or Reinforced Learning. In unsupervised network, there is no teacher i.e. labels are not provided along with the data to the network. Thus, the network does not get any feedback about the errors. The network itself discovers the interesting categories or features in the input data. In many situations, the learning goal is not known in terms of correct answers. The only available information is in the correlation of input data or signals. The unsupervised networks are expected to recognize the input patterns, classify these on the basis of correlations and produce output signals corresponding to input categories. It is a type of dynamic programming that trains algorithm using a system of reward and punishment. Agent learns without human interaction and examples and only by interacting with the environment. For our purpose, we have used supervised network or Reinforced Learning for training our model.

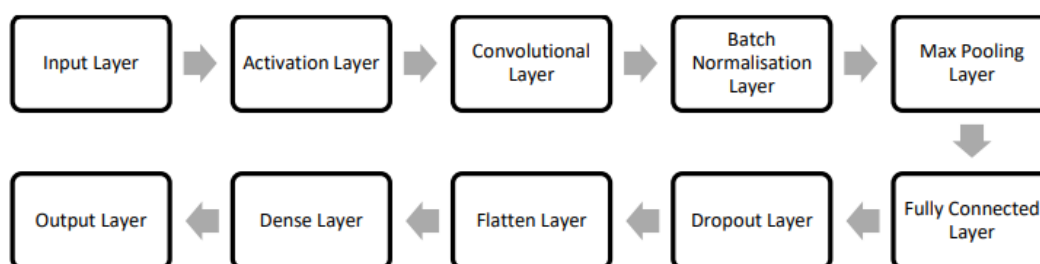


Figure 5.3: A diagram of a model trained from scratch using CNN Architecture

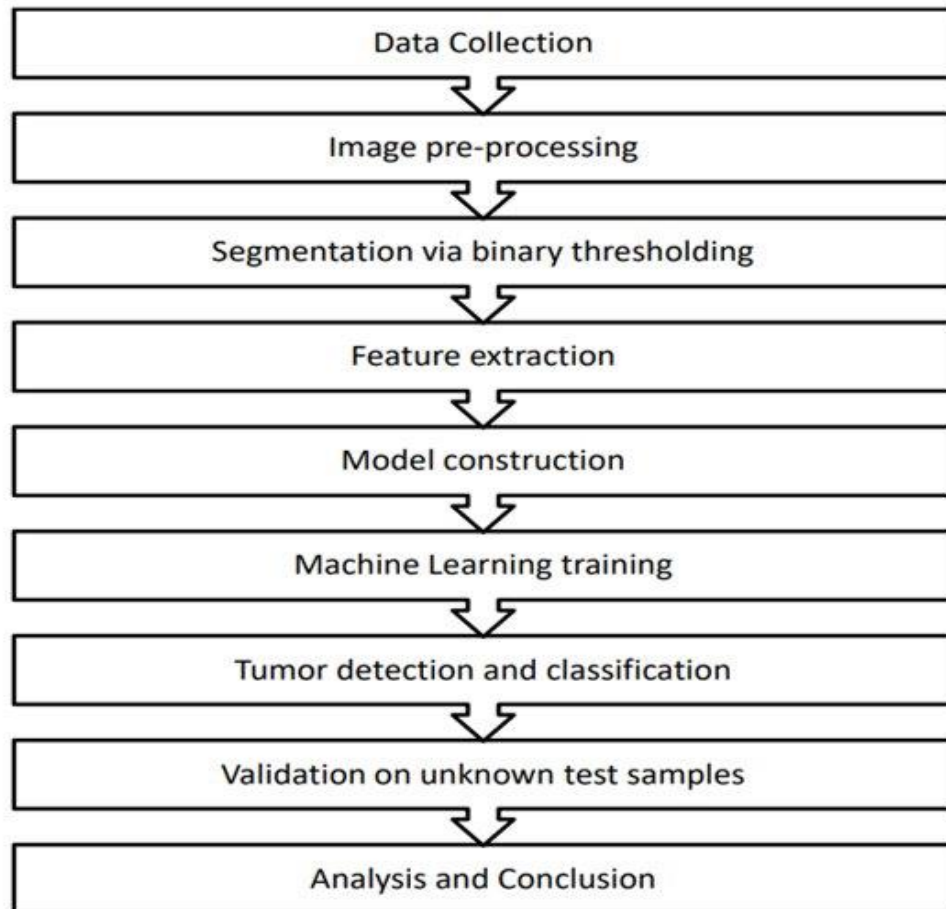


Figure 5.4: Flowchart of Proposed Project

CHAPTER – 6

RESULT DISCUSSION

From the studies, it is observed that the training accuracy is >90% without utilizing a pre-trained Keras model, the overall accuracy is poor, unlike when a pre-trained model is employed. Also, when we trained our dataset without Transfer Learning, it took 40 minutes to compute, whereas when we utilized Transfer Learning, it took just 20 minutes. As a result, using a pre-trained Keras model reduced training and computation time by 50%.

