BRAIN TUMOR-ANALYSIS AND DETECTION USING DEEP LEARNING TECHNIQUES

A Project Report

Submitted By

AKSHAY RAVI U (UNT19CS004)

MEGHA M R (LUNT19CS063)

SONIYA C J (LUNT19CS065)

VARSHA VALSAN (LUNT19CS066)

To

APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree

Of

Bachelor Of Technology

In

Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSAL ENGINEERING COLLEGE

VALLIVATTOM, THRISSUR

MAY,2023

DECLARATION

We undersigned hereby declare that the project report “Brain Tumor-Analysis and Detection Using Deep Learning Techniques” submitted for the partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of our guide Dr. Sreeraj R., Professor, UEC. This submission represents our idea in our own words and where ideas or words of others have been included; we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/ or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Vallivattom
Date : 02-05-2023

Signature:
Name of Student: Akshay Ravi U

Signature:
Name of Student: Megha M R

Signature:
Name of Student: Soniya C J

Signature:
Name of Student: Varsha Valsan

DEPARTMENT OF COMPUTER SCIENCE

AND ENGINEERING

UNIVERSAL ENGINEERING COLLEGE

VALLIVATTOM, THRISSUR

BONAFIDE CERTIFICATE

This is to certify that the project report titled "**BRAIN TUMOR-ANALYSIS AND DETECTION USING DEEP LEARNING TECHNIQUES**" submitted by "**AKSHAY RAVI U (UNT19CS004), MEGHA M R(LUNT19CS063),SONIYA C J(LUNT19CS065), VARSHA VALSAN (LUNT19CS066)**" to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project by carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

PROJECT GUIDE	**PROJECT CO-ORDINATOR**	**HEAD OF THE DEPARTMENT**
Dr. Sreeraj R	Ms. Muneebah Mohyiddeen	Dr. Sreeraj R
Professor	Assistant Professor	Professor

Vallivattom

02-05-2023

ACKNOWLEDGEMENT

We wish to record our indebtedness and thankfulness to all those who helped us to prepare this report titled **“BRAIN TUMOR-ANALYSIS AND DETECTION USING DEEP LEARNING TECHNIQUES”** and present it in a satisfactory way.

First and foremost, we thank God Almighty for His providence and for being the guiding light throughout the project.

We feel highly privileged in mentioning **Dr. Jose K Jacob -Principal, UEC** for his co-operation and help. We deeply and wholeheartedly thank **Dr. Sreeraj R., HOD, Computer Science and Engineering** and our honourable guide and supervisor, for his extremely valuable advice, suggestions, encouragement and critical inputs in the preparation of this report.

We are especially thankful to our coordinator, **Ms. Muneebah Mohyiddeen, Assistant Professor, UEC** for giving us valuable suggestions and critical inputs that helped us in the completion of our project. We would like to extend our heartfelt thanks to our guide, **Dr.Sreeraj R, Professor, CSE** for her meticulous guidance and support that helped us in the completion of our project.

We would like to extend our sincere gratitude to all faculties of Computer Science and Engineering department for the support and suggestions that helped us in the development of our project to what it is now. We thank our parents for the mental support provided during the course of the project at the times when our energies were the lowest.

ABSTRACT

A brain tumor is an abnormal growth of cells inside the brain or skull ,some are benign, others malignant. Diagnosis and treatment will be applied depending on the tumor type, size and location. Therefore we proposed a modified computer aided diagnosis system that detect tumor conditions, the types of tumor and the current stage. Here we proposed to design a Deep Learning model using Recurrent Neural Networks(RNNs) and Convolutional Neural Networks(CNNs) models for this project. The evaluation of the proposed model is based on the dataset from kaggle. Here we consider the modality of image as MRI. The proposed system will be developed using python.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

MRI	Magnetic Resonance Imaging
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
SVM	Support Vector Machine
RBFN	Radial Basis Function Network
DWT	Discrete Wavelet Transform
FKM	fuzzy k-mean
WNDCHRM	Weighted Neighbour Distance using Compound Hierarchy of Algorithms Representing Morphology
VGGNet	Visual Geometry Group Network
DCNN	Deep Convolutional Neural Network
KNN	K-Nearest Neighbour
PCA-ANN	principal component analysis-artificial neural network
KELM	Kernel Extreme Learning Machine

CHAPTER 1

INTRODUCTION

The advances in biomedical and human intelligence have overcome diverse diseases in the last few years but people are still, suffering from cancer due to its unpredictable nature. This disease is still a significant problem for humanity. Brain tumor cancer is one of the serious and utmost emergent ailments. In the USA, almost 23,000 patients were identified brain tumor cancer in 2015 [1]. In another report of cancer indicators [2], this disease is uniform in both adults and children. Approximately 80,000 fresh cases of primary brain tumors were reported in 2018 [3]: Meningioma represented 36.3% (29,320), Gliomas 26.5% (21,200), Pituitary tumors represented nearly 16.2% (13,210) and rest of the cases belonged to other types of brain tumor such as Malignant, Medulloblastoma, and Lymphomas. The principal causes of such disease are cancer-related ailment and morbidity. Effective handling of this disease is crucial which depends in its timely and accurate detection. The dedication of a therapy modality relies upon: the degree of the tumor at the time of the investigation, the pathological type, and the category of the tumor. The brain is the most complicated and key part of the human anatomy that contains tissues and nerve cells to regulate the key actions of the whole body like breathing, the operation of our senses and muscles. A cell has its capabilities; with its functionality, some cells are developed normally, some decrease their capabilities, stop their growth, and then become abnormal. Such a bulk group of irregular cells produces the tissue which is described as a tumor. So, brain tumors are independent and irregular propagation of brain cells.

1.1 PURPOSE OF THE PROJECT

In various research papers, the detection of brain tumor is done by applying Machine Learning and Deep Learning algorithms. When these algorithms are applied on the MRI images the prediction of brain tumor is done very fast and a higher accuracy helps in providing the treatment to the patients. 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. In this system, these 2 deep learning techniques are combined to get the accurate and effective prediction results. CNN is considered to be more powerful than RNN and this CNN takes inputs of fixed sizes and generates fixed size outputs. And also RNN includes less feature compatibility when compared to CNN and RNN can handle arbitrary input/output lengths. These benefits of both networks can give better outcomes.

1.2 SCOPE OF THE PROJECT

Over the last decade, from various research results it is being observed that it is very time consuming, but it will get faster if use combination of Recurrent Neural Network and Convolutional Neural Networks in our system. In future, combine two or more segmentation methods to improve the feature extraction for better results. This proposed system has robustly intended to widen this research to trial through huge dataset. And also we are planning the identification of glioma tumor grades. Currently the identification of tumor stages is difficult. There is a possibility of further development for the prediction of tumor stages.

1.3 PROBLEM STATEMENTS

A brain tumor is an abnormal growth of cells inside the brain or skull ,some are benign, others malignant. Diagnosis and treatment will be applied depending on the tumor type, size and location. Therefore we proposed a modified computer aided diagnosis system that detect tumor conditions, the types of tumor and the current stage. Here we proposed to design a Deep Learning model using Recurrent Neural Network (RNNs) and Convolutional Neural Networks (CNNs) models for this project.

CHAPTER 2

LITERATURE SURVEY

The authors in [4] employed a two-dimensional discrete transform based on wavelets and Gabor filters for extraction of features of brain MRI. In this paper, we present a framework for classification of brain tumors in MRI images that combines statistical features and neural network algorithms. This algorithm uses region of interest (ROI), i.e. the tumor segment that is identified either manually by the technician/radiologist or by using any of the ROI segmentation techniques. Mainly focus on feature selection by using a combination of the 2D Discrete Wavelet Transform (DWT) and 2D Gabor filter techniques. Create the features set using a complete set of the transform domain statistical features. For classification, back propagation neural network classifier has been selected to test the features selection impact. To do so, used a large dataset consisting of 3,064 slices of T1-weighted MRI images with three types of brain tumors, Meningioma, Glioma, and Pituitary tumor. Here obtained a total accuracy of 91.9%, and specificity of 96%, 96.29%, and 95.66% for Meningioma, Glioma, and Pituitary tumor respectively. Experimental results validate the effectiveness of the features selection method and indicate that it can compose an effective feature set to be used as a framework that can be combined with other classifications technique to enhance the performance. Here used the 2D Discrete Wavelet transform (DWT), 2D Gabor filter, and first and second order statistics of transform domain data to generate a features pool that encompasses the tumors attributes.

Muneer et al. [5] used the real dataset from clinics of the United States. They used a customized classification algorithm based on Wndchrm. This paper presents automatic glioma tumor grade identification from magnetic resonant images using Wndchrm tool based classifier (Weighted Neighbour Distance using Compound Heirarchy of Algorithms Representing Morphology) and VGG-19 deep Convolutional Neural Network (DNN). For experimentation, DICOM images are collected from reputed government hospital and the proposed intelligent system categorized the tumor into four grades such as low grade glioma, oligodendroglioma, anaplastic glioma and glioblastoma multiform. After pre-processing, features are extracted, optimized and then classified using Wndchrm tool where the most significant features are selected on the basis of Fisher score. In the case of DNN classifier, data augmentation is also performed before applying the images into the deep learning network. The performance of the classifiers are analysed with various measures such as accuracy, precision, sensitivity, specificity and F1-score. Automatic classification of glioma tumor grades using two artificial intelligent systems based on, Weighted Neighbour Distance using Compound Hierarchy of Algorithms Representing Morphology tool (Wndchrm) and VGG-19 deep learning networks is successfully performed and compared. The implemented medical decision support system used MR glioma tumor images of clinically proven cases from Government medical college hospital Kozhikode, India. The proposed system yielded good classification accuracy and categorized the images into four grades such as low grade glioma (Grade I), oligodendroglioma (Grade II), anaplastic glioma (grade III) and glioblastoma mutiforme (Grade IV). According to the performance measures evaluated, both the proposed systems gave better specificity measure with a value higher than 97%. It was observed that VGG-19 DNN based classifier yielded an increase in accuracy of 2.28% compared to the Wndchrm tool based classifier. It is also observed that when the number of input images get increased, the accuracy is also increased where the DNN classifier gave a maximum accuracy of 98.25%. The better performance of the DNN intelligent system substantiates its significant role in clinical applications. The performance of the proposed methodologies is also compared with similar recent works. The experimental results obtained here validate the use of new biomedical feature extraction and optimization tool and the potential application of the deep learning networks for real data classification.

Another study [6], based on CapsNets as a Capsule network model, classified brain tumors. Capsule networks are robust to rotation and affine transformation, and require far less training data, which is the case for processing medical image datasets including brain Magnetic Resonance Imaging (MRI) images. This paper focus to achieve the following four objectives: (i) Adopt and incorporate CapsNets for the problem of brain tumor classification to design an improved architecture which maximizes the accuracy of the classification problem at hand; (ii) Investigate the over-fitting problem of CapsNets based on a real set of MRI images; (iii) Explore whether or not CapsNets are capable of providing better fit for the whole brain images or just the segmented tumor, and; (iv) Develop a visualization paradigm for the output of the CapsNet to better explain the learned features. Our results show that the proposed approach can successfully overcome CNNs for the brain tumor classification problem. For the goal of tumor type classification, two types of images can be used as the input to the aforementioned Capsule network. Can use either the whole brain tissue as the input, or instead, the tumor regions can be segmented first and then use these regions as the input to the classification model. Capsules tend to model everything in the input image, thus they do not perform as good as possible for images with miscellaneous backgrounds. Due to this fact, we expect our Capsule network to have a better result when fed with segmented tumors instead of the whole brain images. The study improved the level of accuracy by bringing a variation in the maps of CapsNet at some convolution layer. The study claimed a pronounced accuracy of 86.50% by using CapsNet in convolution layer. The setup was achieved by using 64 feature map to enhance the accuracy measures.

The aim of this paper [7] is to use develop and evaluate a Magnetic Resonance Imaging (MRI) for brain tumor and seizures classification using Recurrent Neural Network (RNN). Medical Science in Image Processing is an emerging field which has proposed a lot of advanced techniques in detection and analysis of a particular disease. Treatment of brain tumors in recent years is getting more and more challenging due to complex structure, shape and texture of the tumor. Therefore, by advancing in image processing, various methodologies have been proposed to identify the tumors in the brain. The advancement in this field created an urge in me to research more on the techniques and methodologies developed for tumor extraction. Hence, propose a system to extract Tumor from the brain using MRI images. This technique involves different image processing methodologies such as noise removal, filtering, segmentation and morphological operations. Extraction of Brain tumor can be accomplished successfully by performing these operations on MATLAB software. Cross-correlation is computed between the target variable vector and the tumor region to determine how pixels' values of the tumor region are closely related using image processing and RNN technique achieving an accuracy of 96.71%. This paper proposes five different segmentation techniques which are then applied on the image to extract the tumor. These segmentation techniques include image segmentation, image pre-processing, feature extraction, watershed segmentation and RGB conversion. Applying each of the segmentation techniques allows us to determine the most appropriate method to segment the tumor from each of the images.

RNN technique is used for classification of MRI tumor and seizures that is proposed to extract the tumor from the brain image. RNN is an unsupervised classification technique where the initial set of neural links along with the layer centers needs to be initialized before segmenting using this RNN algorithm. Therefore, classifying the tumor using RNN technique entirely depends on the value of neuron in different layers and the mean centers initialized. The RNN manages to show fairly accurate results for all of the images. The result shows that RNNs are a powerful tool for image classification. In the research, the classification using RNN technique proposed have been tested only on 187 images of the dataset. These images are taken under different lighting conditions and at different angles.

This paper [8] proposes an intelligent diagnostic method for early detection of brain tumor based on radial basis function neural network (RBFNN) and efficient deep features of magnetic resonance imaging (MRI) scans. The developed method includes four main modules including the segmentation, feature extraction, classification and learning modules. In the segmentation module, Grab cut method is applied for segmenting tumor region. In the feature extraction module, a Convolutional neural Network (ConvNet) is utilized for extraction of new deep features from segmented images. The extracted deep features are fed into RBFNN in the classification module. In the RBFNN, learning algorithm has a high impact on the network performance. Therefore, a new learning algorithm based on the bees algorithm (BA) has been used in the learning module. The developed method applied on Brain Tumor Segmentation (BraTS) 2015 datasets and the obtained results showed that the developed method is effective and can be used in computer-aided systems to detect brain tumor.

One of the critical challenges in the optimization task is defining a suitable fitness function. Recognition Accuracy (RA) which is calculated through confusion matrix has been utilized as the fitness function of the BA in our developed method. The confusion matrix consists of four entries, namely: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). In this matrix, TP refers to the cases' number which has been classified accurately as HGG, FP refers to the cases' number which has been classified incorrectly when they are LGG, TN refers to the cases' number which has been classified correctly when they are LGG, and FN refers to the cases' number classified incorrectly when they are HGG. RA refers to a test's capability in differentiating between LGG and HGG cases in an accurate way. In this study, a new hybrid method proposed for brain tumor detection and classification, and applied on BraTs 2015 dataset. In order to evaluate the performance of the developed method, several experiment was performed. In the first experiment, ConvNet followed by FLC are implemented for classification, and 96.8% accuracy achieved. In the next experiment, used optimized RBFNN as the classifier The obtained results showed that using efficient features and high performance classifier leads to highest accuracy, 97.6%. The obtained results showed that the developed method, ConvNet-BA-RBFNN, is effective and can be used in computer-aided systems to detect brain tumor.

In [9] presented a method for brain tumors detection using a novel Self Organizing Map (SOM) and fuzzy k-mean (FKM). The concept of a self-organizing map, or SOM, was first put forth by Kohonen. It is a way to reduce data dimensions since it is an unsupervised neural network that is trained using unsupervised learning techniques to build a low-dimensional, discretized representation from the input space of the training samples. Each data from data set recognizes themselves by competeting for representation. K -means clustering algorithm is an unsupervised algorithm and it is used to segment the interest area from the background. But before applying K -means algorithm, first partial stretching enhancement is applied to the image to improve the quality of the image. The two algorithms used two different approaches, in k-means, data will be included in one particular cluster, whereas in FCM, a data can be included in all existing clusters, but with varying degrees of membership, in a range of values. Their segments results have been validated by experienced radiologists. However, their proposed approach is complex and time consuming with real practical applications.

In [10] proposed a technique to improve MRI images quality. Their technique computes the initial value of cluster centers with helping of a subjective algorithm. These type of algorithms process the information or output on the basis of an assumption. Algorithms define how a particular task is to be performed. They provide computers with instructions. They tell machines on assembly lines how specific parts should be put together. Or they evaluate application forms and recommend the best candidates. They used another contrast stretching algorithm to enhance the input image quality. They also used K-mean algorithm in their classification process, but still, there’re lack of classification accuracy. The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major application of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.

In [11] proposed a Principal Component Analysis-Artificial Neural Network (PCA-ANN) for several classed brain tumor classification. Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is one of the popular tools that is used for exploratory data analysis and predictive modelling. It is a technique to draw strong patterns from the given dataset by reducing the variances. They get a number of Region Of Interests (ROIs) by the Content-Based Active Contour (CBAC). Contouring algorithm is something that turns your implicit function into a contour. Marching cubes and dual method are the most noted examples. There is also a simpler algorithm. Using contour detection, the system can detect the borders of objects, and localize them easily in an image. It is often the first step for many interesting applications, such as image-foreground extraction, simple-image segmentation, detection and recognition. Their experiments results showed respectable enhance in the accuracy from 77% without PCA to 91% with PCA.

In [12] proposed an approach to detect brain tumors using the U_Net-based deep convolutional neural network. UNet is a convolutional neural network architecture that expanded with few changes in the CNN architecture. It was invented to deal with biomedical images where the target is not only to classify whether there is an infection or not but also to identify the area of infection. U-Net is a model of Convolutional Neural Network (CNN) which has U-shaped architecture. MRI employs a non-invasive technique and can very well provide soft-tissue contrast and hence, for the detection and description of the brain tumor, this imaging method can be beneficial. classification of the brain tumor in Magnetic Resonance Imaging (MRI) images using the U-Net model, then evaluate parameters that indicate the performance of the model. Also discuss the extraction of the tumor region from brain image and description of the tumor regarding its position and size. Here consider the case of Gliomas, one of the types of brain tumors, which occur in common and can be fatal depending on their position and growth. U-Net is a model of Convolutional Neural Network (CNN) which has U-shaped architecture. MRI employs a non-invasive technique and can very well provide soft-tissue contrast and hence, for the detection and description of the brain tumor, this imaging method can be beneficial. Manual delineation of tumors from brain MRI is laborious, time-consuming and can vary from expert to expert. Their work forms a computer aided technique which is relatively faster and reproducible, and the accuracy is very much on par with ground truth. The results of the work can be used for treatment planning and further processing related to storage or transmission of images. Their method consists of encoding and decoding, which, allowed them to efficiently train their model by performing a set of data augmentation approaches.

In [13] proposed a hybrid model based on a convolutional neural network (CNN) and SVM to detect brain tumors in the MRI images. CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice. Convolution neural network (also known as ConvNet or CNN) is a type of feed-forward neural network used in tasks like image analysis, natural language processing, and other complex image classification problems. It is unique in that it can pick out and detect patterns from images and text and make sense of them. In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though they say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyper-plane in an N-dimensional space that distinctly classifies the data points. They also applied a pre-processing approach on the MRI images, which hugely improves their accuracy score. However, their evaluation process was insufficient because they used only 100 cases for training and 220 cases for testing.