It shows the loss and accuracy of Pre-trained Keras model – Vgg16, its Train and Test accuracy is 97% and 97%, Overall accuracy of this model is 97%.

Test Accuracy = 0.97

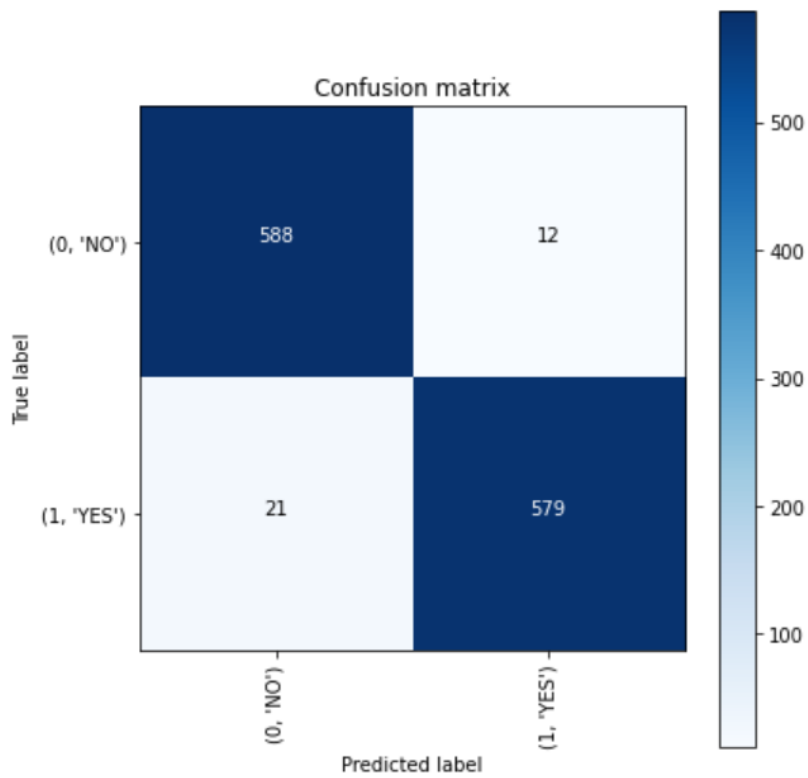


Figure 6.1: Test Accuracy of the model-ResNet50

It shows the loss and accuracy of Pre-trained Keras model – ResNet 50, its Train and Test accuracy is 97% and 97%, Overall accuracy of this model is 97%.

[] Test Accuracy = 0.97

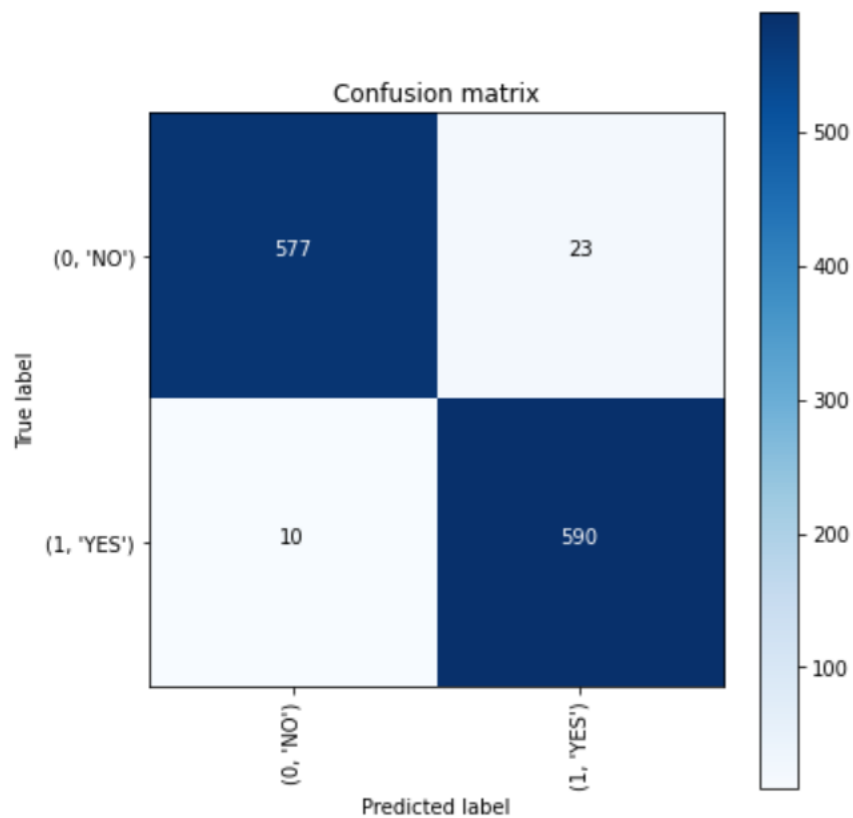


Figure 6.2: Test Accuracy of the model – VGG16

It shows the loss and accuracy of Pre-trained Keras model – Inception_V3, its Train and Test accuracy is 74% and 72%, Overall accuracy of this model is 72%.



Test Accuracy = 0.72

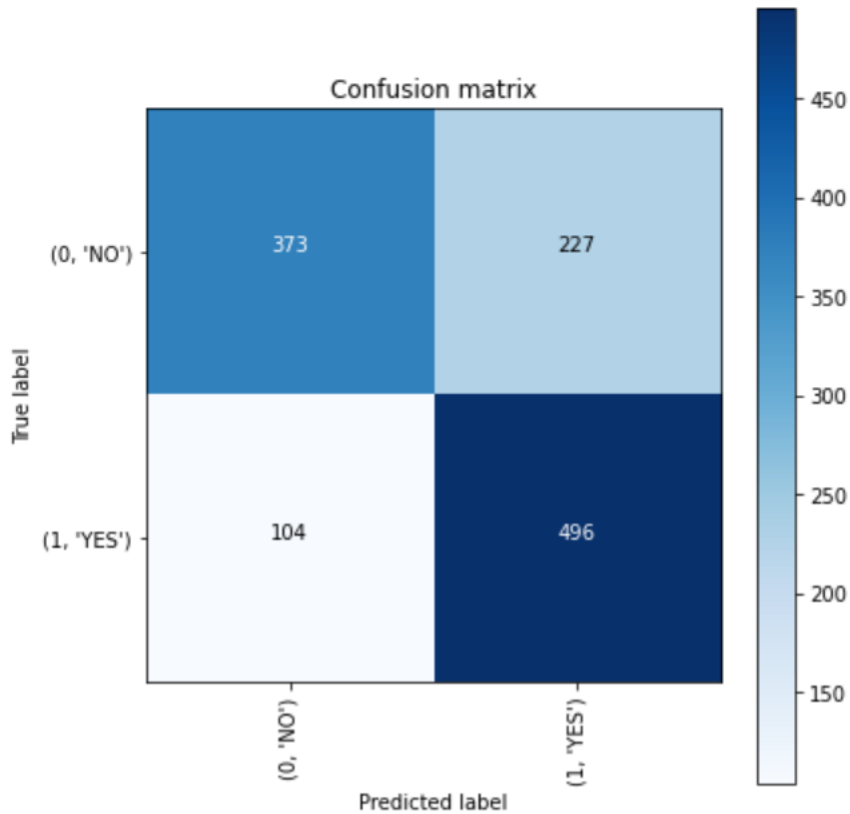


Figure 6.3: Test Accuracy of the Model – Inception_V3

TABLE 6.1: Comparison of three models

Metric	VGG 16	ResNet50	InceptionV3
Train accuracy	0.97	0.97	0.74
Test accuracy	0.97	0.97	0.72
Val accuracy	0.98	0.98	0.73
Overall accuracy	0.9725	0.9725	0.7241666666666666
Precision	0.9727	0.9726	0.7340005393320981
Recall	0.9725	0.9725	0.7241666666666666
F1 score	0.9725	0.9725	0.721237922590887
CKS	0.945	0.945	0.44833333333333336

Here we observe and compare the accuracy of three models namely VGG16, ResNet50, Inception_V3 among these, The ResNet 50 model has the best overall accuracy and F1 score. Therefore, we should use ResNet50 model to predict the abnormalities in the Brain MRI images

CHAPTER – 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

For the segmentation of brain tumors, we suggested a CNN model (ResNet 50) that delivers the best accuracy. Before training the CNN model, the MRI pictures of the brain were pre-processed, cropped, improved, scaled, and augmented. All CNN models' performance is measured using the performance metrics recall, precision, F score specificity, and total accuracy. According to the results of the trial, the ResNet50 Model outperforms and outperforms other CNN models in every way, with an overall accuracy of 90%. We can readily spot anomalies in Brain MRI scans using this model. We can utilize UNet architecture instead of CNN for more complicated datasets to gain more accuracy and less time consumption.

7.2 FUTURE WORK

Build an app-based user interface in hospitals which allows doctors to easily determine the impact of tumor and suggest treatment accordingly Since performance and complexity of ConvNets depend on the input data representation we can try to predict the location as well as stage of the tumor from Volume based 3D images. For more complex datasets, we can use U-Net architecture rather than CNN where the max pooling layers are just replaced by up sampling ones. Ultimately, we would like to use very large and deep convolutional nets on video sequences where the temporal structure provides very helpful information that is missing or far less obvious in static images. Unsupervised transfer learning may attract more and more attention in the future