In [14] discussed about the various clustering algorithms such as k-means, fuzzy c-means etc., Fuzzy k-c-means is introduced which is nothing but the combination of k-means and fuzzy c-means clustering algorithms for the betterment of time utilization. The implementation input of the clustering algorithm is obtained from Human brain MRI images. The clustering algorithm is an unsupervised method, where the input is not a labelled one and problem solving is based on the experience that the algorithm gains out of solving similar problems as a training schedule. Clustering algorithms are used to group data points based on certain similarities. There's no criterion for good clustering. Clustering determines the grouping with unlabelled data. It mainly depends on the specific user and the scenario. Kmeans clustering may be the most widely known clustering algorithm and involves assigning examples to clusters in an effort to minimize the variance within each cluster. In image segmentation, you'd mostly use the k-means clustering algorithm as it's quite simple and efficient. Segmentation algorithms partition an image into sets of pixels or regions. The purpose of partitioning is to understand better what the image represents. The recording of these algorithm outcomes is done and compared with the prevailing methods with respect to advantages and disadvantages. The concept of image segmentation is also discussed in this research.

In [15] obtained a solution for denoising by means of adaptive Wiener filtering and non-brain tissue is eliminated by utilizing morphological operations along with the noised reduction inefficient manner. The merging of K-means++ clustering with Gaussian Kernel-based Fuzzy C-means (GKFCM) algorithm is accomplished for image segmentation process which in turn helps in enhancing the stability of the algorithm and reduction in clustering sensitivity parameters. The post-processing of extracted tumor images is accomplished by expending morphological operations. The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise, also having applications in signal processing. The GKFCM algorithm adds kernel information to the traditional fuzzy c-means algorithm.

Overcomes the disadvantage that FCM algorithm can’t handle the small differences between The median filters also take the responsibility of brain tumors precise representations and the outcomes obtained are contrasted with current segmentation algorithms whose performance metrics are accuracy, sensitivity, specificity, and recall are taken as comparison metrics.

In [16] merged Template-based K-means and modified Fuzzy C-means (TKFCM) clustering algorithm for segmentation process by eliminating operators and equipment error thereby achieving robustness. The convolution between gray level intensity is a vital factor in choosing the template in the brain and brain tumor image. The prominence of initial segmentation is attained utilizing template choice by K-means algorithm. A modified FCM technique is utilized from attaining sharp segmented image results based on updated membership and automatic cluster selection which is a red marked tumor. It will reduce the problem of template or gray level selection and noise sensitivity accustomed to FCM and, Region growing. Template based k-means extension has applied through pixel intensity positioning. Very little bit of pixel intensity is not avoided as the template is selected based on the number of gray level to be used and the coarse image. FCM algorithm is modified on the basis of updated membership and number of clusters for the filtered image. The membership values are updated based on the features such as intensity, entropy, contrast, and homogeneity of the MRI brain image. By choosing right membership input & output variables along with adjusted number of feature and cluster the detection of tumor is completed nicely. The whole performance is analyzed through neural network, which is highly accepted in the segmentation field. It is done by preparing train and target data for the relevant image through the network with feed forward back-propagation algorithm. The gray level intensity of normal and abnormal tissue recognition in terms of small deviation is attained by means of TKFCM. The neural network is greatly involved in achieving the TKFCM performances which include improved regression and minimal error and reveals that it is an effective approach for tumor detection in multiple intensity-based brain MRI image.

In [17] utilized Fuzzy-Possibilistic C-Means (FPCM) for segmenting the brain tumor by using clustering method obtained from MR images. The precise tumor region is recognized by utilizing shape-based topological properties. The skull stripping nothing but a pre-processing step involved in brain tissue extraction which is achieved by patch-based K-means. In Fuzzy Possibilistic C Mean (FPCM) the constraint according to which the sum of all typically values of all data to a cluster must be equal to one may lead to problems for big datasets. In order to solve this problem proposed a new and improved algorithm called Possibilistic Fuzzy c means algorithm (PFCM). FPCM constrains the typicality values so that the sum over all data points of typicalities to a cluster is one. The row sum constraint produces unrealistic typicality values for large data sets. PFCM produces memberships and possibilities simultaneously, along with the usual point prototypes or cluster centers for each cluster. The main advantage of fuzzy c – means clustering is that it allows gradual memberships of data points to clusters . This gives the flexibility to express that data points can belong to more than one cluster. The Proposed method reveals superior performance established on MRI standard benchmark datasets depending on volume metrics when compared with the erstwhile advanced systems on the basis of ground truth (manual segmentation).

In [18] established a system for categorizing brain tumors such as meningioma, glioma, and pituitary by utilizing Convolutional Neural Networks (CNNs) architectures such as AlexNet, GoogLeNet, and Visual Geometry Group Network (VGGNet). Specifically, fine-tuning is a process that takes a model that has already been trained for one given task and then tunes or tweaks the model to make it perform a second similar task. With fine-tuning, first change the last layer to match the classes in our dataset, as they have done before with transfer learning. But besides, they also retrain the layers of the network that they want. VGG16 is a 16 layer transfer learning architecture and is quite similar to earlier architectures as it's foundation is based on CNN only but the arrangement is a bit different. VGG16 is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. The transfer learning technique is greatly utilized for fine-tuning and freezing expending brain tumor MRI slices dataset-Figshare. The enhancement of data samples and overfitting diminishing are achieved by the MRI slices application along with data augmentation techniques for result analysis. The outcomes exhibit 98.69% accuracy with respect to classification and detection by means of fine-tuning VGG16 architecture.

In [19] utilized deep transfer learning notion for system classification while the brain MRI images features are extracted using pre-trained GoogLeNet. In addition to it, proven classifier model plays a significant role in classification. The MRI dataset obtained from figshare undergoes patient-level five-fold cross-validation process. This methodology gives 98% mean classification accuracy when contrasted with other prevailing methods. Apart from these various metrics such as Area Under the Curve (AUC), precision, F-score, specificity recall etc. are also investigated and this investigation is carried out with fewer training samples.

In [20] investigated three common forms of brain tumors namely Glioma, Meningioma, and Pituitary by exploiting Convolutional Neural Network (CNN). The CNN implementation is accomplished from a hidden layer by convolution, max-pooling, and flattening layers respectively with the entire connection. The CNN training for analyzing brain tumor utilizes 3064 T-1 weighted Contrast-Enhanced Magnetic Resonance Imaging (CE-MRI) images which is accessible via figshare Cheng. The training accuracy of 98.51% and validation accuracy of 84.19% is attained expending basic architecture and deprived of any prior region-based segmentation which is contrasted with other complicated region-based segmentation algorithms where accuracy lies between 71.39%and 94.68% on identical dataset.

In [21] demonstrated for extracting image hidden features by utilizing CNN beside with Kernel Extreme Learning Machine (KELM). Compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need to be tuned. These hidden nodes can be randomly assigned and never updated (i.e. they are random projection but with nonlinear transforms), or can be inherited from their ancestors without being changed. In most cases, the output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model.

The name "Extreme Learning Machine" (ELM) was given to such models by Guang-Bin Huang. The idea goes back to Frank Rosenblatt, who not only published a single layer perceptron in 1958, but also introduced a multi layer perceptron with 3 layers: an input layer, a hidden layer with randomized weights that did not learn, and a learning output layer. The several brain tumors forms such as meningioma, glioma and pituitary tumor are concentrated for effectiveness evaluation of the projected system in T1-weighted CE-MRI images. The CNN and KELM (KE-CNN) combination are contrasted with Support Vector Machine, Radial Base Function, and some other classifiers which reveal improved outcomes.

In [22] projected a brain metastases segmentation system utilizing deep learning CNN algorithm with the aid of contrast-enhanced T1-weighted MRI datasets. The nonlinear association among abnormal voxels and their neighbours are analyzed expending sequentially connected convolutional filters stack by means of CNN algorithm and derivation of voxel characterization model is done. Brain Tumor Image Segmentation (BRATS) is greatly utilized for validation of multimodal and clinical patients' data with the combination of CNN-based algorithm into automatic brain metastases segmentation. The average Dice Coefficients (DCs) of 0.75±0.07 in the tumor core and 0.81±0.04 in the enhancing tumor is achieved by validating the BRATS data analyzed in 2015 BRATS challenge. Also, an average of DCs 0.67±0.03 is attained using segmentation results including patient cases and additionally value of 0.98±0.01 is attained under the receiver operating characteristic curve area.

In [23] performed various types of classification of brain tumor involving two publicly available datasets by designing a system for deep learning (DL) model constructed on a CNN. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.Initially classification of meningioma, glioma, and pituitary tumor is accomplished by the mentioned technique along with other differentiating among the glioma grades such as Grade II, Grade III, and Grade IV. The datasets comprise 233 and 73 patients with a total of 3064 and 516 images on T1-weighted contrast-enhanced images for the first and second datasets, respectively. The significant outcome of 96.13% and 98.7% overall accuracy for the two case investigated respectively which signpost the model for brain tumor multi-classification capability.

In [24] suggested a methodology for brain tumor classification utilizing Residual Network (ResNet). The first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, Every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when we increase the number of layers, there is a common problem in deep learning associated with that called the Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when we increases number of layers, the training and test error rate also increases.

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Blocks. In this network, we use a technique called skip connections. The skip connection connects activations of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together.

The advantage of adding this type of skip connection is that if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by vanishing/exploding gradient.The anticipated is evaluated by involving benchmark dataset comprising 3064 MRI images of 3 brain tumor types (Meningiomas, Gliomas, and Pituitary tumors). The proposed deep learning model outperforms good results comparatively when compared to existing approaches for diagnosing brain cancer. The finest deep learning model and architecture helps in attaining the enhanced performance which optimally categorizes brain cancer.

CHAPTER 3

FEASIBILITY STUDY

3.1 INTRODUCTION

A project feasibility study is a comprehensive report that examines in detail the five frames of analysis of a given project. It also takes into consideration its four Ps, its risks and POVs, and its constraints (calendar, costs, and norms of quality). The goal is to determine whether the project should go ahead, be redesigned, or else abandoned all together. A feasibility study is an analysis used in measuring the ability and likelihood to complete a project successfully including all relevant factors. It must account for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. Feasibility studies allow companies to determine and organize all the details to make a business work.

A feasibility study helps identify logistical problems, and nearly all business-related problems and their solutions. Feasibility studies can also lead to the development of marketing strategies that convince investors or a bank that investing in the business is a wise choice. Feasibility studies examine potential risks to determine whether they are worth taking. A comprehensive feasibility study can distinguish real economic opportunities from investments that could fail.

3.2 TYPES OF FEASIBILITY STUDY

The feasibility of the system is based on the technical, operational, economic and schedule feasibilities. A feasibility study is an analysis that takes all of a project's relevant factors into account including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it. Feasibility studies also can provide a company's management with crucial information that could prevent the company from entering blindly into risky businesses. Feasibility study examine potential risks to examine whether they are worth taking. A comprehensive feasibility study can distinguish real economic opportunities from investments that could fail.

3.2.1 Technical Feasibility

Technical feasibility study involves checking whether the computers are capable of running models. Today almost all of the systems are equipped to state of the art technology that offers high performance, and the models that we created are smaller, and use less system resources.

3.2.2 Operational Feasibility

This measures how well your product/system will be able to solve. It is used for identifying the importance of certain problem in project and how it is to be solved. It also measures how solution of the problems will work for any project .It analyses the behaviour of the proposed system and whether the proposed system is easier than the existing system for the users of the system. The proposed system helps in identifying Brain tumor and its classification. By using deep learning, the results are provided in real time much faster and thereby solving the network latency issues in the existing systems.

3.2.3 Economic Feasibility

It is used to determine the financials resources of the project. It measures all costs incurred in the development of the new system. It is called cost benefit analysis because it determines the total cost for the development of new system and benefits derived from the new system. Benefits of new system should be more than the cost incurred to achieve profit from new system. The proposed system is built in aim to reduce the overall implementation cost of the system. Since the Deep learning models are placed on the devices the server requirements are highly reduced.

3.2.4 Schedule Feasibility

Schedule feasibility is defined as the likelihood of a project being completed within its scheduled time frame. If the project has a high likelihood of completion by the desired due date then schedule feasibility is considered to be high. We studied about the requirements of our project before 3 months of project initiation. After discussing various problems about the existing system we ended with the exact requirements of the new system and After then we designed the system. We made various design model like data flow diagram, block diagram etc.

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

4.1.1 Server Requirements

- Operating System - Windows 8.1 or better, Mac OS X-Mojave or newer
- Processor - Intel Processors (i3,i5), AMD Processors (A8,A10)
- RAM - 8 GB
- Integrated Graphics Card
- Programming Languages: Python 2.7 and above
- Platform: Jupyter notebook v6.4.11
- Framework: FastAPI
- Supporting Libraries: Tensorflow, Keras, OpenCV etc

4.2 SOFTWARE REQUIREMENTS

4.2.1 Tensorflow

TensorFlow is an open-source library developed by Google primarily for deep learning applications. It also supports traditional machine learning. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind. However, it proved to be very useful for deep learning development as well, and therefore Google open-sourced it. TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors. Multi-dimensional arrays are very handy in handling large amounts of data. TensorFlow works on the basis of data flow graphs that have nodes and edges. As the execution mechanism is in the form of graphs, it is much easier to execute TensorFlow code in a distributed manner across a cluster of computers while using GPUs.

TensorFlow provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use. The TensorFlow team has open sourced a large number of models. It can be found in the tensorflow/models repo. For many of these, the released code includes not only the model graph, but also trained model weights. This means that you can try such models out of the box, and you can tune many of them further using a process called transfer learning.

4.2.2 Python IDLE

Every Python installation comes with an Integrated Development and Learning Environment, which you'll see shortened to IDLE or even IDE. These are a class of applications that help you write code more efficiently. While there are many IDEs for you to choose from, Python IDLE is very bare-bones, which makes it the perfect tool for a beginning programmer.

The best place to experiment with Python code is in the interactive interpreter, otherwise known as a shell. The shell is a basic Read-Eval-Print Loop (REPL). It reads a Python statement, evaluates the result of that statement, and then prints the result on the screen. Then, it loops back to read the next statement.The Python shell is an excellent place to experiment with small code snippets. Can access it through the terminal or command line app on your machine. Can simplify your workflow with Python IDLE, which will immediately start a Python shell when you open it.Its main features are:

- Multi-window text editor with syntax highlighting,autocompletion,smart indent and other.
- Python shell with syntax highlighting.
- Integration debugger with stepping, persistent breakpoints, and call stack visibility.

4.2.3 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012.As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python.There is also a small, bootstrap version of

Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language, including multi-language projects. The conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio

4.2.4 Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust. It is based on the Electron framework which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

4.2.5 Python Tkinter

Tkinter is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest.As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

There are several popular GUI library alternatives available, such as wxPython, PyQt, PySide, Pygame, Pyglet, and PyGTK. Python has a lot of GUI frameworks, but Tkinter is the only framework that’s built into the Python standard library. Tkinter has several strengths. It’s cross-platform, so the same code works on Windows, macOS, and Linux. Visual elements are rendered using native operating system elements, so applications built with Tkinter look like they belong on the platform where they’re run. Although Tkinter is considered the de facto Python GUI framework, it’s not without criticism. One notable criticism is that GUIs built with Tkinter look outdated. TKinter is good choice because of the following reasons:

- Easy to learn.
- Use very little code to make a functional desktop application.
- Layered design.
- Portable across all operating systems including Windows,macOS,and Linux.
- Pre-installed with the standard Python library.

However, Tkinter is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern sheen is unnecessary, and the top priority is to quickly build something that’s functional and cross-platform. Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −Import the Tkinter module,Create the GUI application main window,Add one or more of the above-mentioned widgets to the GUI application.

4.2.6 Python 3.0

Python is an interpreted language. Interpreted languages do not need to be compiled to run.A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not turned into machine code ahead of time. Instead, this happens as the program is running. Python has become one of the most famous programming languages on the world as of late. It's utilized in all that from AI to building sites and programming testing. It tends to be utilized by engineers and non-designers the same. Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

Python's developers try to avoid changing the language to make it better until they have a lot of things to change. Also, they try not to make small repairs, called patches, to unimportantparts of the CPython reference implementation that would make it faster. When speed is important, a Python programmer can move some of the work of the program to other parts written in programming languages like C or PyPy, a just-in-time compiler. It translates a Python script into C and makes direct C-level API calls into the Python interpreter.

Python is usually used for creating sites and programming, task robotization, information investigation, and information representation. Since it's moderately simple to learn, Python has been taken on by numerous non-software engineers like bookkeepers and researchers, for different regular undertakings, such as coordinating funds. Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries libraries have been examined to provide wonderful ends in varied areas like Machine Learning (ML), Deep Learning, etc. These libraries make it a powerful language; it can do many different things. Python's statements include:

- The assignment statement, or the sign. In Python, the statement x = 2 means that the name x is bound to the integer 2. Names can be rebound to many different types in Python, which is why Python is a dynamically typed language. For example, you could now type the statement x = 'spam' and it would work, but it wouldn't in another language like C or C++.
- The if statement, which runs a block of code if certain conditions are met, along with else and elif (a contraction of else if from other programming languages). The elif statement runs a block of code if the previous conditions are not met, but the conditions for the elif statement are met. The else statement runs a block of code if none of the previous conditions are met.
- The for statement, which iterates over an iterable object such as a list and binds each element of that object to a variable to use in that block of code, which creates a for loop.
- The while statement, which runs a block of code as long as certain conditions are met, which creates a while loop.
- The def statement, which defines a function or method.
- The pass statement, which means "do nothing."
- The class statement, which allows the user to create their own type of objects like what integers and strings are.
- The import statement, which imports Python files for use in the user's code.
- The print statement, which outputs various things to the console.

Python's expressions include some that are similar to other programming languages and others that are not

- Addition, subtraction, multiplication, and division, represented by +... and /.
- Exponents, represented by **.
- Python uses the words "and", "or", and "not" for its boolean expressions.
- To compare two values,Python uses==.

CHAPTER 5

IMPLEMENTATION

5.1 SYSTEM ARCHITECTURE

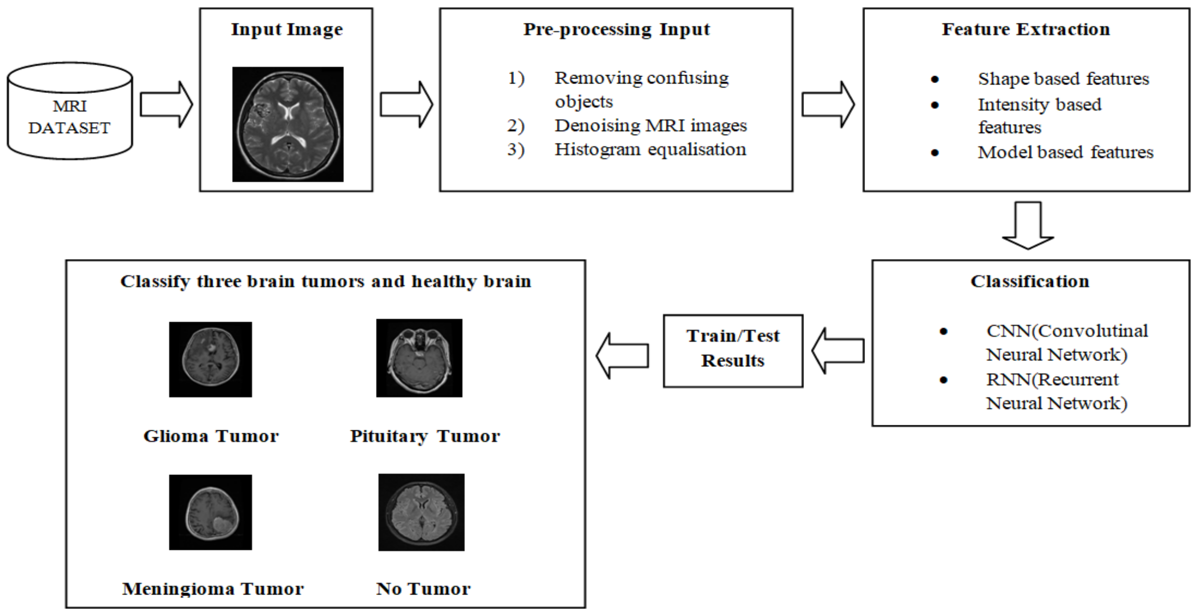


Figure 5.1.1. System Architecture

The system as a whole consists of 3 modules. The first module is the image dataset import and the pre-processing module. Data pre-processing is a process which transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. To ensure reliable findings, the techniques are typically applied at the very beginning of the machine learning and AI development pipeline. The algorithm's ability to quickly analyse the properties of the data is essential for a model to be accurate and exact in its predictions. The dataset is pre-processed in order to check missing values, noisy data, and other inconsistencies before executing it to the algorithm. Pre-processing data before applying it to a machine learning algorithm is a crucial step in the ML workflow. It helps to improve the accuracy, reduce the time and resources required to train the model, prevent overfitting, and improve the interpretability of the model.