REFERENCES

- [1] S. Bauer et al., —A survey of MRI-based medical image analysis for brain tumour studies, *Phys. Med. Biol.*, vol. 58, no. 13, pp.97–129, 2013.
- [2] B. Menze et al., —The multimodal brain tumour image segmentation benchmark (BRATS), *IEEE Trans. Med. Imag.*, vol. 34, no.10, pp. 1993–2024, Oct. 2015
- [3] B. H. Menze et al., —A generative model for brain tumour segmentation in multi-modal images, *in Medical Image Computing and Compute.- Assisted Intervention-MICCAI 2010*. New York: Springer, 2010, pp. 151–159
- [4] S. Bauer, L.-P. Nolte, and M. Reyes, —Fully automatic segmentation of brain tumour images using support vector machine classification in combination with hierarchical conditional random field regularization, *in Medical Image Computing and Comput.-Assisted Intervention-MICCAI 2011*. New York: Springer, 2011, pp. 354–361.
- [5] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 4278–4284.
- [6] Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. pp. 1097–1105
- [7] R. Meier et al., —A hybrid model for multimodal brain tumour segmentation, *in Proc. NCIMICCAI BRATS*, 2013, pp. 31–37.
- [8] K. He, X. Zhang, S. Ren, J. Sun--Deep residual learning for image recognition *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770-778
- [9] Sergio Pereira, Adriano Pinto, Victor Alves, and Carlos A. Silva —Brain Tumour Segmentation Using Convolutional Neural Networks in MRI Images *IEEE TRANSACTIONS ON MEDICAL IMAGING*, VOL. 35, NO. 5, MAY 2016
- [10] Rasel Ahmmed, Md.Foissal Hossain —Tumor Detection in Brain MRI Image Using Template based K-means and Fuzzy C-means Clustering Algorithm *in 2016 International Conference on Computer Communication and Informatics (ICCCI -2016)*, Jan. 07 –09, 2016, Coimbatore, INDIA
- [11] S.C. Turaga, J.F. Murray, V. Jain, F. Roth, M. Helmstaedter, K.Briggman, W.Denk, and H.S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538, 2010
- [12] A Reliable Method for Brain Tumor Detection Using CNN Technique NeethuOuseph C1, Asst. Prof. Mrs.Shruti K2 1(Digital electronics ECE, Malabar Institute of Technology, India) 2(Electronics and Communication Engineering, Malabar Institute of Technology, India)
- [13] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)", *IEEE Transactions on Medical Imaging* 34(10), 1993-2024 (2015) DOI: 10.1109/TMI.2014.2377694
- [14] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby, et al., "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", *Nature Scientific Data*, 4:170117 (2017) DOI: 10.1038/sdata.2017.117

- [15] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, et al., "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge", arXiv preprint arXiv:1811.02629 (2018)
- [16] A.Vidyarthi, N.Mittal, —CLAP: Closely link associated pixel based extraction of brain tumour in MR images, International Advance Computing Conference on Signal Processing and Communication (IACC), pp. 202 -206
- [17] International Journal of Computer Science and communication Vol. 2, No. 2, JulyDecember-2011,pp. 325-331
- [18] k. Ramya and L. K. Joshila Grace —Brain tumor detection based on watershed transformation, International Conference on Communication and Signal Processing (ICCSP),2016.
- [19] Luxit Kapoor and Sanjeev Thakur —A survey on brain tumor detection using Image processing techniques, 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence, 2017.
- [20] Minu Samantaray ,Millee Panigrahi, K.C .Patra ,Avipsa S.Panda and Rina Mahakud —An adaptive filtering technique filtering technique for brain tumor analysis and detection, 10th International Conference on Intelligent and Control (ISCO),2016.
- [21] S.U. Aswathy, G.Glan Deva Dhas and S.S. Kumar —A Survey on detection of brain tumor from MRI Brain images, 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence 2017.
- [22] S. Banerjee, S. Mitra, and B. U. Shankar, —Synergetic neuro-fuzzy feature selection and classification of brain tumours, in 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2017, pp.1–6
- [23] Karen Simonyan & Andrew Zisserman-- Very deep convolutional networks for large-scale image recognition and Visual Geometry Group, Department of Engineering Science, University of Oxford, Published as a conference paper at ICLR 2015
- [24] Sinno Jialin Pan and Qiang Yang -- A Survey on Transfer Learning , IEEE transactions on knowledge and data engineering, vol. 22, no. 10, October 2010

APPENDICES

A.SOURCE CODE

```
from google.colab import drive
drive.mount('/content/gdrive')
import os
os.environ['KAGGLE_CONFIG_DIR']='/content/gdrive/MyDrive/Kaggle_dataset'
%cd /content/gdrive/MyDrive/Kaggle_dataset
!ls
! kaggle datasets download -d abhranta/brain-tumor-detection-mri
!mkdir brain-tumor-detection-mri
!mv brain-tumor-detection-mri.zip brain-tumor-detection-mri
%cd brain-tumor-detection-mri
! unzip brain-tumor-detection-mri.zip
from IPython.display import clear_output
!pip install imutils
clear_output()
import numpy as np
import pandas as pd
import cv2
from PIL import Image
import scipy

import tensorflow as tf
from tensorflow.keras.applications import *
from tensorflow.keras.optimizers import *
from tensorflow.keras.losses import *
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
from tensorflow.keras.callbacks import *
from tensorflow.keras.preprocessing.image import *
from tensorflow.keras.utils import *
# import pydot

from sklearn.metrics import *
from sklearn.model_selection import *
import tensorflow.keras.backend as K

from tqdm import tqdm, tqdm_notebook
!pip install colorama
from colorama import Fore
import json
```

```

import matplotlib.pyplot as plt
import seaborn as sns
from glob import glob
from skimage.io import *
%config Completer.use_jedi = False
import time
from sklearn.decomposition import PCA
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import lightgbm as lgb
import xgboost as xgb
import numpy as np
from tqdm import tqdm
import cv2
import os
import shutil
import itertools
import imutils
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix

import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
from plotly import tools
from keras.preprocessing.image import ImageDataGenerator
from keras.applications.vgg16 import VGG16, preprocess_input
from keras import layers
from keras.models import Model, Sequential
from tensorflow.keras.optimizers import Adam, RMSprop
from keras.callbacks import EarlyStopping

init_notebook_mode(connected=True)
RANDOM_SEED = 123

print("All modules have been imported")
!apt-get install tree
clear_output()
# create new folders
!mkdir TRAIN TEST VAL TRAIN/YES TRAIN/NO TEST/YES TEST/NO VAL/YES
VAL/NO

```

```

!tree -d
IMG_PATH="/content/gdrive/MyDrive/Kaggle_dataset/brain-tumor-detection-
mri/Brain_Tumor_Detection"
# split the data by train/val/test
ignored = {"pred"}
# split the data by train/val/test
for CLASS in os.listdir(IMG_PATH):
    if CLASS not in ignored:
        if not CLASS.startswith('.'):
            IMG_NUM = len(os.listdir(IMG_PATH + "/" + CLASS))
            for (n, FILE_NAME) in enumerate(os.listdir(IMG_PATH + "/" + CLASS)):
                img = IMG_PATH + '/' + CLASS + '/' + FILE_NAME
                if n < 300:
                    shutil.copy(img, 'TEST/' + CLASS.upper() + '/' + FILE_NAME)
                elif n < 0.8*IMG_NUM:
                    shutil.copy(img, 'TRAIN/' + CLASS.upper() + '/' + FILE_NAME)
                else:
                    shutil.copy(img, 'VAL/' + CLASS.upper() + '/' + FILE_NAME)
def plot_confusion_matrix(cm, classes,
                           normalize=False,
                           title='Confusion matrix',
                           cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.figure(figsize = (6,6))
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=90)
    plt.yticks(tick_marks, classes)
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    thresh = cm.max() / 2.
    cm = np.round(cm,2)
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

```

```

plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
TRAIN_DIR = 'TRAIN/'
TEST_DIR = 'TEST/'
VAL_DIR = 'VAL/'
IMG_SIZE = (224,224)
X_train, y_train, labels = load_data(TRAIN_DIR, IMG_SIZE)
X_test, y_test, _ = load_data(TEST_DIR, IMG_SIZE)
X_val, y_val, _ = load_data(VAL_DIR, IMG_SIZE)
y = dict()
y[0] = []
y[1] = []
for set_name in (y_train, y_val, y_test):
    y[0].append(np.sum(set_name == 0))
    y[1].append(np.sum(set_name == 1))

trace0 = go.Bar(
    x=['Train Set', 'Validation Set', 'Test Set'],
    y=y[0],
    name='No',
    marker=dict(color='#33cc33'),
    opacity=0.7
)
trace1 = go.Bar(
    x=['Train Set', 'Validation Set', 'Test Set'],
    y=y[1],
    name='Yes',
    marker=dict(color='#ff3300'),
    opacity=0.7
)
data = [trace0, trace1]
layout = go.Layout(
    title='Count of classes in each set',
    xaxis={'title': 'Set'},
    yaxis={'title': 'Count'}
)
fig = go.Figure(data, layout)
iplot(fig)
print("Graph printed")
def plot_samples(X, y, labels_dict, n=50):

```

```
"""
```

Creates a gridplot for desired number of images (n) from the specified set

```
"""
```

```
for index in range(len(labels_dict)):
    imgs = X[np.argwhere(y == index)][:n]
    j = 10
    i = int(n/j)

    plt.figure(figsize=(15,6))
    c = 1
    for img in imgs:
        plt.subplot(i,j,c)
        plt.imshow(img[0])

        plt.xticks([])
        plt.yticks([])
        c += 1
    plt.suptitle('Tumor: {}'.format(labels_dict[index]))
    plt.show()
```

```
plot_samples(X_train, y_train, labels, 30)
def crop_imgs(set_name, add_pixels_value=0):
```

```
"""
```