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. It can process images and videos to identify objects, faces, or even the handwriting of a human. It is one of the most fundamental and important techniques in image processing. OpenCV is a free open source library used in real-time image processing. It's used to process images, videos, and even live streams, but in this system, we will process images only as a first step.

In pre-processing there are mainly three steps. Initially the dataset path will be defined and the data directories are listed. Then there is a process which is essential for accurate result called image resizing. Image resizing is an important role in image processing technique, to enlarge and reduce the given image size in pixel format. Image interpolation can be divided into two different ways, they are image down-sampling and up-sampling which is necessary when resizing the data for matching either the specific communication channel or the output display. The most frequent justification for diminishing images is to make them more usable for tumour prediction by reducing the size of huge images.

Simply throwing away pixels is how an image is reduced in size.Alternately, increasing t he size of an image adds more pixels, which could degrade the quality.A crucial stage in comput er vision's preprocessing is image resizing.Generally speaking, smaller images allow our machin e learning models to train more quickly.Our network must learn from four times as many pixels i n an input image that is twice as big, which takes time.Remove the confusing objects from images which helps to improve the quality of an image after removing irrelevant image data from image in various applications and domains. Medical images contain lot of irrelevant and unwanted parts in its actual format of the scanned images. To remove such annoying parts in an image, it is required some of the image pre-processing techniques in order for better visualization of the images before finding the diseases in particular. This step provides a fast and accurate means to predict the location of an unwanted object in an image. Sharpness is arguably the most important single image quality factor; it determines the amount of detail an image can convey. Acquisition geometry-Image acquisition geometric factors affecting image quality include a unnecessary or unwanted texts in the images, source to image receptor distance, orientation, the amount of magnification, and size of the focal spot. These issues can be solved by this pre-processing step.

The next important and necessary step is denoising MRI images. Medical images often consist of low-contrast objects corrupted by random noise arising in the image acquisition process. Thus, image denoising is one of the fundamental tasks required by medical imaging analysis. Nonlocal means (NL-means) method provides a powerful framework for denoising. In this work, we investigate an adaptive denoising scheme based on the patch NL-means algorithm for medical imaging denoising. In contrast with the traditional NL-means algorithm, the proposed adaptive NL-means denoising scheme has three unique features. First, we use a restricted local neighbourhood where the true intensity for each noisy pixel is estimated from a set of selected neighbouring pixels to perform the denoising process. Second, the weights used are calculated. Finally, we apply the steering kernel to preserve the details of the images. The proposed method has been compared with similar state-of-art methods over synthetic and real clinical medical images showing an improved performance in all cases analyzed. Non-local means is an algorithm in image processing for image denoising. Unlike "local mean" filters, which take the mean value of a group of pixels surrounding a target pixel to smooth the image, non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared with local mean algorithms. If compared with other well-known denoising techniques, non-local means adds "method noise" (i.e. error in the denoising process) which looks more like white noise, which is desirable because it is typically less disturbing in the denoised product. Recently non-local means has been extended to other image processing applications such as deinterlacing, view interpolation, and depth maps regularization.

After the 2 steps, apply histogram equalization by using a function called equalizeHist. It can be defined as, the mapping of each pixel of the input image to the relating pixels of the output. This method equalizes the intensity values to full range of the histogram to get an enhanced output. It increases the brightness and contrast of each pixel giving rise to dynamic range expansion. But, it does not consider the mean brightness of the input image into account, will gives rise to flattening of the output image histogram, false coloring, annoying artifacts in background, un-natural enhancement, excessive change in brightness, most importantly decreasing the contrast and no brightness preservation. After pre-processing, the data and labels are saved into a pickle file.

Pickle is a tool that may be used to serialise Python object structures, which is the process of transforming an object from memory into a byte stream that can be saved as a binary file on d isc.This binary file can be deserialized into a Python object when it is loaded back into a Python programme. When this tool is used to save data ,we don't have to construct the same object again and again. We will create an object once and then save it into a disk (pickling), and later on, we load this object from the disk (unpickling) without having to create the object again. Pickling is mostly useful in Machine Learning.

First module has been completely executed. Second module is the brain tumor detection model with the combination of RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network). Recurrent Neural Network is a type of artificial neural network commonly used in speech recognition and natural language processing. Recurrent neural networks recognize data's sequential characteristics and use patterns to predict the next likely scenario. A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice. In this system, these 2 deep learning techniques are combined to get the accurate and effective prediction results. CNN is considered to be more powerful than RNN and this CNN takes inputs of fixed sizes and generates fixed size outputs. And also RNN includes less feature compatibility when compared to CNN and RNN can handle arbitrary input/output lengths. These benefits of both networks can give better and accurate outcomes. This process is executed for both training and testing datasets. After that perform train-test-splitting where the data and labels are splitted according to the dataset characteristics such as shapes. Then in both the datasets, expand the dimensions. A class from sklearn library is used here called LabelBinarizer that accepts Categorical data as input and returns an Numpy array. Unlike Label Encoder, it encodes the data into dummy variables indicating the presence of a particular label or not. Encoding make column data using Label Binarizer.

After these steps, let's move onto the model creation. The combined model will be defined in a different python file, i.e, the function Model=CNN_RNN() will direct to a python file known as model.py.This combination will be imported from Mode. Tensorflow library is used here for combining or adding different layers. TensorFlow is one of the best library available for working with Machine Learning on Python. For different Machine Learning tasks you must combine different types of Layers into a Model that can be trained with data to predict future values. TensorFlow.js is supporting different types of Models and different types of Layers. A TensorFlow Model is a Neural Network with one or more Layers.

After the completion of these processes and function call, the model is compiled with the parameters such as loss, optimizer and matrices. Adam in optimizer is derived from Adaptive Moment estimation. This optimization algorithm is a further extension of stochastic gradient descent to update network weights during training. Based on the loss function the optimizer will adjust the weights. A function called fit is used in the training set and then define the no. of epochs in which the number of times that data needed to train. After that the process of validation_data will be conducted as passing the testing data. This process is to evaluate how the model works on that set."Checkpoint" is used to save the epochs and also save the model too. A method "val_acc" is used to find the performance of a testing set, if it is high then it will be saved. Verbose is used to save the model. Plot generation is performed where Matplotlib is a Python plotting package that makes it simple to create two-dimensional plots from data stored in a variety of data structures including lists, numpy arrays, and pandas data frames. Matplotlib uses an object oriented approach to plotting. The purpose of plotting scientific data is to visualize variation or show relationships between variables, but not all data sets require a plot. If there are only one or two points, it is easy to examine the numbers directly, and little or nothing is gained by putting them on a graph. After completing all these functions and processes, the prediction will be performed as the different types of tumor. A confusion matrix is used to show the actual and predicted outcomes. A heat map is used for the system. It is often desirable to show data which depends on two independent variables as a color coded image plot. This is often referred to as a heatmap. If the data is categorical, this would be called a categorical heatmap. Matplotlib's imshow function makes production of such plots particularly easy.

5.2 DESIGN

A system must be designed based on our requirements and the detailed analysis of the system. The particular phase is known as system designing. g. It is the most crucial phase in the development of a system. First a logical design of the system is already determined. The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This logical system design arrived as a result of system analysis is converted into physical system design. The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. The logical design produced during the analysis is turned into a physical design - a detailed description of what is needed to solve original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. Data structure, control process, equipment source, workload and limitation of the system, Interface, documentation, training, procedures of using the system, taking backups and staffing requirement are decided at this stage.

There are several tools and techniques used for describing the system design of the system. These tools and techniques are Flowchart, Data flow diagram (DFD),UML diagram.

5.2.1 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

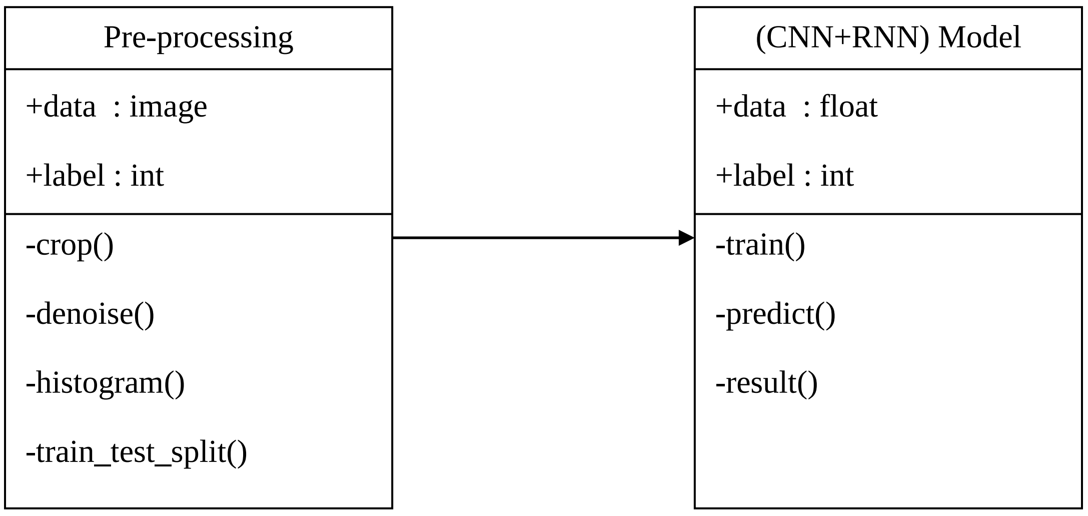


Figure 5.2.1.1 Class Diagram

Figure 5.2.1.1 shows the class diagram. A class diagram looks like a flowchart where classes are represented by boxes with three rectangles inside each box. The class name is located in the top rectangle, its attributes are located in the middle rectangle, and its methods, which are also known as operations, are located in the lower rectangle. A class diagram is a UML diagram type that describes a system by visualizing the different types of objects within a system and the kinds of static relationships that exist among them. It also illustrates the operations and attributes of the classes. Class diagrams give you a sense of orientation. They provide detailed insight into the structure of your systems. At the same time they offer a quick overview of the synergy happening among the different system elements as well as their properties and relationships.

5.2.2 Sequence Diagram

A sequence diagram shows object interaction arranged in time sequence. It depicts the object and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionalities of the scenario. Sequence diagram are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenario. A sequence diagram shows, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple run time scenario in a graphical manner.

Figure 5.2.2.1 shows the sequence diagram . Inputted image is used to create the model in this diagram,and the model itself executes all the operations include pre-processing,feature extraction,training,classification and prediction.The user will then receive the predicted result.

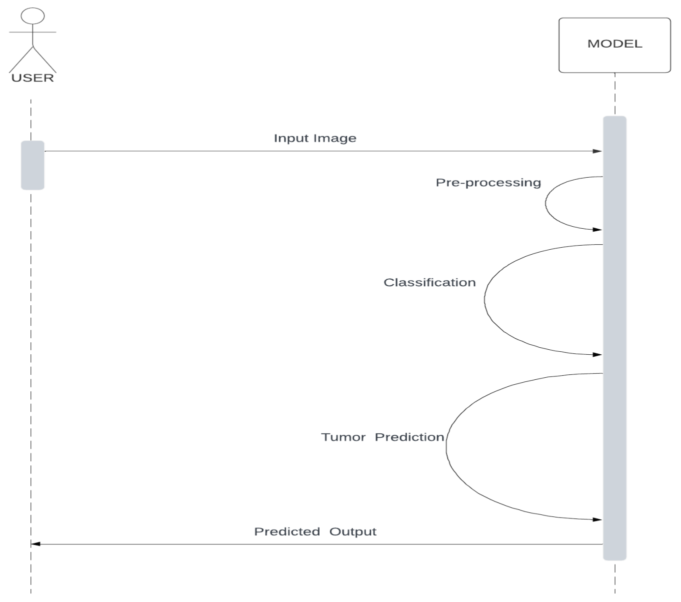


Figure 5.2.2.1 Sequence Diagram

5.2.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

activity diagram :Figure 5.2.2.1 shows the activity diagram. It explains how the models activities works. The user inputted the image into the model as initial activity. Then perform three step pre-processing, feature extraction. After that classification. if the MRI image matches the prediction value then conform the result otherwise the operation is terminated.

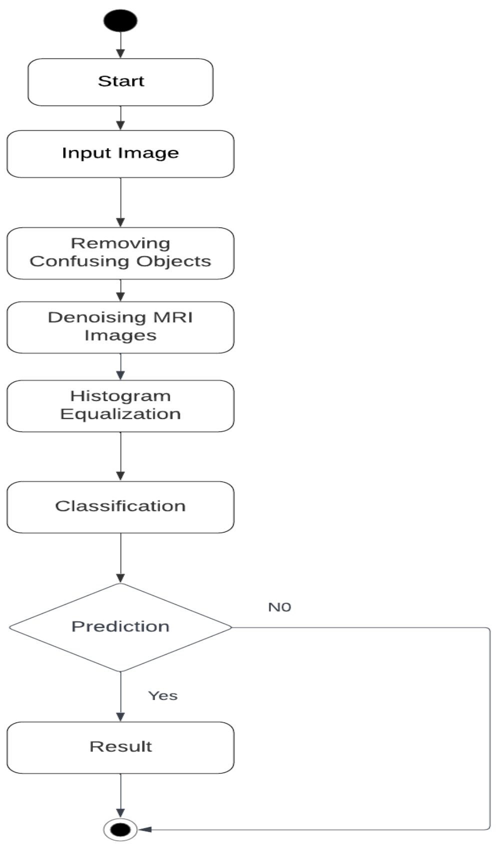


Figure 5.2.3.1 Activity Diagram

5.2.4 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one.

i. DFD Level 0

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows. Here the system is viewed as a single process, with its relationship to external entities.

Another name for it is context diagram. It is intended to be an abstraction view that presents the system as a lone process with its connections to outside entities. It depicts the complete system as a single bubble with incoming/outcoming arrows designating input and output data. In this case, the system is seen as a single process with connections to outside entities.

Here DFD 0 level is the basic model of our system and it only describe what is the model simply doing. It only shows the inputs and outputs of the model and how it flow in simple form. To inputs the MRI images to brain tumor detection system at this level. The system makes certain adjustments to produce better results. The system then uses the input MRI image to conduct prediction before sending results.

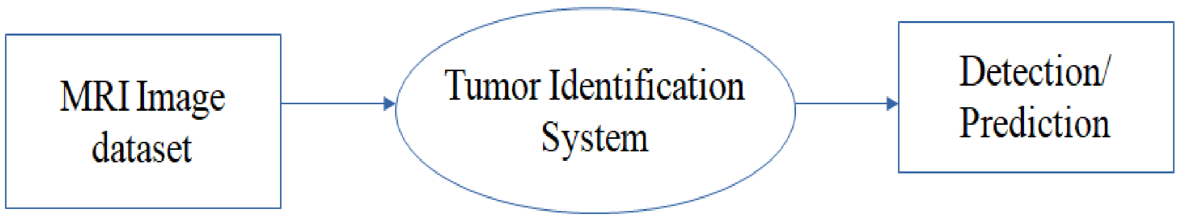


Figure 5.2.4.1 DFD Level 0

ii. DFD Level 1

The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information. It provides a more detailed view of the Context Level Diagram. Here, the main functions carried out by the system are highlighted as we break into its sub-processes.

The level 0 DFD is divided into a level DFD that is more detailed. The systems foundational modules and data flow between different modules are shown in level 1 DFD. Additionaly, level 1DFD includes fundamental procedure and information sources. It offers a more through perspective of the context level diagram. As we dissect the system's sub-processes, the primary tasks performed by the system are highlighted in this section

Here, perform image pre-processing on MRI images. Then perform feature extraction. The combined architecture(RNN+CNN) is used for classification. After that the model predicted result.

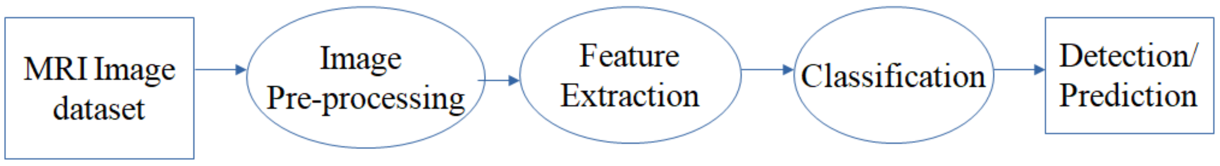


Figure 5.2.4.2 DFD Level 1

ii. DFD Level 2

A level 2 data flow diagram (DFD) offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system. In here chat module is decomposed into encryption, decryption processes and socket process. The encryption process encrypts the message to be communicated with other clients in the chat while the decryption process decrypts the incoming messages. AES based encryption is used in here. The socket process is responsible for sending and receiving the message real-time with the help of back end server.DFD level 1 offers more comperhensive perspective of the processes that make up an information system than a DFD level 0 does make-up of a system.

Here, the pre-processing contain three step process include removing confusing objects, denoising MRI images, histogram equalization. Then extract the features from MRI images based on shape, intensity and model. The combined architecture(RNN+CNN) used for classification. The features from MRI images pass to the trained model after loading at, and it will be predicted as no tumor, meningioma, pituitary, glioma.