Finds the extreme points on the image and crops the rectangular out of them

```
"""
```

```
set_new = []
for img in set_name:
    gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
    gray = cv2.GaussianBlur(gray, (5, 5), 0)

    # threshold the image, then perform a series of erosions +
    # dilations to remove any small regions of noise
    thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
    thresh = cv2.erode(thresh, None, iterations=2)
    thresh = cv2.dilate(thresh, None, iterations=2)

    # find contours in thresholded image, then grab the largest one
    cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    c = max(cnts, key=cv2.contourArea)
```

```

# find the extreme points
extLeft = tuple(c[c[:, :, 0].argmin()][0])
extRight = tuple(c[c[:, :, 0].argmax()][0])
extTop = tuple(c[c[:, :, 1].argmin()][0])
extBot = tuple(c[c[:, :, 1].argmax()][0])

ADD_PIXELS = add_pixels_value
new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS, extLeft[0]-
ADD_PIXELS:extRight[0]+ADD_PIXELS].copy()
set_new.append(new_img)

return np.array(set_new)
import imutils
img = cv2.imread('/content/gdrive/MyDrive/Kaggle_dataset/brain-tumor-detection-
mri/VAL/NO/no729.jpg')
img = cv2.resize(
    img,
    dsize=IMG_SIZE,
    interpolation=cv2.INTER_CUBIC
)
gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)

# threshold the image, then perform a series of erosions +
# dilations to remove any small regions of noise
thresh = cv2.threshold(gray, 45, 255, cv2.THRESH_BINARY)[1]
thresh = cv2.erode(thresh, None, iterations=2)
thresh = cv2.dilate(thresh, None, iterations=2)

# find contours in thresholded image, then grab the largest one
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
c = max(cnts, key=cv2.contourArea)

# find the extreme points
extLeft = tuple(c[c[:, :, 0].argmin()][0])
extRight = tuple(c[c[:, :, 0].argmax()][0])
extTop = tuple(c[c[:, :, 1].argmin()][0])
extBot = tuple(c[c[:, :, 1].argmax()][0])

```

```

# add contour on the image
img_cnt = cv2.drawContours(img.copy(), [c], -1, (0, 255, 255), 4)

# add extreme points
img_pnt = cv2.circle(img_cnt.copy(), extLeft, 8, (0, 0, 255), -1)
img_pnt = cv2.circle(img_pnt, extRight, 8, (0, 255, 0), -1)
img_pnt = cv2.circle(img_pnt, extTop, 8, (255, 0, 0), -1)
img_pnt = cv2.circle(img_pnt, extBot, 8, (255, 255, 0), -1)

# crop
ADD_PIXELS = 0
new_img = img[extTop[1]-ADD_PIXELS:extBot[1]+ADD_PIXELS, extLeft[0]-
ADD_PIXELS:extRight[0]+ADD_PIXELS].copy()
plt.figure(figsize=(15,6))
plt.subplot(141)
plt.imshow(img)
plt.xticks([])
plt.yticks([])
plt.title('Step 1. Get the original image')
plt.subplot(142)
plt.imshow(img_cnt)
plt.xticks([])
plt.yticks([])
plt.title('Step 2. Find the biggest contour')
plt.subplot(143)
plt.imshow(img_pnt)
plt.xticks([])
plt.yticks([])
plt.title('Step 3. Find the extreme points')
plt.subplot(144)
plt.imshow(new_img)
plt.xticks([])
plt.yticks([])
plt.title('Step 4. Crop the image')
plt.show()
X_train_crop = crop_imgs(set_name=X_train)
X_val_crop = crop_imgs(set_name=X_val)
X_test_crop = crop_imgs(set_name=X_test)
base_Neural_Net= VGG16(input_shape=(224,224,3), weights='imagenet',
include_top=False)
model=Sequential()
model.add(base_Neural_Net)

```

```

model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(256,kernel_initializer='he_uniform'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1,activation='sigmoid'))

```

```

for layer in base_Neural_Net.layers:
    layer.trainable = False

```

```

model.compile(
    loss='binary_crossentropy',
    optimizer='adam',
    metrics=['accuracy', 'AUC']
)

```

```

model.summary()
EPOCHS = 30
es = EarlyStopping(
    monitor='val_acc',
    mode='max',
    patience=6
)
history = model.fit_generator(
    train_generator,
    steps_per_epoch=5,
    epochs=EPOCHS,

    validation_data=validation_generator,
    validation_steps=25,
    callbacks=[es]
)
predictions = model.predict(X_train_prep)
predictions = [1 if x>0.5 else 0 for x in predictions]

```

```

accuracy = accuracy_score(y_train, predictions)
print('Train Accuracy = %.2f' % accuracy)

```

```

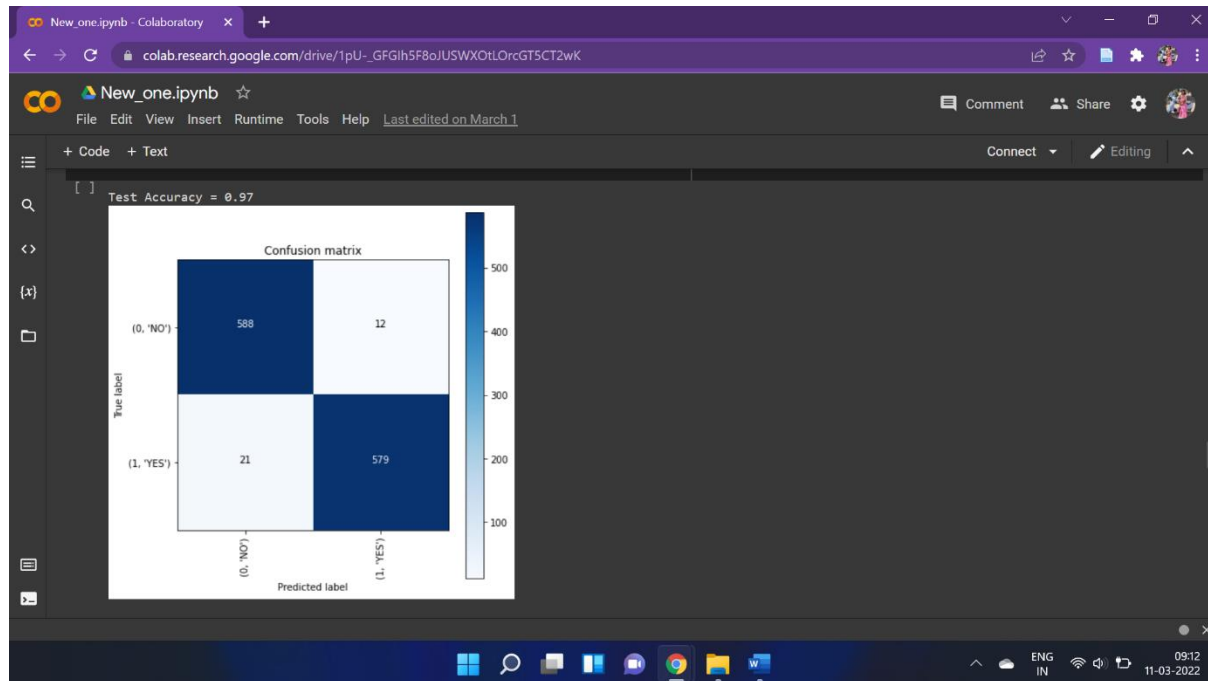
confusion_mtx = confusion_matrix(y_train, predictions)

```

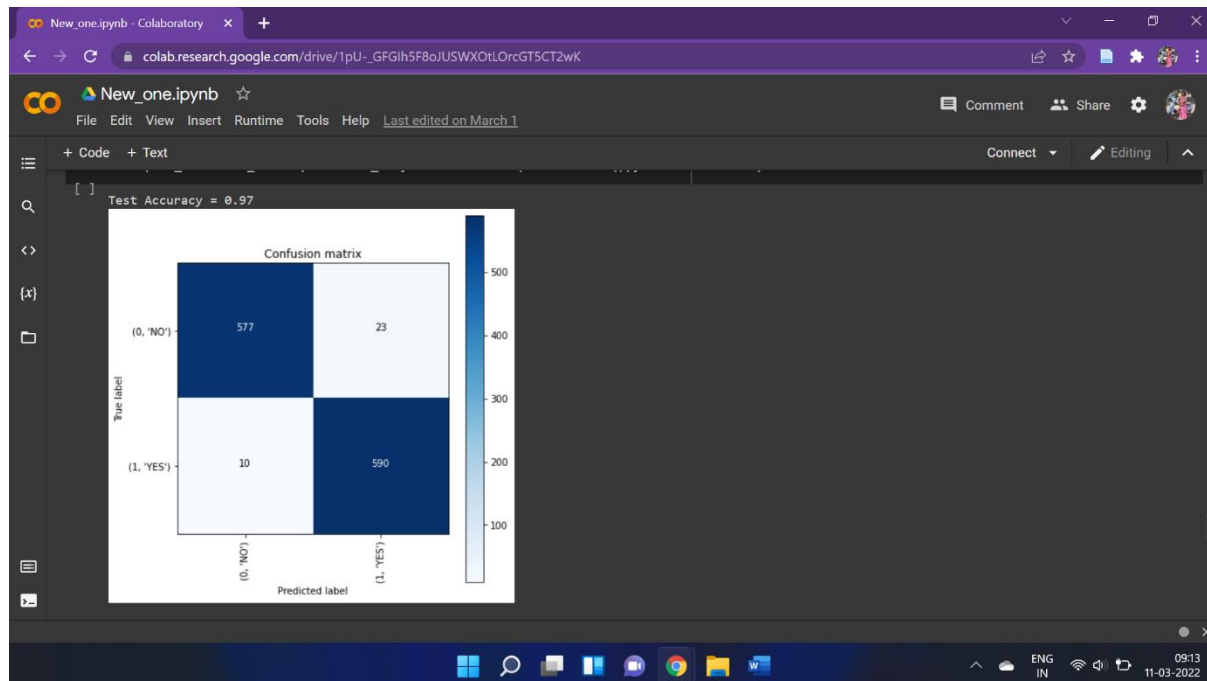
```
cm = plot_confusion_matrix(confusion_mtx, classes = list(labels.items()),  
normalize=False)
```