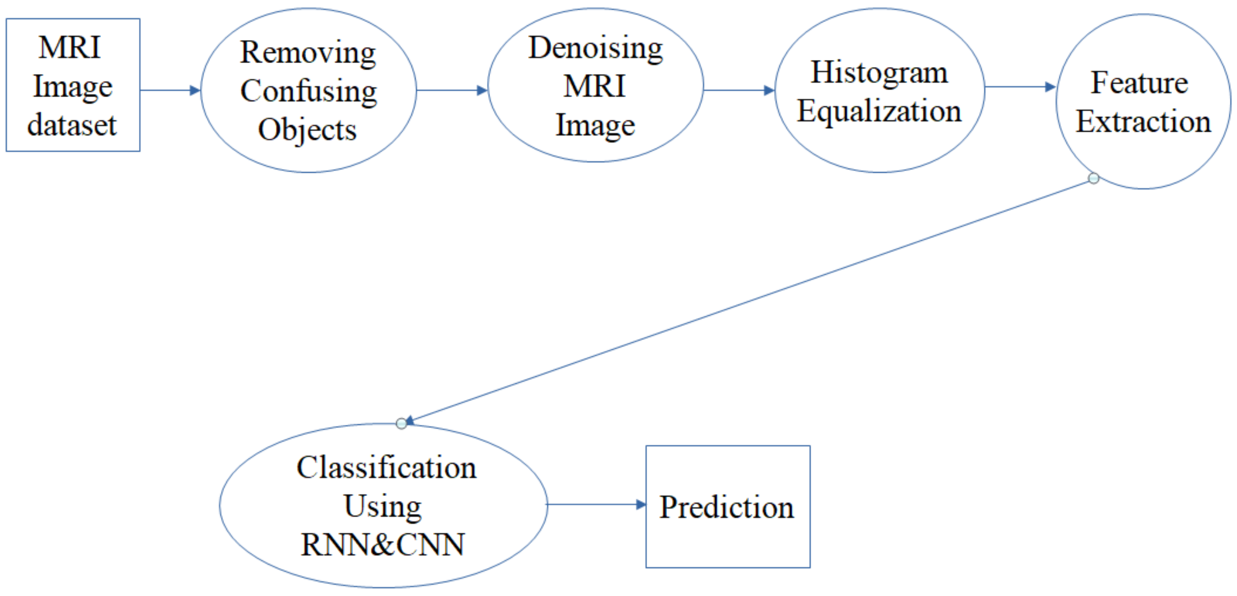


Figure 5.2.4.3 DFD Level 2

CHAPTER 6

TESTING

6.1 SOFTWARE TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

6.1.1 Software Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Unit Testing falls into white-box testing class. The requirement for unit test is that software element under test is isolated from the rest of the code and tested separately. Each unit test must be a standalone case independent from the rest of the tests. A good rule of thumb is that unit test should only work with objects that are in working memory of the machine that the test is run on. That means unit does not need network access, I/O operations or database access.

In this project, there are three modules likes image pre-processing, model creation(RNN+CNN) and prediction . Each module contains multiple functions and classes. After each module development, we have undergone multiple software unit testing to ensure the intended modules are working fine and in full efficiency. For example in tumor recognition have to check whether if the tumor is correctly classified.

6.1.2 Integration Testing

Integration Testing is the level of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies test defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Data can be lost across any interface. One module can have an adverse effect on another sub-function and when combined, may not produce the desired major function. Integration testing is a systematic testing for conducting tests to uncover errors associated within the interface. All the modules are combined and tested as a whole. Here the correction of the errors is difficult because of the large program size, so it is done in the next stage of testing.

6.1.3 System Integration Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing and as such, should require no knowledge of the inner design of the coder or logic. System testing takes as its input as integrated software system that has passed integration testing phase. System testing is usually considered appropriate for assessing the non-functional system requirements.

In this case, it is done to check whether all the modules are working as expected after integration. This testing helps us to tune the hardware and also the software to maximize the performance.

6.1.4 System Validation Testing

System Validation is a set of actions used to check the compliance of any element (a system element, a system, a document, a service, a task, a system requirement, etc.) with its purpose and functions. These actions are planned and carried out throughout the life cycle of the system. Validation is a transverse activity to every life cycle stage of the system. Particularly during the development cycle of the system, the validation process is performed in parallel with the system definition and system realization processes and applies to any activity and product resulting from this activity. It may be performed on an iterative basis on every produced engineering element during development and may begin with the validation of the expressed stakeholder requirements.

As a team have gone through each and every module, class, function to ensure its purpose like whether it is actually needed in this project or not or whether the module is serving its purpose or not. We have rewritten several modules, deleted several modules during the development period. This ensures the quality and stability of the project. For example, in the project, if took the Brain tumor detection system, it has undergone several validation testing to ensure the quality of the classification. Several times we have rewritten the codes to get quality out. This also helps in fine-tuning the module. The validation process is not limited to a phase at the end of system development, but generally occurs always at the end of each milestone. When the validation process is applied to the system when completely integrated, it is often called final validation.

CHAPTER 7

METHOD AND METHODOLOGY

A project can be brought to a successful end in various ways. But the best and most popular project management methodologies, methods, and frameworks are always changing. New concepts appear all the time. An entire string of methods, tools, and techniques lies behind all successful projects. However, project management methods, methodologies, and frameworks are not just for project managers. The entire project team must understand their usage, purpose, and basic terms. This will ensure that the whole process will go smoothly regardless of your choice. Remember that no project or team is the same. A methodology or framework that worked for someone else might not be the right one for you. That’s why it’s best to test how can use them for the own projects.

7.1 METHOD

Method defines as “a procedure or process for attaining an object: such as a systematic procedure, technique, or mode of inquiry employed by or proper to a particular discipline or a systematic plan followed in presenting material for instruction”. In other words, a method refers to a single action, tool, technique, process, or way of doing something.

Edge computing is a distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible. Data is the lifeblood of modern business, providing valuable business insight and supporting real-time control over critical business processes and operations. Today's businesses are awash in an ocean of data, and huge amounts of data can be routinely collected from sensors and IoT devices operating in real time from remote locations and inhospitable operating environments almost anywhere in the world. But this virtual flood of data is also changing the way businesses handle computing. The traditional computing paradigm built on a centralized data center and everyday internet isn't well suited to moving endlessly growing rivers of real-world data. Bandwidth limitations, latency issues and unpredictable network disruptions can all conspire to impair such efforts. Businesses are responding to these data challenges through the use of edge computing architecture. In simplest terms, edge computing moves some portion of storage and compute resources out of the central data center and closer to the source of the data itself. Rather than transmitting raw data to a central data center for processing and analysis, that work is instead performed where the data is actually generated -- whether that's a retail store, a factory floor, a sprawling utility or across a smart city. Only the result of that computing work at the edge, such as real-time business insights, equipment maintenance predictions or other actionable answers, is sent back to the main data centre for review and other human interactions. Thus, edge computing is reshaping IT and business computing. Take a comprehensive look at what edge computing is, how it works, the influence of the cloud, edge use cases, trade-offs and implementation considerations.

7.2 METHODOLOGY

A methodology is essentially a set of guiding principles and processes for managing a project. It is a collection of methods, practices, processes, techniques, procedures, and rules. In project management, methodologies are specific, strict, and usually contain a series of steps and activities for each phase of the project's life cycle. The choice of methodology defines how the work and communication is happened. They are defined approaches that show us exactly what steps to take next, the motivation behind each step, and how a project stage should be performed. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. A methodology offers a theoretical perspective for understanding which method, set of methods, or best practices can be applied.

7.2.1 Deep Learning

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, convolutional neural networks and transformers have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Artificial Neural Networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, artificial neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue. The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a non-polynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation that is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability.

Machine learning algorithms leverage structured, labelled data to make predictions—meaning that specific features are defined from the input data for the model and organized into tables. This doesn't necessarily mean that it doesn't use unstructured data; it just means that if it does, it generally goes through some pre-processing to organize it into a structured format.

Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let's say that we had a set of photos of different pets, and we wanted to categorize by "cat", "dog", "hamster", et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.

Machine learning and deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labelled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn't require labelled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics. Reinforcement learning is a process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward. Then, through the processes of gradient descent and back-propagation, the deep learning algorithm adjusts and fits itself for accuracy, allowing it to make predictions about a new photo of an animal with increased precision.

From another angle to view deep learning, deep learning refers to 'computer-simulate" or "automate human learning processes from a source (e.g., an image of dogs) to a learned object (dogs). Therefore, a notion coined as "deeper" learning or "deepest" learning makes sense. The deepest learning refers to the fully automatic learning from a source to a final learned object. A deeper learning thus refers to a mixed learning process: a human learning process from a source to a learned semi-object, followed by a computer learning process from the human learned semi-object to a final learned object.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

Another process called back-propagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and back-propagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces. Models are trained by using a large set of labelled data and neural network architectures that contain many layers .Real-world deep learning applications are a part of our daily lives, but in most cases, they are so well-integrated into products and services that users are unaware of the complex data processing that is taking place in the background.

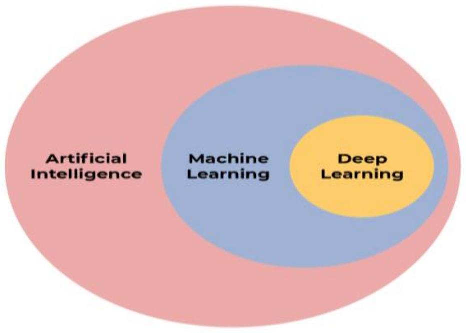


Figure 7.2.1.1 Deep learning, a subset of machine learning

In Brain tumor detection we use the concept of machine learning and deep learning to solve various problems. Both tumor recognition and it's classification uses deep learning based image classification algorithm.

7.2.2 CNN

A Convolutional Neural Network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. CNN is always known for taking a greater number of highlights from given raw RGB picture and it is one of the best options for image processing. Using CNN, the input image data is processed by extracting both simple and the complex information from the image thereby helping the system to work efficiently since image processing is the major part in the system.

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 the reusability of weights. In other words, the network can be trained to understand the sophistication of the image better. The ability of artificial intelligence to close the gap between human and computer skills has been growing dramatically. Both professionals and amateurs focus on many facets of the field to achieve great results. The field of computer vision is one of several such disciplines.

The goal of this field is to give robots the ability to see the environment similarly as humans do and to use that understanding for a variety of activities, including image and video recognition, image analysis, media recreation, recommendation systems, natural language processing, etc. With time, one particular algorithm a Convolutional Neural Network has been developed and optimised, primarily leading to breakthroughs in computer vision with deep learning.

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network that has a wholesome understanding of images in the dataset, similar to how we would. There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in the case of the former, or Same Padding in the case of the latter.

Convolutional Neural Network (CNN, or ConvNet) is a class of deep neural network, most commonly applied to analyse visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. The innovation of convolutional neural networks is the ability to automatically learn a large number of filters in parallel specific to a training dataset under the constraints of a specific predictive modelling problem, such as image classification. The result is highly specific features that can be detected anywhere on input images.

They are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. Classification of images with different positions, co-ordinate frame are the disadvantages of CNN The following figure shows the architecture of CNN.

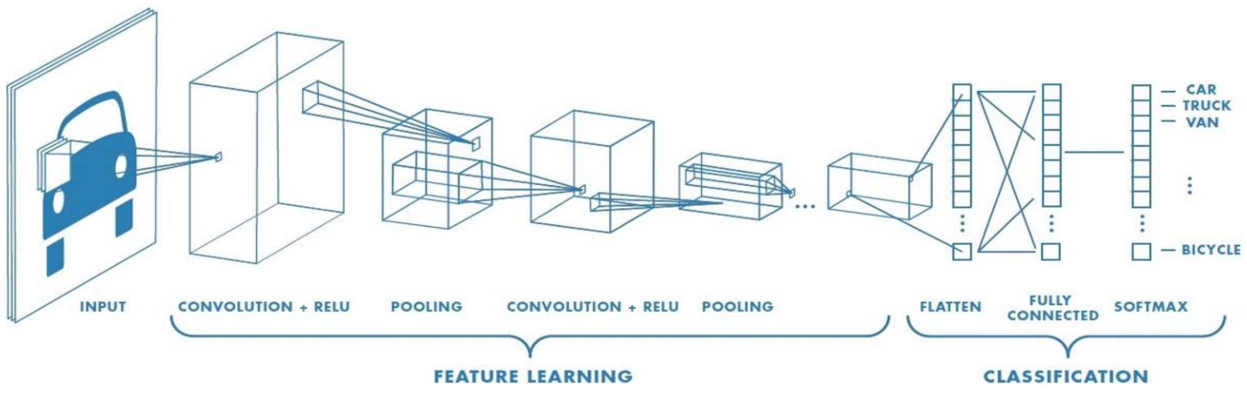


Figure 7.2.2.1 Architecture of CNN

7.2.3 RNN

A Recurrent Neural Network (RNN) is a class of artificial neural networks where connections between nodes can create a cycle, allowing output from some nodes to affect subsequent input to the same nodes. This allows it to exhibit temporal dynamic behaviour.

Derived from feed-forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as un-segmented, connected handwriting recognition or speech recognition. Recurrent neural networks are theoretically Turing complete and can run arbitrary programs to process arbitrary sequences of inputs.

A Neural Network consists of different layers connected to each other, working on the structure and function of a human brain. It learns from huge volumes of data and uses complex algorithms to train a neural net. Here is an example of how neural networks can identify a dog's breed based on their features.

- The image pixels of two different breeds of dogs are fed to the input layer of the neural network.
- The image pixels are then processed in the hidden layers for feature extraction.
- The output layer produces the result to identify if it's a German Shepherd or a Labrador.
- Such networks do not require memorizing the past output.

RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer.RNN were created because there were a few issues in the feed-forward neural network.

- Cannot handle sequential data.
- Consider only the current input.
- Cannot memorize previous inputs.

The solution to these issues is the RNN. An RNN can handle sequential data, accepting the current input data, and previously received input inputs. RNNs can memorize previous inputs due to their internal memory. he Recurrent Neural Network will standardize the different activation functions and weights and biases so that each hidden layer has the same parameters. Then, instead of creating multiple hidden layers, it will create one and loop over it as many times as required.

A feed-forward neural network allows information to flow only in the forward direction, from the input nodes, through the hidden layers, and to the output nodes. There are no cycles or loops in the network. In a feed-forward neural network, the decisions are based on the current input. It doesn't memorize the past data, and there's no future scope. Feed-forward neural networks are used in general regression and classification problems. RNNs are used to caption an image by analyzing the activities present. Any time series problem, like predicting the prices of stocks in a particular month, can be solved using an RNN, Text mining and Sentiment analysis can be carried out using an RNN for Natural Language Processing (NLP).Given an input in one language, RNNs can be used to translate the input into different languages as output. RNNs are designed to handle input sequences of variable length, which makes them well-suited for tasks such as speech recognition, natural language processing, and time series analysis. RNNs have a memory of past inputs, which allows them to capture information about the context of the input sequence. This makes them useful for tasks such as language modelling, where the meaning of a word depends on the context in which it appears.

RNNs share the same set of parameters across all time steps, which reduces the number of parameters that need to be learned and can lead to better generalization. RNNs use non-linear activation functions, which allows them to learn complex, non-linear mappings between inputs and outputs. RNNs process input sequences sequentially, which makes them computationally efficient and easy to parallelize. RNNs can be adapted to a wide range of tasks and input types, including text, speech, and image sequences. RNNs have been shown to achieve state-of-the-art performance on a variety of sequence modelling tasks, including language modelling, speech recognition, and machine translation. These advantages make RNNs a powerful tool for sequence modelling and analysis, and have led to their widespread use in a variety of applications, including natural language processing, speech recognition, and time series analysis.

RNNs can suffer from the problem of vanishing or exploding gradients, which can make it difficult to train the network effectively. This occurs when the gradients of the loss function with respect to the parameters become very small or very large as they propagate through time. RNNs can be computationally expensive to train, especially when dealing with long sequences. This is because the network has to process each input in sequence, which can be slow. Although RNNs are designed to capture information about past inputs, they can struggle to capture long-term dependencies in the input sequence. This is because the gradients can become very small as they propagate through time, which can cause the network to forget important information. RNNs are inherently sequential, which makes it difficult to parallelize the computation. This can limit the speed and scalability of the network. There are many different variants of RNNs, each with its own advantages and disadvantages. Choosing the right architecture for a given task can be challenging, and may require extensive experimentation and tuning. The output of an RNN can be difficult to interpret, especially when dealing with complex inputs such as natural language or audio. This can make it difficult to understand how the network is making its predictions. These disadvantages are important when deciding whether to use an RNN for a given task. However, many of these issues can be addressed through careful design and training of the network and through techniques such as regularization and attention mechanisms.

The term "recurrent neural network" is used to refer to the class of networks with an infinite impulse response, whereas "Convolutional Neural Network" refers to the class of finite impulse response. Both classes of networks exhibit temporal dynamic behaviour. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feed forward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled.

Both finite impulse and infinite impulse recurrent networks can have additional stored states, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units. This is also called Feedback Neural Network (FNN).

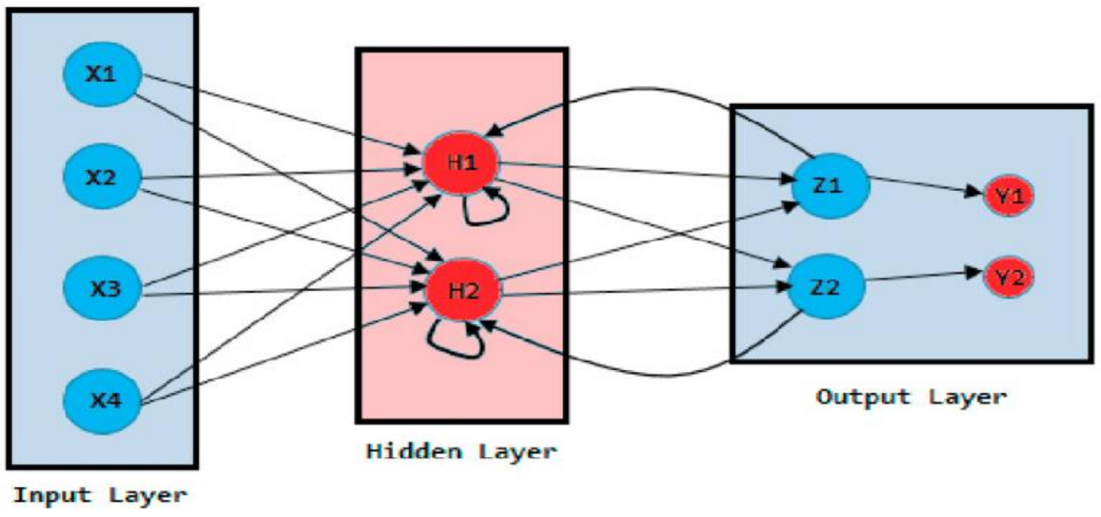


Figure 7.2.3.1 Architecture of RNN

7.2.4 Histogram Equalization

Histogram is a graphical representation of the intensity distribution of an image. In simple terms, it represents the number of pixels for each intensity value considered. Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the image is represented by a narrow range of intensity values. Through this adjustment, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the highly populated intensity values which are used to degrade image contrast.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are either over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique adaptive to the input image and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

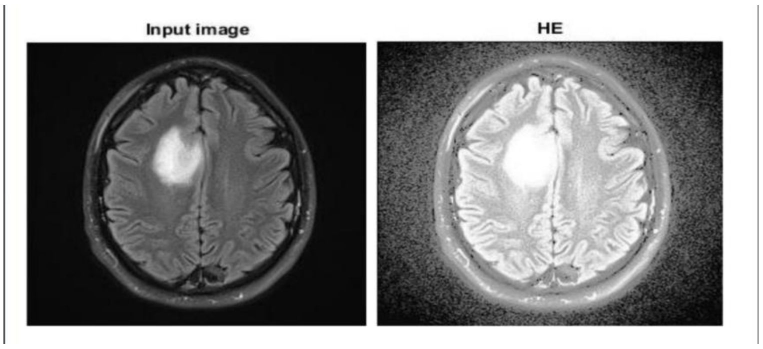


Figure 7.2.4.1 Histogram Equalization

7.2.5 NLM

Non-local means is an algorithm in image processing for image denoising. Unlike "local mean" filters, which take the mean value of a group of pixels surrounding a target pixel to smooth the image, non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared with local mean algorithms.

If compared with other well-known denoising techniques, non-local means adds "method noise" (i.e. error in the denoising process) which looks more like white noise, which is desirable because it is typically less disturbing in the denoised product. Recently non-local means has been extended to other image processing applications such as de-interlacing, view interpolation, and depth maps regularization. Medical images often consist of low-contrast objects corrupted by random noise arising in the image acquisition process. Thus, image denoising is one of the fundamental tasks required by medical imaging analysis.