B.OUTPUT SCREEN SHOTS

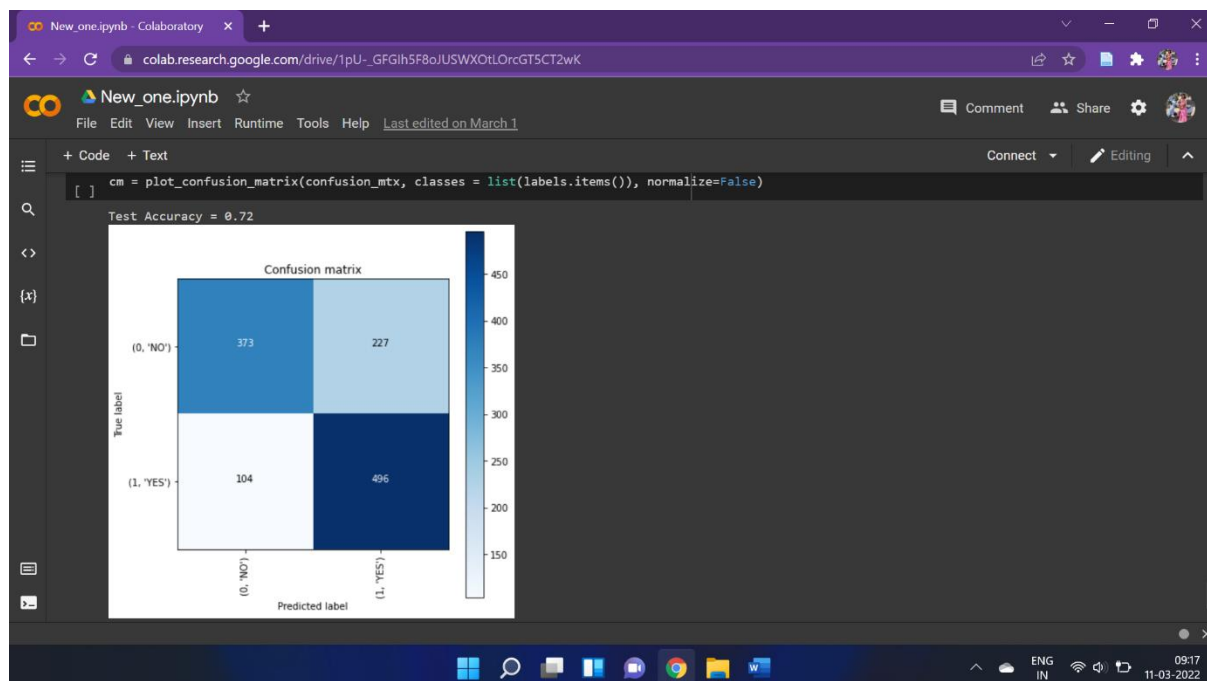
RESNET 50



VGG-16



INCEPTION V3



C.PUPLICATION WITH PLAGARISM REPORT

PUPLICATION

Detection of Abnormalities in Brain using Machine Learning in
Medical Image Analysis---one drive.docx

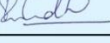
ORIGINALITY REPORT

	SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY) Accredited with Grade 'A' By NAAC 12B Status by UGC [Approved By AICTE] www.sathyabama.ac.in	 
School of Computing Department of Information Technology		
Certificate of Presentation		
This is to certify that		
DEEPIKA S		
of		
Sathyabama Institute of Science and Technology		
has presented a paper entitled		
Detection of Abnormalities in Brain using Machine Learning in Medical Image Analysis		
in the AICTE sponsored		
International Conference on Artificial Intelligence and Machine Learning (IAIM-2022), held from 27 th January 2022 - 29 th January 2022.		
 Dr. R. SUBHASHINI HOD [Information Technology]	 Dr. T. SASIKALA Dean [School of Computing]	 Dr. T. SASIPRABA Vice Chancellor

5

www.isindexing.com
Internet Source

<1 %

	SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY) Accredited with Grade 'A' By NAAC 12B Status by UGC [Approved By AICTE] www.sathyabama.ac.in	 
School of Computing Department of Information Technology		
Certificate of Presentation		
This is to certify that		
GAYATHRI. S		
of		
Sathyabama Institute of Science and Technology		
has presented a paper entitled		
Detection of Abnormalities in Brain using Machine Learning in Medical Image Analysis		
in the AICTE sponsored		
International Conference on Artificial Intelligence and Machine Learning (IAIM-2022), held from 27 th January 2022 - 29 th January 2022.		
 Dr. R. SUBHASHINI HOD [Information Technology]	 Dr. T. SASIKALA Dean [School of Computing]	 Dr. T. SASIPRABA Vice Chancellor

8

oakumen.pub
Internet Source

<1 %

9

ebin.pub
Internet Source

<1 %

10

"Brainlesion: Glioma, Multiple Sclerosis,
Stroke and Traumatic Brain Injuries", Springer
Science and Business Media LLC, 2019
Publication

<1 %

11

"Design, Learning, and Innovation", Springer
Science and Business Media LLC, 2021
Publication

<1 %

12

"Medical Image Computing and Computer
Assisted Intervention - MICCAI 2018",
Springer Science and Business Media LLC,
2018
Publication

<1 %

13

Guangrong Shen, Apostolos Sarris.
"Application of Texture Analysis in Land Cover
Classification of High Resolution Image", 2008

<1 %