Nonlocal means (NL-means) method provides a powerful framework for denoising. In this work, we investigate an adaptive denoising scheme based on the patch NL-means algorithm for medical imaging denoising. In contrast with the traditional NL-means algorithm, the proposed adaptive NL-means denoising scheme has three unique features. First, we use a restricted local neighbourhood where the true intensity for each noisy pixel is estimated from a set of selected neighbouring pixels to perform the denoising process. Second, the weights used are calculated thanks to the similarity between the patch to denoise and the other patches candidates. Finally, we apply the steering kernel to preserve the details of the images. The proposed method has been compared with similar state-of-art methods over synthetic and real clinical medical images showing an improved performance in all cases analyzed.

CHAPTER 8

RESULTS AND DISCUSSION

The experimental results of the proposed models are discussed here. The deep learning models are built using python with the help of Keras and Tensorflow. The models are trained over 30 epochs.

8.1 PERFORMANCE ANALYSIS

The performance of various state of the art models for both brain tumor recognition and classification is discussed in here. The following figures show the accuracy and loss of the proposed model. We have 70% of training set and 30% of validation set among dataset. We train the model using the training data and check its performance on both the training and validation sets (evaluation metric is accuracy).

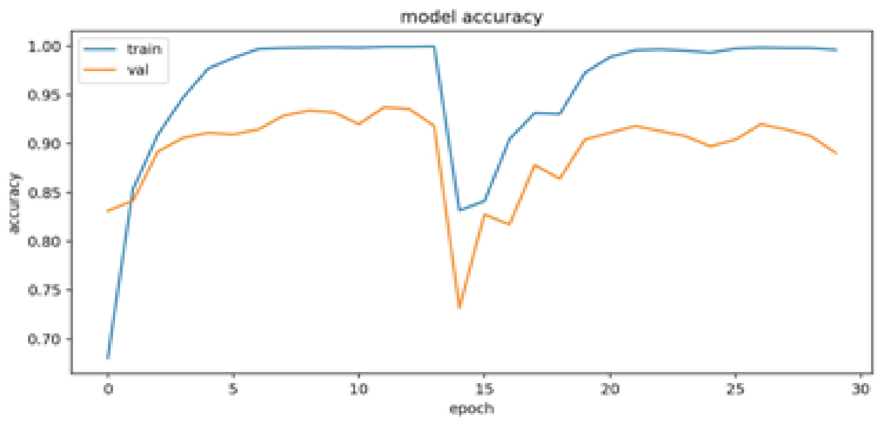


Figure 8.1.1 Training and validation accuracy

As you can see in the diagram, the accuracy increases rapidly in the first five epochs, indicating that the network is learning fast. Generally, if the training data accuracy keeps improving while the validation data accuracy gets worse, if data are encountering overfitting. It indicates that the model is starting to memorize the data. At training, we get 99%, and at validation, we get 93%.

The training loss is a metric used to assess how a deep learning model fits the training data. That is to say, it assesses the error of the model on the training set. Note that, the training set is a portion of a dataset used to initially train the model. Computationally, the training loss is calculated by taking the sum of errors for each example in the training set. validation loss is a metric used to assess the performance of a deep learning model on the validation set. The validation set is a portion of the dataset set aside to validate the performance of the model. The validation loss is similar to the training loss and is calculated from a sum of the errors for each example in the validation set.

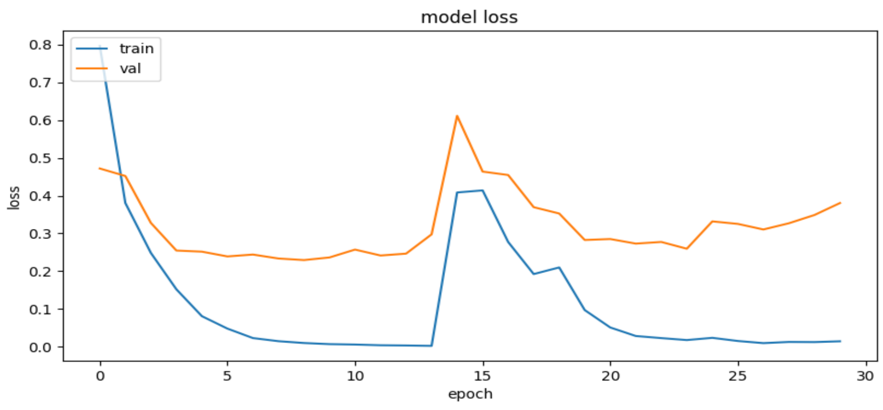
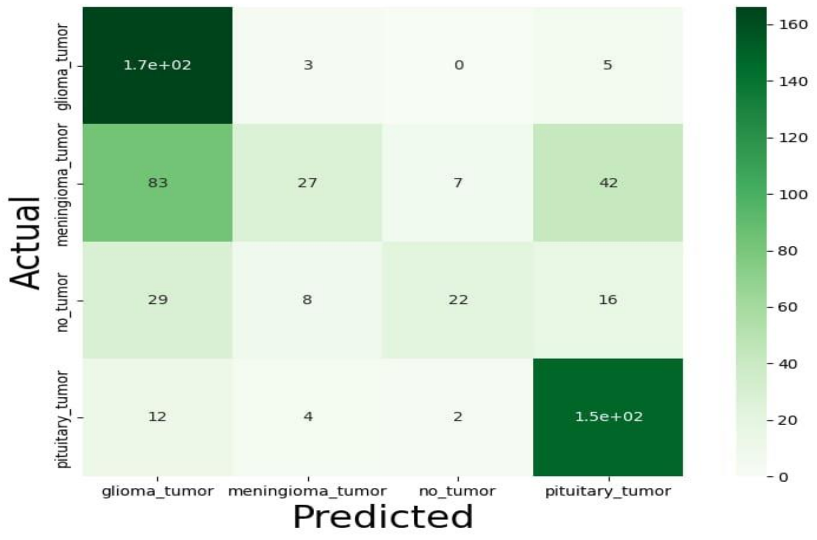


Figure 8.1.2 Training and validation loss

As you can see in the diagram, the loss on the training set decreases rapidly for the first five epochs. For the test set, the loss does not decrease at the same rate as the training set, but remains almost flat for multiple epochs. This means our model is generalizing well to unseen data.



Actual \ Predicted	glioma_tumor	meningioma_tumor	no_tumor	pituitary_tumor
glioma_tumor	1.7e+02	3	0	5
meningioma_tumor	83	27	7	42
no_tumor	29	8	22	16
pituitary_tumor	12	4	2	1.5e+02

Figure 8.1.3 Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm. A multi-class confusion matrix here. There are four classes such as pituitary_tumor, no_tumor, meningioma_tumor, gloioma_tumor. The explored models have two main parts in their structures,the convolutional part and the classifier part. The convolutional part extracts the inputted image’s features and the classifier part classifies these features into one of the intended classes. Since "Brain Tumor" has four classes, we must modify the classifier component of these models. Max-pooling layers help speed up diagnosis of MRI images since they reduce the output size of the convolutional layer.

CHAPTER 9

FUTURE SCOPE

Over the last decade, from various research results it is being observed that it is very time consuming, but it will get faster if use combination of Recurrent Neural Network and Radial Basis Function Networks in our system. In future, combine two or more segmentation methods to improve the feature extraction for better results. This proposed system has robustly intended to widen this research to trial through huge dataset.

And also we are planning the identification of glioma tumor grades. Currently the identification of tumor stages is difficult. There is a possibility of further development for the prediction of tumor stages.

CHAPTER 10

CONCLUSION

A brain tumour is currently thought to be one of the most susceptible life-threatening illnesses. The inner portion of the human brain, a brain tumor, is surrounded by abnormal cells that have gathered into a cluster. In order to diagnose the tumour accurately and effectively, the patients are required for segmenting and finding the tumour. In order to detect glioma, meningioma, and pituitary brain diseases, a deep convolution neural network architecture is suggested with the goal of achieving high classification accuracy quickly.

first, a suitable brain tumour dataset for carrying out the training and testing procedure quickly. Second, a three-step pre-processing method was used to clear the MRI images of any distracting factors, denoise them, and improve their contrast. All of the investigated models were favourably and significantly impacted by this method. Third, as part of a training plan, we train our model from scratch on the desired patterns. Fourth, we employed our model to quickly and accurately classify the MRI images based on their features. We test the suggested model using a dataset containing 2870 MRI pictures. The precision of the suggested model was 99%. In actual use, the suggested model can be viewed as an automated computer-aided detector instrument to accurately and promptly identify brain abnormalities in MRI images.

REFERENCES

[1]. R. L. Siegel, K. D. Miller, and A. Jemal, ''Cancer statistics,'' Cancer J. Clin., vol. 65, no. 1, pp. 5–29, 2015.

[2]. R. Siegel, C. R. Miller, and A. Jamal, ''Cancer statistics,'' Cancer J. Clin., vol. 67, no. 1, pp. 7–30, 2017.

[3]. Brain Tumor Statistics, American Brain Tumor Association. Accessed: Oct. 26, 2019. [Online]. Available: http://abta.pub30.convio.net/

[4]. M. R. Ismael and I. Abdel-Qader, ''Brain tumor classification via statistical features and vii back-propagation neural network,'' in Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT),May 2018, pp. 0252–0257.

[5]. K. V. A. Muneer, V. R. Rajendran, and J. K. Paul, ''Glioma tumor grade identification using artificial intelligent techniques,'' J. Med. Syst., vol. 43, no. 5, p. 113, Mar. 2019.

[6]. P. Afshar, A. Mohammadi, and K. N. Plataniotis, ''Brain tumor type classification via capsule networks,'' in Proc. 25th IEEE Int. Conf. Image Process. (ICIP), Oct. 2018, pp. 3129–3133, doi: 10.1109/ICIP.2018.8451379.

[7]. T. Gopi Krishna a*, Satyasis Mishra b, Sunita Satapathy c, K. V. N. Sunitha d and Mohamed A. Abdelhadi e,"Classification of Brain Tumor Types on Magnetic Resonance Images Using Hybrid Deep Learning Approach with Radial Basis Function Neural Network",DOI:10.9734/bpi/ramrcs/v7/1579B.

[8]. Jalluri Gnana SivaSai, P. Naga Srinivasu, Munjila Naga Sindhuri,Kola Rohitha, and Sreesailam Deepika,"An Automated Segmentation of Brain MR Image Through Fuzzy Recurrent Neural Network", https://doi.org/10.1007/978-981-15-5495-7_9.

[9]. Dasgupta, Archya et al. "Indian data on central nervous tumors: A summary of published work." South Asian journal of cancer vol. 5,3 (2016): 147-53.doi:10.4103/2278-330X.187589.

[10]. G. Vishnuvarthanan, M. P. Rajasekaran, P. Subbaraj, and A. Vishnu varthanan, ''An unsupervised learning method with a clustering approach for tumor identification tissue segmentation in magnetic resonance brain images,'' Appl. Soft Comput., vol. 38, pp. 190–212,Jan. 2016, doi: 10.1016/j.asoc.2015.09.016.

[11]. N. Dhanachandra, K. Manglem, and Y. J. Chanu, ''Image segmen-tation using K-means clustering algorithm and subtractive clusteringalgorithm,'' Proc. Comput. Sci., vol. 54, pp.764–771, Jan. 2015, doi:10.1016/j.procs.2015.06.090.

[12]. J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja,''Segmentation,

[13]. feature extraction, and multiclass brain tumor classification,'' J. Digit. Imag., vol. 26, no. 6, pp.1141–1150, Dec. 2013, doi:10.1007/s10278-013-9600-0.

[14]. H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, ''Automatic brain tumor detection and segmentation using U-Net based fully convolutional net-works,'' in Proc. Annu. Conf. Med. Image Understand. Anal., 2017,pp. 506–517, doi: 10.1007/978-3-319-60964-5_44.

[15]. M. O. Khairandish, M. Sharma, V. Jain, J. M. Chatterjee, and N. Z. Jhanjhi,''A hybrid CNN-SVM threshold segmentation approach for tumor detection and classification of MRI brain images,'' IRBM, Jun. 2021, doi:10.1016/j.irbm.2021.06.003.

[16]. A. A. Funmilola, O. A. Oke, T. O. Adedeji, O. M. Alade, and E. A. Adewusi, ''Fuzzy kc means clustering algorithm for medical image segmentation,'' J. Inf. Eng. Appl., vol. 2, no.6, pp. 21–32, 2012.

[17]. C. Zhang, X. Shen, H. Cheng, and Q. Qian, ''Brain tumor segmentation based on hybrid clustering and morphological operations,'' Int. J. Biomed. Imag., vol. 2019, pp. 1–11, Apr. 2019.

[18]. R. Ahmmed and M. F. Hossain, ''Tumor detection in brain MRI image using template based K-means and fuzzy C-means clustering algorithm,''in Proc. Int. Conf. Comput. Commun. Informat. (ICCCI), Jan. 2016,pp. 1–6.

[19]. A. Bal, M. Banerjee, P. Sharma, and M. Maitra, ''Brain tumor segmenta-tion on MR image using K-Means and fuzzy-possibilistic clustering,'' inProc. 2nd Int. Conf. Electron., Mater. Eng. Nano-Technol. (IEMENTech),May 2018, pp. 1–8.

[20]. A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, ''A deep learning-based framework for automatic brain tumors classification using transfer learning,''Circuits, Syst., Signal Process., vol. 39, no. 2,pp. 757–775, Feb. 2020.

[21]. S. Deepak and P. M. Ameer, ''Brain tumor classification using deep CNN features via transfer learning,'' Comput. Biol. Med., vol. 111, Aug. 2019,Art. no. 103345.

[22]. N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, andT. R. Mengko, ''Brain tumor classification using convolutional neural net-work,'' in World Congress on Medical Physics and Biomedical Engineer-ing. Singapore: Springer, 2018, pp. 183–189.

[23]. A. Pashaei, H. Sajedi, and N. Jazayeri, ''Brain tumor classification via convolutional neural network and extreme learning machines,'' in Proc.8th Int. Conf. Comput. Knowl. Eng. (ICCKE), Oct. 2018, pp. 314–319.

[24]. H. H. Sultan, N. M. Salem, and W. Al-Atabany, ''Multi-classification of brain tumor mages using deep neural network,'' IEEE Access, vol. 7,pp. 69215–69225, 2019.

APPENDIX – 1

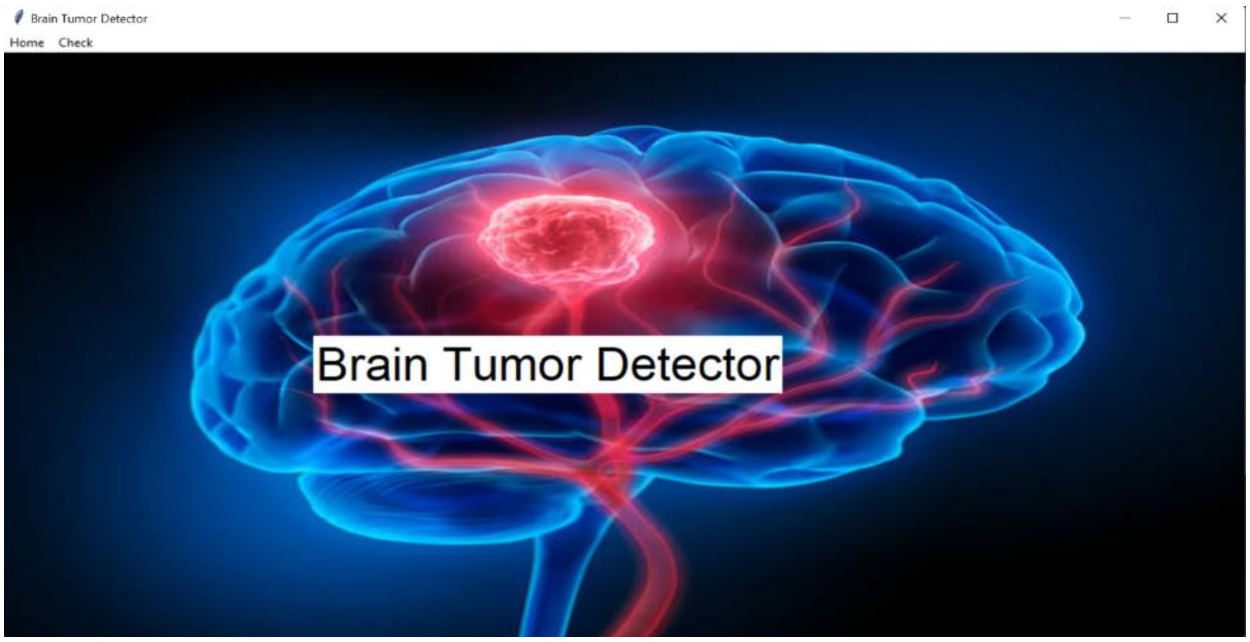


Figure 1: Home Page

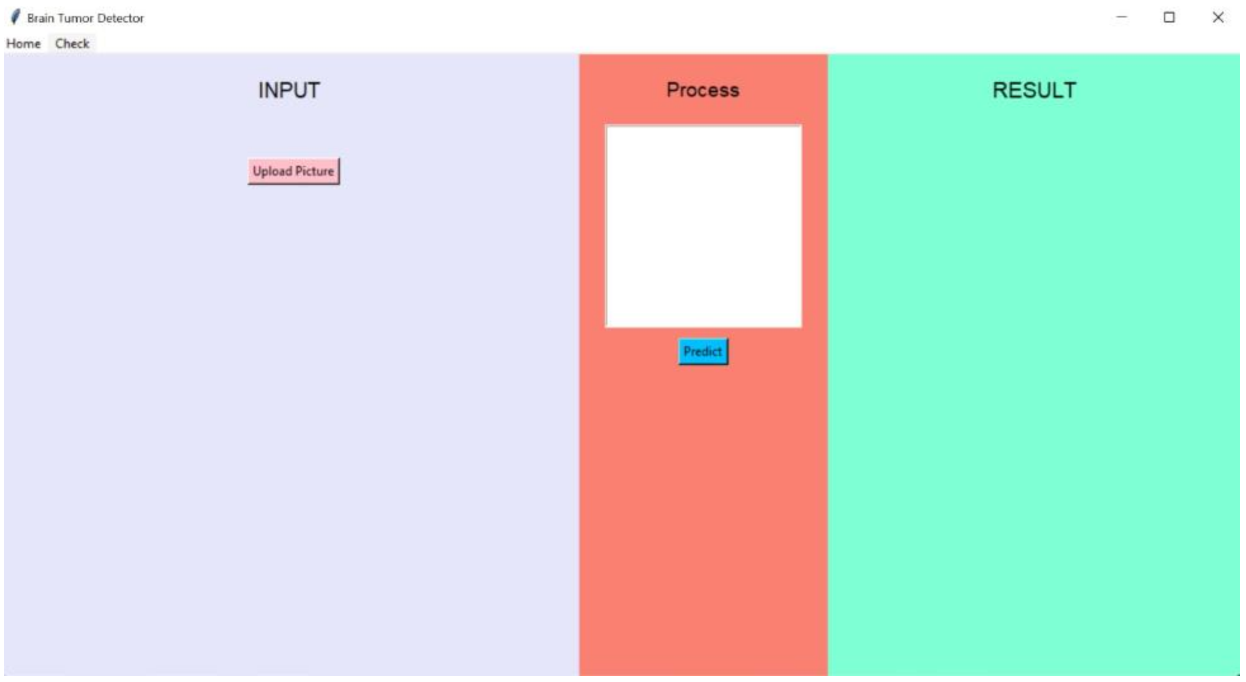


Figure 2:Selecting image using Check button

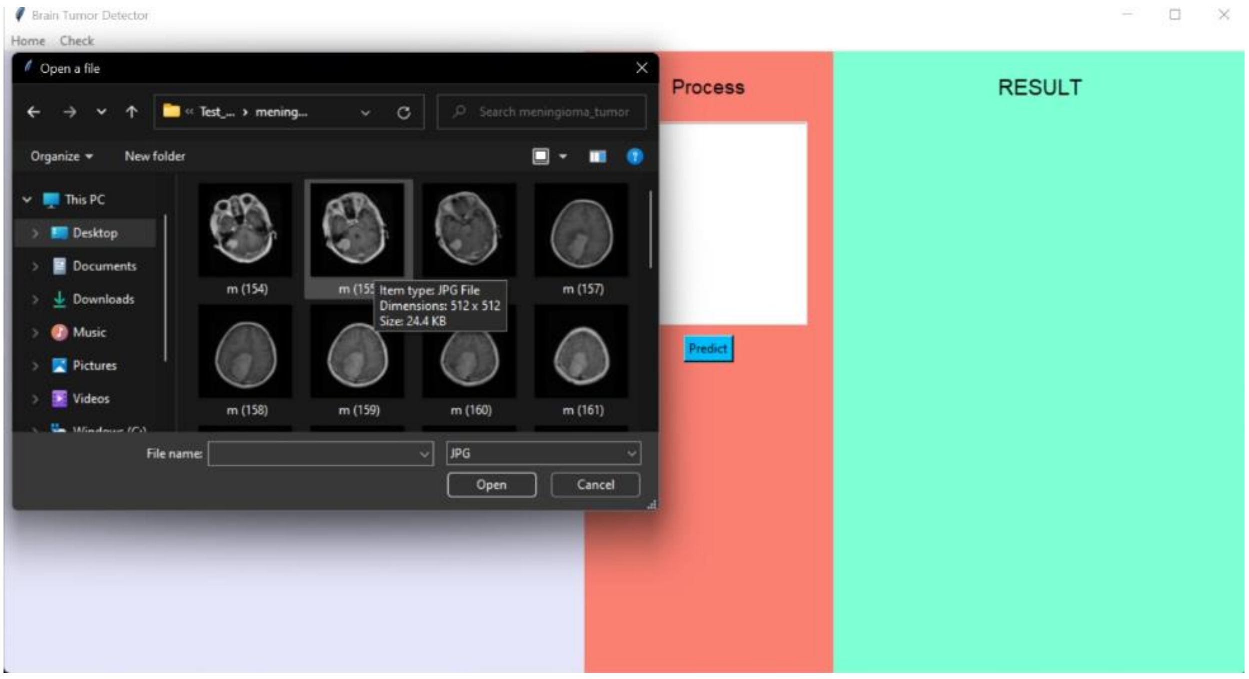


Figure 3 : Selecting Image

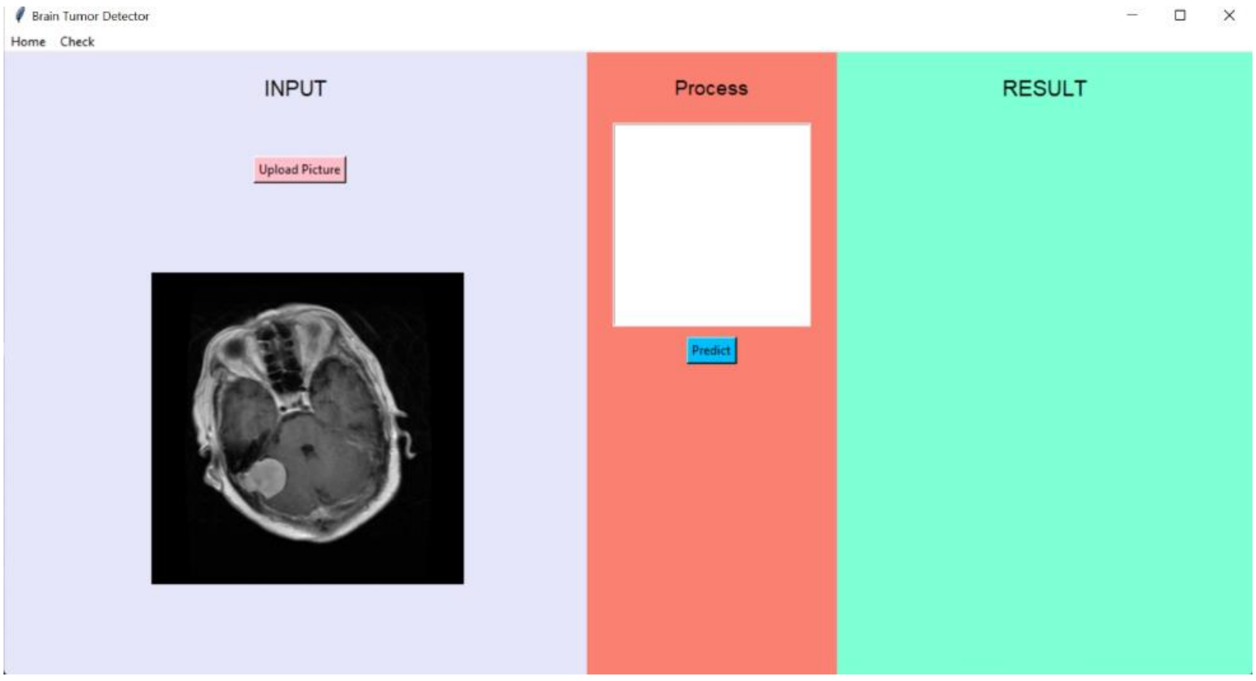


Figure 4: Image Uploading

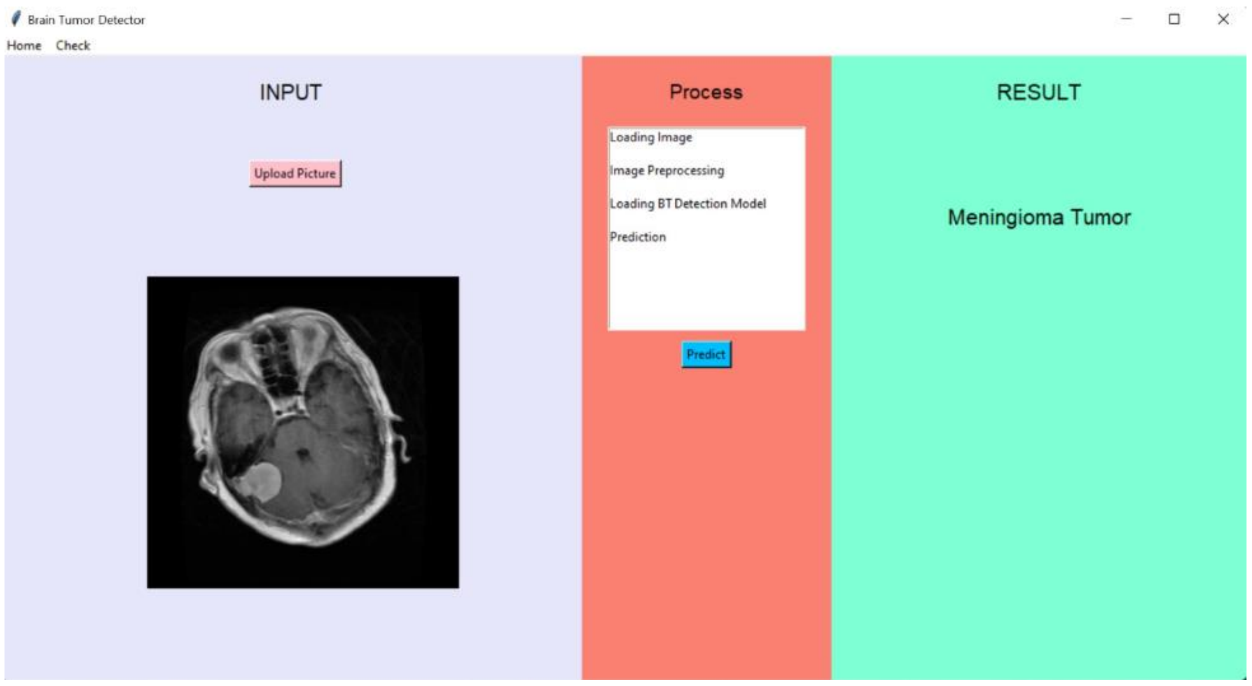


Figure 5: Meningioma tumor

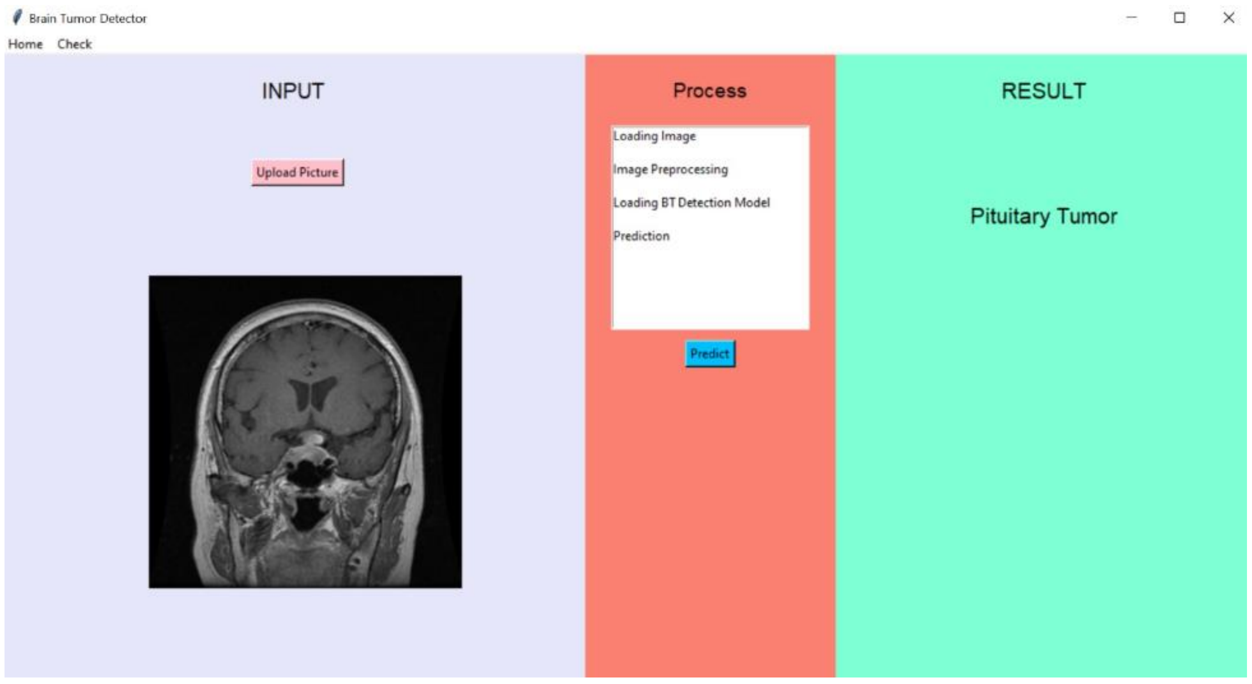
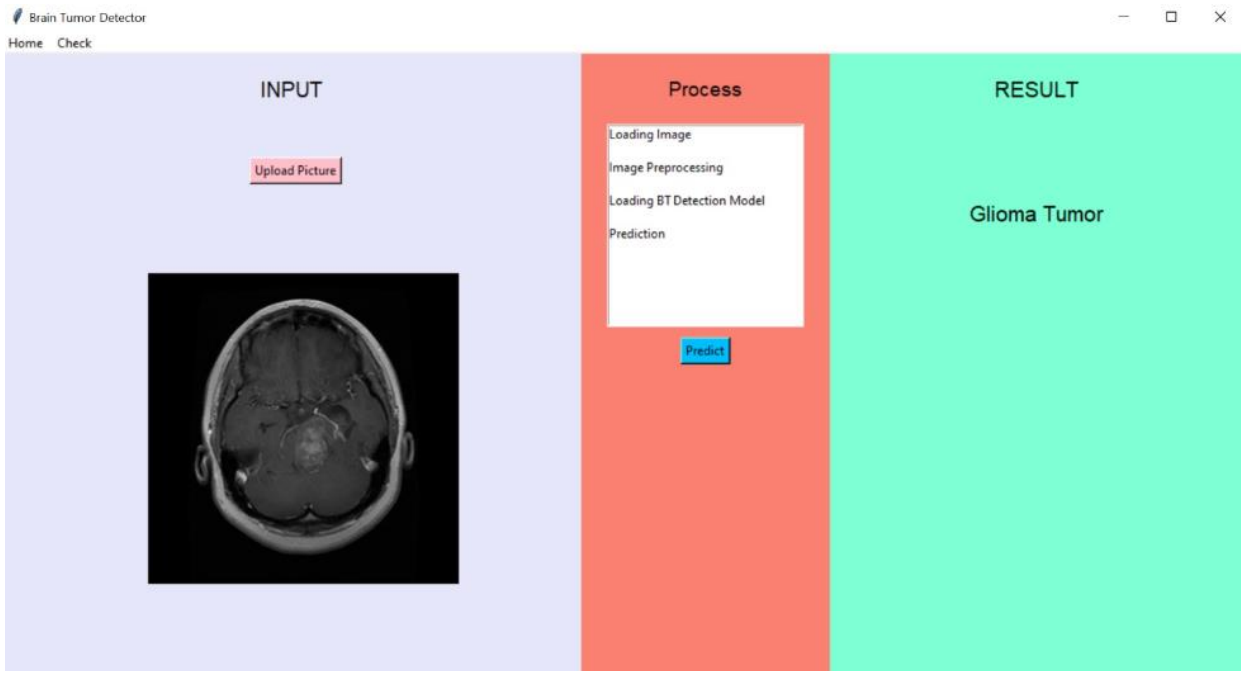


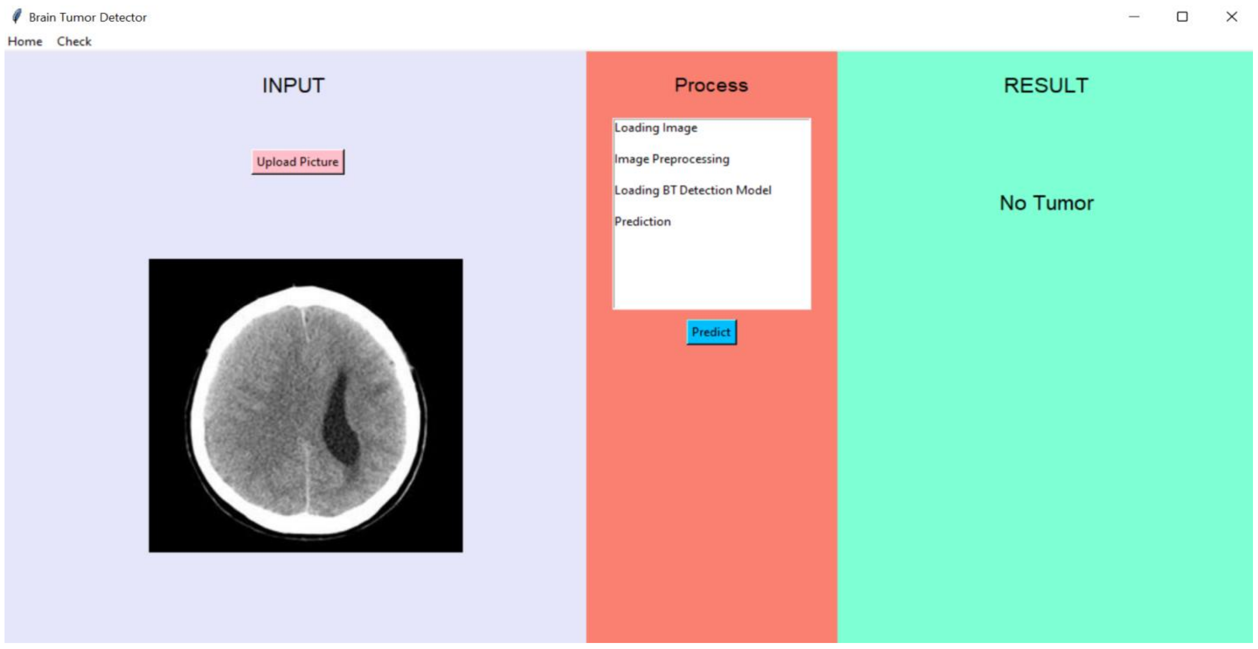
Figure 6: pituitary tumor



Loading Image
Image Preprocessing
Loading BT Detection Model
Prediction

Glioma Tumor

Figure 7: Glioma tumor



Loading Image
Image Preprocessing
Loading BT Detection Model
Prediction

No Tumor

Figure 8: no tumor

APPENDIX - 2

MODEL TRAINING

```
#importing necessary libraries

import pandas as pd

import cv2

from tensorflow.keras.callbacks import ModelCheckpoint

from tensorflow.keras.layers import BatchNormalization

from tensorflow.keras.optimizers import Adam

from sklearn.metrics import confusion_matrix

import seaborn as sns

from matplotlib import pyplot as plt

from tensorflow.keras.optimizers import SGD

import itertools

import tensorflow as tf

import numpy as np

import os

import pickle

from keras.models import Sequential, load_model

#define dataset path

data_path="Project_Dataset/Training"

my_list=os.listdir(data_path)

print(my_list)
print(len(my_list))

#perform denoising

def denoise(image):

    #denoising using Non-local mean algorithm

    out = cv2.fastNlMeansDenoisingColored(image,None,10,10,7,21)

    return out

#initialize variables

data=[]

labels=[]

img_size=150

for i in my_list:

    image_list=os.listdir(data_path+"/"+i)

    for j in image_list:

        # try:

        path=data_path+"/"+i+"/"+j

        #read image

        img=cv2.imread(path)

        #perform image resizing

        img=cv2.resize(img,(img_size,img_size))

        #denoising

        output=denoise(img)

        img_yuv = cv2.cvtColor(output,cv2.COLOR_BGR2YUV)
        # apply histogram equalization
        img_yuv[:,:,0] = cv2.equalizeHist(img_yuv[:,:,0])
        hist_eq = cv2.cvtColor(img_yuv, cv2.COLOR_YUV2BGR)
        data.append(hist_eq)
        print("i : ",i)
        if i=="glioma_tumor":
            labels.append(0)
        elif i=="meningioma_tumor":
            labels.append(1)
        elif i=="no_tumor":
            labels.append(2)
        else:
            labels.append(3)
print(len(data))
print(len(labels))
# #saving as pickle file
# pickle.dump(data,open('data.pkl','wb'))
# pickle.dump(labels,open('labels.pkl','wb'))
##########################
#loading pickle files
data=pickle.load(open('data.pkl','rb'))
labels=pickle.load(open('labels.pkl','rb'))
print("******")
print(len(data))

print(len(labels))

data=np.array(data)

labels=np.array(labels)

print(data.shape)

print(labels.shape)

from sklearn.model_selection import train_test_split

#perform train-test splitting

x_train, x_test, y_train, y_test = train_test_split(data, labels, random_state=0, shuffle=True,test_size=0.2)

print("\nTraining Set")

print(x_train.shape)

print(y_train.shape)

print("\nTesting Set")

print(x_test.shape)

print(y_test.shape)

x_train = np.expand_dims(x_train, axis=1)

x_test = np.expand_dims(x_test, axis=1)

print("x_train shape after : ",x_train.shape)

print("x_test shape after : ",x_test.shape)

x_train=x_train/255

x_test=x_test/255
```

```
#perform Label encoding (return binary)

from sklearn.preprocessing import LabelBinarizer

#initialize

label_as_binary = LabelBinarizer()

y_train = label_as_binary.fit_transform(y_train)

y_test = label_as_binary.fit_transform(y_test)

from model import CNN_RNN

#function call

model=CNN_RNN()

#compiling the mode;

model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["acc"])

#saving the model

checkpoint=ModelCheckpoint("Project_Saved_Models/brain_tumor_model.h5",

                    monitor="acc",

                    save_best_only=True,

                    verbose=1)

#training

history=
model.fit(x_train,y_train,epochs=30,batch_size=4,validation_data=(x_test,y_test),callbacks=[che
ckpoint])

#plotting

plt.plot(history.history['acc'])

plt.plot(history.history['val_acc'])
plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'val'], loc='upper left')

plt.savefig("Project_Extra/acc_final.png")

plt.show()

plt.plot(history.history['loss'])

plt.plot(history.history['val_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'val'], loc='upper left')

plt.savefig("Project_Extra/loss_final.png")

plt.show()

y_pred = model.predict(x_test)

y_pred = np.argmax(y_pred, axis=1)

y_true = np.argmax(y_test, axis=1)

disease_types=['glioma_tumor', 'meningioma_tumor','no_tumor','pituitary_tumor']

cm = confusion_matrix(y_true, y_pred)

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

ax = sns.heatmap(cm, cmap=plt.cm.Greens, annot=True, square=True, xticklabels=disease_types, yticklabels=disease_types)

ax.set_ylabel('Actual', fontsize=24)
ax.set_xlabel('Predicted', fontsize=24)

plt.show()
```

PREDICTION

```
import numpy as np

import cv2

from PIL import Image

from tensorflow.keras.models import load_model

#load trained model

loaded_model = load_model("Project_Saved_Models/brain_tumor_model.h5")

#perform denoising

def denoise(image):

    #denoising using Non-local mean algorithm

    out = cv2.fastNlMeansDenoisingColored(image,None,10,10,7,21)

    return out

def main1(path):

    image = cv2.imread(path)

    image_resize=cv2.resize(image,(150,150))

    output=denoise(image_resize)

    img_yuv = cv2.cvtColor(output,cv2.COLOR_BGR2YUV)

    # apply histogram equalization

    img_yuv[:,:,0] = cv2.equalizeHist(img_yuv[:,:,0])

    hist_eq = cv2.cvtColor(img_yuv, cv2.COLOR_YUV2BGR)

    print(hist_eq.shape)
    out_image=np.expand_dims(hist_eq,axis=0)

    out_image=np.expand_dims(out_image,axis=0)/255

    print(out_image.shape)

    my_pred = loaded_model.predict(out_image)

    print(my_pred)

    my_pred=np.argmax(my_pred,axis=1)

    print(my_pred)

    if my_pred==0:

        print("Glioma Tumor")

    elif my_pred==1:

        print("Meningioma Tumor")

    elif my_pred==2:

        print("No Tumor")

    elif my_pred==3:

        print("Pituitary Tumor")

if _name=="main_":

    from tkinter.filedialog import askopenfilename

    path=askopenfilename()

    main1(path)
```

MODEL CREATION

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout, Flatten

from tensorflow.keras.layers import Conv2D
from tensorflow.keras.applications import VGG16

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import BatchNormalization

from keras.models import Sequential, load_model

from keras.layers import Dropout, Dense, GlobalMaxPooling2D,GlobalAveragePooling2D,Flatten

from tensorflow.keras.callbacks import ModelCheckpoint

from tensorflow.keras.layers import BatchNormalization

from tensorflow.keras.layers import TimeDistributed, Conv2D, MaxPooling2D, Flatten, Dropout, Dense,LSTM

#Model Architecture

def CNN_RNN():

    model = Sequential()

    model.add(TimeDistributed(Conv2D(64,(3,3),activation='relu'),input_shape=(1,150,150,
3)))

    model.add(TimeDistributed(MaxPooling2D(pool_size=(2,2))))

    model.add(TimeDistributed(Conv2D(32,(3,3),activation='relu')))

    model.add(TimeDistributed(MaxPooling2D(pool_size=(2,2))))

    model.add(TimeDistributed(Flatten()))

    #RNN

    model.add(LSTM(100,return_sequences=False))

    model.add(Dense(4,activation='softmax'))

    #Getting model summary
```

```
        print(model.summary())

        return model

def DCNN_():

    input_shape=(150,150,3)

    base_cnn = VGG16( weights='imagenet', include_top=False, input_shape=input_shape)

    model = Sequential()

    model.add(base_cnn)

    # don't train existing weights

    for layer in base_cnn.layers:

        layer.trainable = False

    model.add(GlobalMaxPooling2D(name="gap"))

    model.add(Dense(256, activation="relu"))

    model.add(Dense(4, activation="softmax"))

    return model
```

GUI

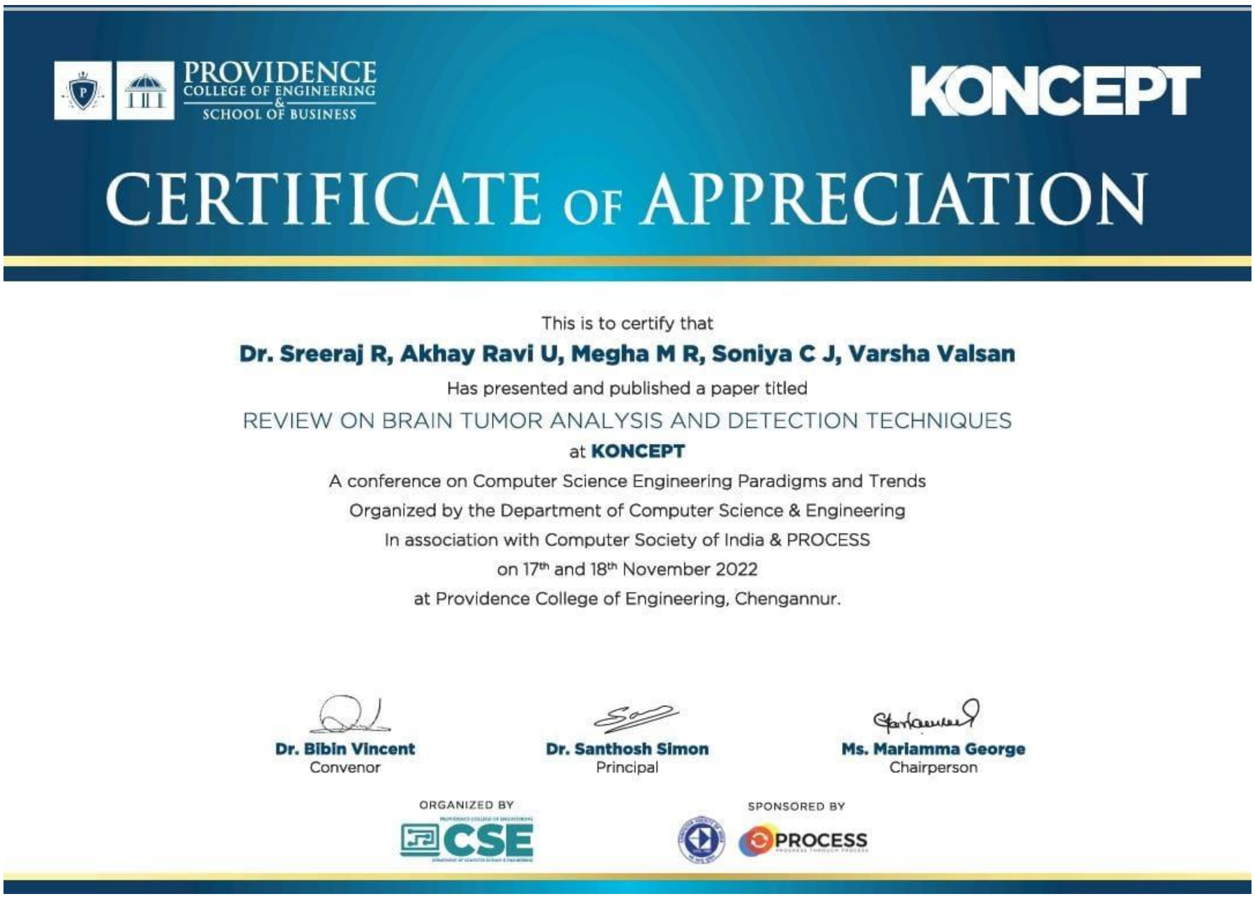
```
from tkinter import *

from tkinter import messagebox

from PIL import ImageTk,Image

from tkinter.filedialog import askopenfilename

import cv2 as cv

import numpy as np

from tensorflow.keras.models import load_model

import cv2
```

```
from grad_cam import get_main
a=Tk()
a.title("Brain Tumor Detector")
a.geometry("1200x600")
a.minsize(1200,600)
a.maxsize(1200,600)
loaded_model1 = load_model("Project_Saved_Models/brain_tumor_model.h5")
def denoise(image):
    #denoising using Non-local mean algorithm
    out = cv2.fastNlMeansDenoisingColored(image,None,10,10,7,21)
    return out
def prediction1():
    list_box.insert(1,"Loading Image")
    list_box.insert(2,"")
    list_box.insert(3,"Image Preprocessing")
    list_box.insert(4,"")
    list_box.insert(5,"Loading BT Detection Model")
    list_box.insert(6,"")
    list_box.insert(7,"Prediction")
    image = cv2.imread(path)
    image_resize=cv2.resize(image,(150,150))
    output=denoise(image_resize)
    img_yuv = cv2.cvtColor(output,cv2.COLOR_BGR2YUV)
```

```
img_yuv[:,:,0] = cv2.equalizeHist(img_yuv[:,:,0])

hist_eq = cv2.cvtColor(img_yuv, cv2.COLOR_YUV2BGR)

print(hist_eq.shape)

out_image=np.expand_dims(hist_eq,axis=0)

out_image=np.expand_dims(out_image,axis=0)/255

print(out_image.shape)

my_pred = loaded_model1.predict(out_image)

print(my_pred)

my_pred=np.argmax(my_pred,axis=1)

print(my_pred)

my_pred=my_pred[0]

print(my_pred)

if my_pred==0:

    print("Glioma Tumor")

    a="Glioma Tumor"

elif my_pred==1:

    a="Meningioma Tumor"

    print("Meningioma Tumor")

elif my_pred==2:

    a="No Tumor"

    print("No Tumor")

elif my_pred==3:

    a="Pituitary Tumor"
```

```
        print("Pituitary Tumor")
    out_label.config(text= a)
    get_main(path)
def Check():
    global f
    f.pack_forget()
    f=Frame(a,bg="white")
    f.pack(side="top",fill="both",expand=True)
    global f1
    f1=Frame(f,bg="Lavender")
    f1.place(x=0,y=0,width=560,height=610)
    f1.config()
    input_label=Label(f1,text="INPUT",font="arial 16",bg="lavender")
    input_label.pack(padx=0,pady=20)
    upload_pic_button=Button(f1,text="Upload Picture",command=Upload,bg="pink")
    upload_pic_button.place(x=240,y=100)
    global label
    label=Label(f1,bg="Lavender")
    global f2
    f2=Frame(f,bg="aquamarine")
    f2.place(x=800,y=0,width=400,height=690)
    f2.config(pady=20)
    result_label=Label(f2,text="RESULT",font="arial 16",bg="aquamarine")
    result_label.pack(padx=0,pady=0)
    global out_label
    out_label=Label(f2,text="",bg="aquamarine",font="arial 16")
    out_label.pack(pady=90)
    global out_label1
    out_label1=Label(f2,text="",bg="aquamarine",font="arial 16")
    out_label1.pack()
    global out_label2
    out_label2=Label(f2,text="",bg="aquamarine",font="arial 16")
    out_label2.pack()
    f3=Frame(f,bg="Salmon")
    f3.place(x=560,y=0,width=240,height=690)
    f3.config()
    name_label=Label(f3,text="Process",font="arial 14",bg="Salmon")
    name_label.pack(pady=20)
    global list_box
    list_box=Listbox(f3,height=12,width=31)
    list_box.pack()
    predict_button1=Button(f3,text="Predict",command=prediction1,bg="deepskyblue")
    predict_button1.pack(side="top",pady=10)
def Upload():
    global path
    label.config(image='')
    list_box.delete(0,END)
    out_label.config(text='')
    path=askopenfilename(title='Open a file',
                         initialdir='Test_Images',
                         filetypes=(("JPG",".jpg"),("JPEG",".jpeg"),("PNG","*.png")))
    print("Path : ",path)
    image=Image.open(path)
    global imagename
    imagename=ImageTk.PhotoImage(image.resize((300,300)))
    label.config(image=imagename)
    label.image=imagename
    # label.pack()
    label.place(x=140,y=210)
def Home():
    global f
    f.pack_forget()
    f=Frame(a,bg="cornflower blue")
    f.pack(side="top",fill="both",expand=True)
    front_image = Image.open("Project_Extra/home.jpg")
    front_photo = ImageTk.PhotoImage(front_image.resize((1200,600), Image.ANTIALIAS))
    front_label = Label(f, image=front_photo)
    front_label.image = front_photo
    front_label.pack()
    home_label=Label(f,text="Brain Tumor Detector",font="arial 35",bg="white")

    home_label.place(x=300,y=290)

f=Frame(a,bg="cornflower blue")

f.pack(side="top",fill="both",expand=True)

front_image = Image.open("Project_Extra/home.jpg")

front_photo = ImageTk.PhotoImage(front_image.resize((1200,600), Image.ANTIALIAS))

front_label = Label(f, image=front_photo)

front_label.image = front_photo

front_label.pack()

home_label=Label(f,text="Brain Tumor Detector",font="arial 35",bg="white")

home_label.place(x=300,y=290)

m=Menu(a)

m.add_command(label="Home",command=Home)

checkmenu=Menu(m)

m.add_command(label="Check",command=Check)

a.config(menu=m)

a.mainloop()
```

APPENDIX – 3



PROVIDENCE COLLEGE OF ENGINEERING & SCHOOL OF BUSINESS

KONCEPT

CERTIFICATE OF APPRECIATION

This is to certify that

Dr. Sreeraj R, Akhay Ravi U, Megha M R, Soniya C J, Varsha Valsan

Has presented and published a paper titled

REVIEW ON BRAIN TUMOR ANALYSIS AND DETECTION TECHNIQUES

at **KONCEPT**

A conference on Computer Science Engineering Paradigms and Trends
Organized by the Department of Computer Science & Engineering
In association with Computer Society of India & PROCESS
on 17th and 18th November 2022
at Providence College of Engineering, Chengannur.

[illegible] [illegible] [illegible]

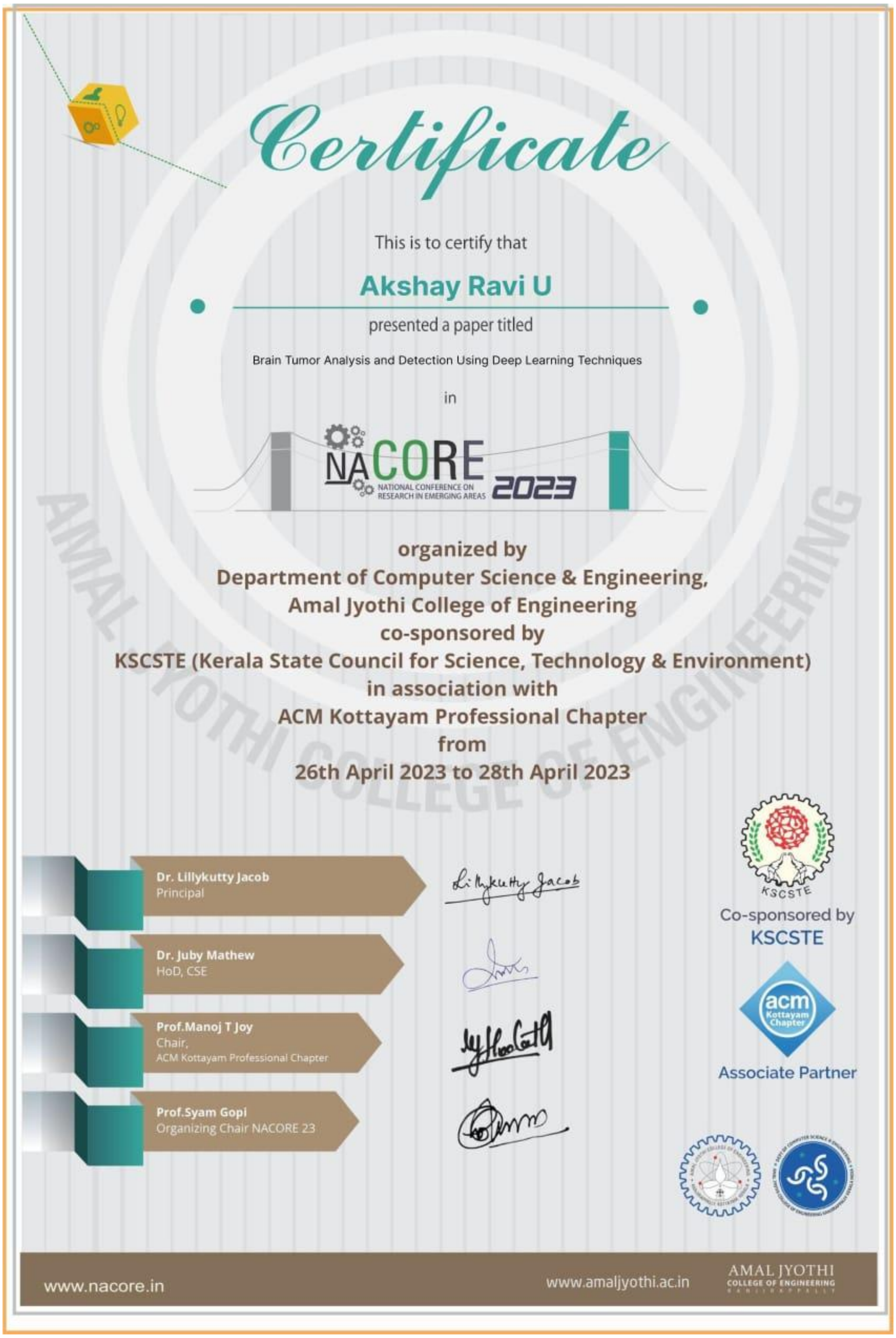
Dr. Bibin Vincent Convenor

Dr. Santhosh Simon Principal

Ms. Mariamma George Chairperson

ORGANIZED BY CSE

SPONSORED BY PROCESS



Certificate

This is to certify that

Akshay Ravi U

presented a paper titled

Brain Tumor Analysis and Detection Using Deep Learning Techniques

in

NACORE 2023
NATIONAL CONFERENCE ON RESEARCH IN EMERGING AREAS

organized by
Department of Computer Science & Engineering,
Amal Jyothi College of Engineering
co-sponsored by
KSCSTE (Kerala State Council for Science, Technology & Environment)
in association with
ACM Kottayam Professional Chapter
from
26th April 2023 to 28th April 2023

Dr. Lillykutty Jacob
Principal

Dr. Juby Mathew
HoD, CSE

Prof.Manoj T Joy
Chair,
ACM Kottayam Professional Chapter

Prof.Syam Gopi
Organizing Chair NACORE 23

Lillykutty Jacob

[illegible]

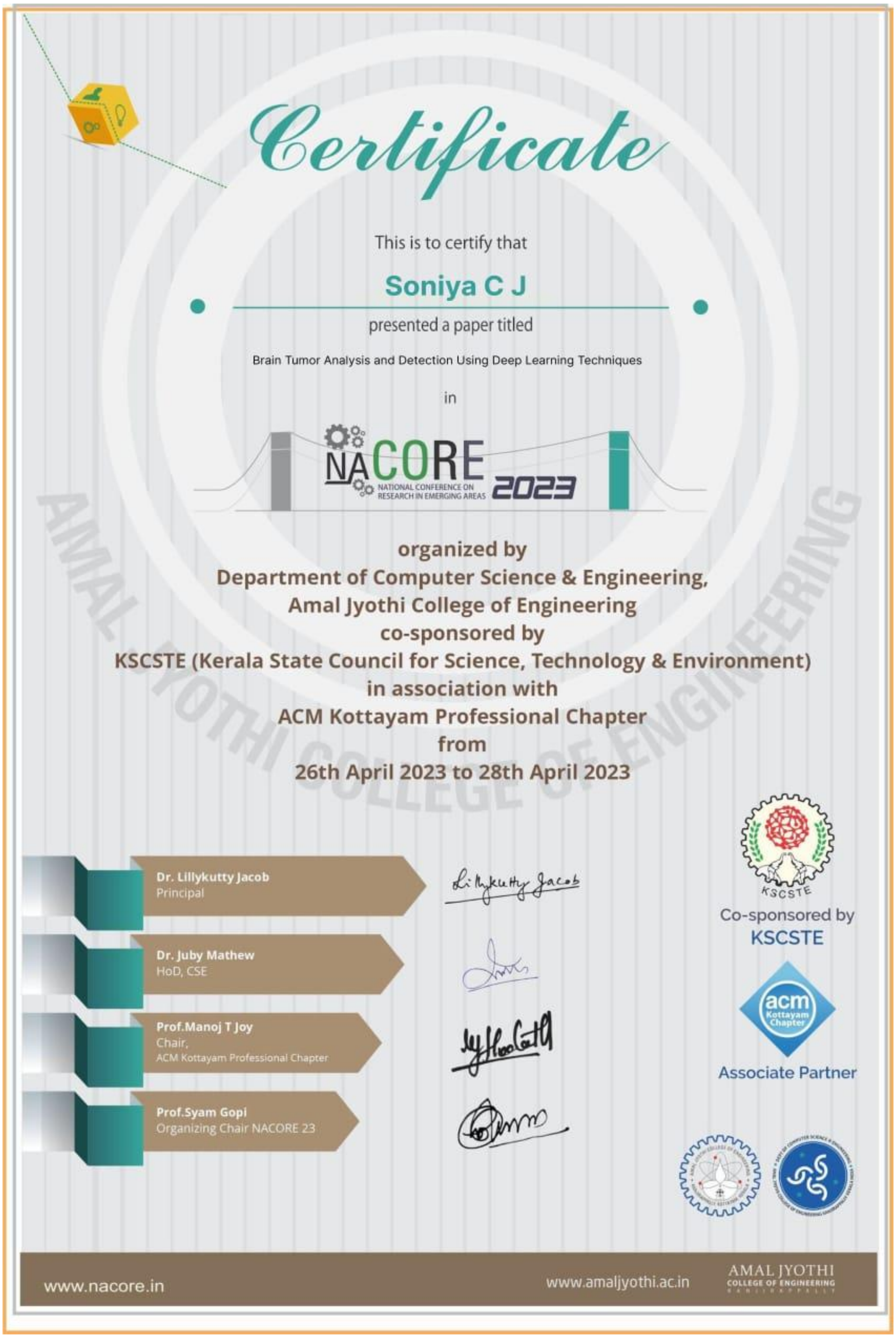
[illegible]

[illegible]

KSCSTE
Co-sponsored by
KSCSTE

acm Kottayam Chapter
Associate Partner

www.nacore.in www.amaljyothi.ac.in AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY



Certificate

This is to certify that

Soniya C J

presented a paper titled

Brain Tumor Analysis and Detection Using Deep Learning Techniques

in

NACORE 2023
NATIONAL CONFERENCE ON RESEARCH IN EMERGING AREAS

organized by
Department of Computer Science & Engineering,
Amal Jyothi College of Engineering
co-sponsored by
KSCSTE (Kerala State Council for Science, Technology & Environment)
in association with
ACM Kottayam Professional Chapter
from
26th April 2023 to 28th April 2023

Dr. Lillykutty Jacob
Principal

Lillykutty Jacob

Dr. Juby Mathew
HoD, CSE

[illegible]

Prof.Manoj T Joy
Chair,
ACM Kottayam Professional Chapter

[illegible]

Prof.Syam Gopi
Organizing Chair NACORE 23

[illegible]

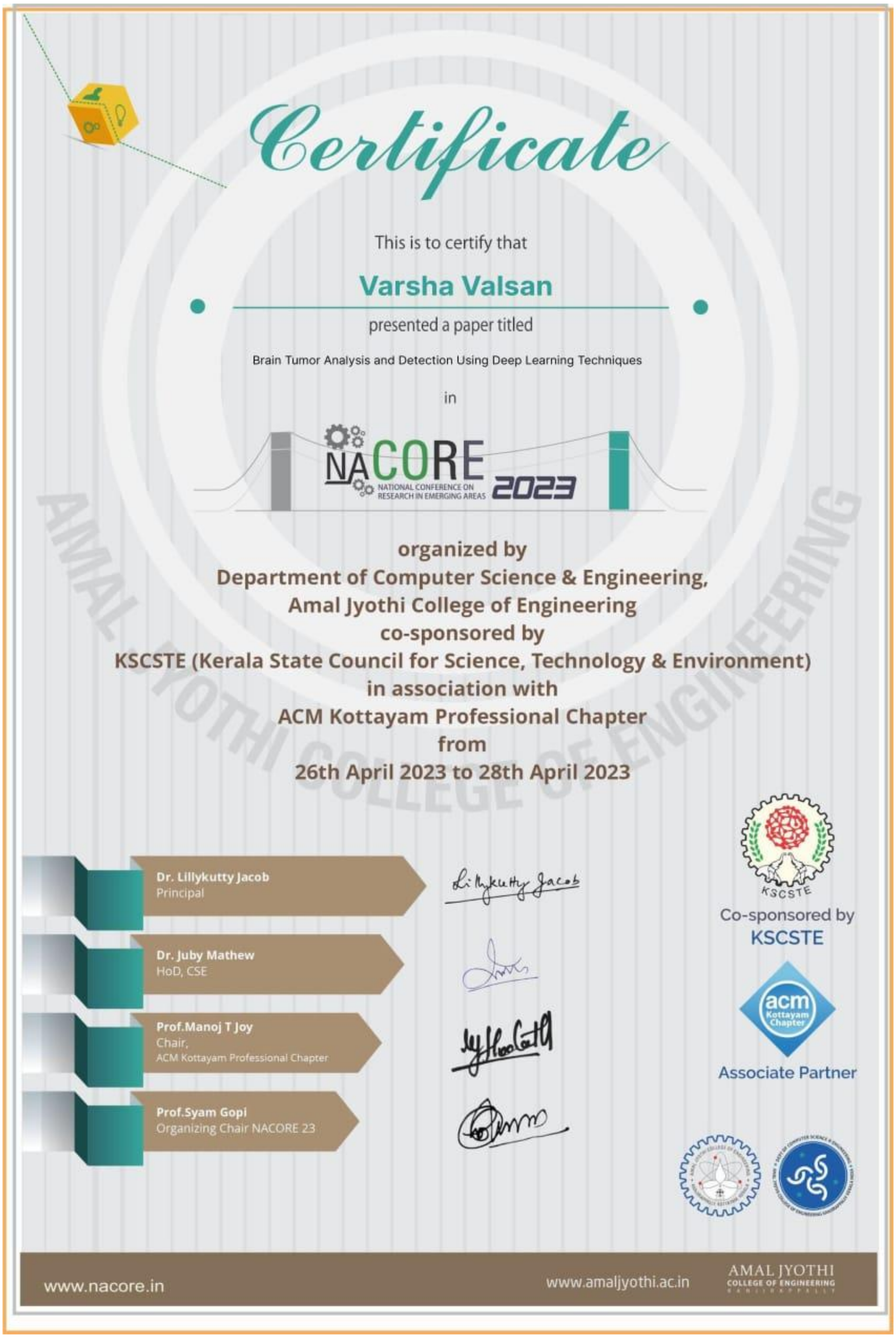
KSCSTE
Co-sponsored by
KSCSTE

acm Kottayam Chapter
Associate Partner

www.nacore.in

www.amaljyothi.ac.in

AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY



Certificate

This is to certify that

Varsha Valsan

presented a paper titled

Brain Tumor Analysis and Detection Using Deep Learning Techniques

in

NACORE 2023
NATIONAL CONFERENCE ON RESEARCH IN EMERGING AREAS

organized by
Department of Computer Science & Engineering,
Amal Jyothi College of Engineering
co-sponsored by
KSCSTE (Kerala State Council for Science, Technology & Environment)
in association with
ACM Kottayam Professional Chapter
from
26th April 2023 to 28th April 2023

Dr. Lillykutty Jacob
Principal

Lillykutty Jacob

Dr. Juby Mathew
HoD, CSE

[illegible]

Prof.Manoj T Joy
Chair,
ACM Kottayam Professional Chapter

[illegible]

Prof.Syam Gopi
Organizing Chair NACORE 23

[illegible]

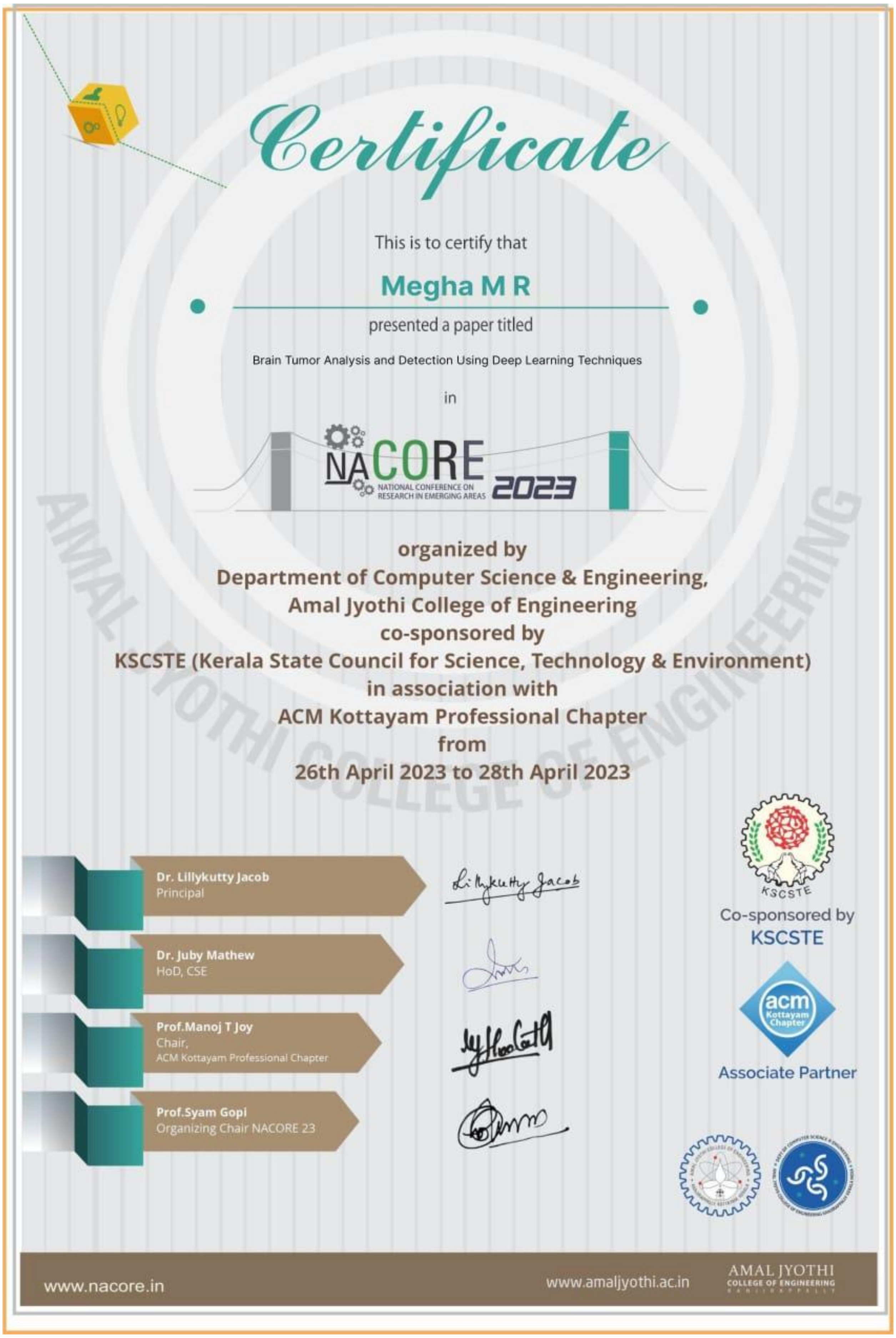
KSCSTE
Co-sponsored by
KSCSTE

acm Kottayam Chapter
Associate Partner

www.nacore.in

www.amaljyothi.ac.in

AMAL JYOTHI COLLEGE OF ENGINEERING



Certificate

This is to certify that

Megha M R

presented a paper titled

Brain Tumor Analysis and Detection Using Deep Learning Techniques

in

NACORE 2023
NATIONAL CONFERENCE ON RESEARCH IN EMERGING AREAS

organized by
Department of Computer Science & Engineering,
Amal Jyothi College of Engineering
co-sponsored by
KSCSTE (Kerala State Council for Science, Technology & Environment)
in association with
ACM Kottayam Professional Chapter
from
26th April 2023 to 28th April 2023

Dr. Lillykutty Jacob
Principal

Lillykutty Jacob

Dr. Juby Mathew
HoD, CSE

[illegible]

Prof.Manoj T Joy
Chair,
ACM Kottayam Professional Chapter

[illegible]

Prof.Syam Gopi
Organizing Chair NACORE 23

[illegible]

KSCSTE
Co-sponsored by
KSCSTE

acm Kottayam Chapter
Associate Partner

www.nacore.in

www.amaljyothi.ac.in

AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY

REVIEW ON BRAIN TUMOR-ANALYSIS AND DETECTION TECHNIQUE

Dr.Sreeraj R
Professor of Computer Science and Engineering
Universal Engineering College
Kerala,India
sreerajr@gmail.com

Akhay Ravi U
Department of Computer science and Engineering
Universal Engineering College
Kerala,India
akshayraviu@gmail.com

Megha M R
Department of Computer and Engineering
Universal Engineering College
Kerala,India
meghamr2222@gmail.com

Soniya C J
Department of Computer science and Engineering
Universal Engineering College
Kerala,India
soniyacj21@gmail.com

Varsha Valsan
Department of Computer science Engineering
Universal Engineering College
Kerala,India
varshavalsan0002@gmail.com

Abstract—A brain tumor is an abnormal growth of cells inside the brain or skull ,some are benign ,others malignant. Diagnosis and treatment will be applied depending on the tumor type, size and location. Therefore we proposed a modified computer aided diagnosis system that detect tumor conditions, the types of tumor and the current stage. Here we proposed to design a Deep Learning model using Recurrent Neural Network(RNNs), Radial Basis Function Networks(RBFNs) models for this project. The evaluation of the proposed model is based on the dataset from kaggle, figshare. Here we consider the modality of image as MRI. The proposed system will be developed using python .

Keywords: Brain tumor, machine learning, deep learning, threasholding, histogram equalization, recurrent neural network, radial basis function networks.

I.INTRODUCTION

The human brain is the main part of the body because it controls most of the human actions such as memory, speech, thoughts and leg and arms movements[1]. Brain deceases are mostly caused by the abnormal growth of brain cells, which directly damage the brain structre and lead to brain cancer[2]. According to world health organization(WHO)records, about 9.6 million in every side of the world died from cancer[3].Brain cancer is dangerous ,rapidly growing and ease deadly. Moreover, the complexity of brain construction is a major challenge, so timely and accurate diagnosis is neccessary. The Magnetic Resonance Images(MRI) can provide better visualization of contrast and special definition[4].The detection of brain abnormalities process is an important issue to determine whether the abnormalities exist or not in MRI Image. Researchers uses Deep Learning in a wide zone with many medical science fields[5].Till now there is no combined model to get more accurate result ,so we implementing a computer aided system for brain tumor detection using Recurrent Neural Network(RNN) and Radial Basis Function Network(RBFN).The paper proposes an efficient deep diagnosis system ,see figure 1.Now some outlines about the paper contribution will be indicated:

- A three step pre-processing method is proposed as initial state.
- This method enhance the quality of MRI images and improve their contrast.
- The combined architecture of RNN and RBFN used to detect tumor or not and then identify type of tumor.
- Batch normalization[18]technique is applied to train the model faster and get a higher learning rate.

We organized the rest of the paper as follows: First section II discuss the related work ,secondly ,section III describes the applied methodologies in this paper ,thirdly section IV present the concluding remarks.

II.LITERATURE REVIEW

Recently, there're many studies and researches about detecting brain tumor in MRI images. In this section, several reliable works are explored. Varthanana et al. [5] presented a method for brain tumors detection using a novel self organizing map (SOM) and fuzzy k-mean (FKM). Their segments results have been validated by experienced radiologists. However, their proposed approach is complex and time consuming with real practical applications. Dhanachdra et al. [6] proposed a technique to improve MRI images quality. Their technique computes the initial value of cluster centers with helping of a subjective algorithm. They used another contrast stretching algorithm to enhance the input image quality. They also used K-mean algorithm in their classification process, but still, there're lack of classification accuracy. Varana et al. [7] used discrete wavelet transform (WDT) based on brain abnormal region. They explored a probabilistic neural network (PNN) to detect brain tumors in the MRI images. Sachdera et al. [8] proposed a principal component analysis-artificial neural network (PCA-ANN) for several classed brain tumor classification. They get a number of regoin of interests (ROIs) by the content-based contour (CBAC). Their experiments results showed respectable enhance in the accuracy from 77% without PCA to 91% with PCA. Bahadure et al.used the Support Vector Machine (SVM) for the classification process. They also explored a Berkeley wavelet transform (BWT) for brain tumor detection. They extracted the relevant features then input them to the SVM. Corso et al.[9] proposed an automatic segmentation approach by combining a generative model-based technique and a graph-based affinities method. Their model was inserted into multi-level segmentation using a weighted aggregation algorithm. Abiwinanda et al. [10] investigated three common forms ofbrain tumors namely Glioma, Meningioma, and Pituitary byexploiting Convolutional Neural Network (CNN). The CNN implementation is accomplished from a hidden layer by con-volution, max-pooling, and flattening layers respectively withthe entire connection. The CNN training for analyzing brain tumor utilizes 3064 T-1 weighted Contrast-Enhanced Magnetic Resonance Imaging (CE-MRI) images which is accessible via figshare Cheng. The training accuracy of 98.51%and validation accuracy of 84.19% is attained expending basic architecture and deprived of any prior region-based segmentation which is contrasted with other complicated region-based segmentation algorithms where accuracy lies between 71.39%and 94.68% on identical dataset. Pashaei et al. [11] demonstrated for extracting image hidden features by utilizing CNN beside with Kernel Extreme Learning Machine (KELM). The several brain tumors forms such as meningioma, glioma and pituitary tumor are concen-trated for effectiveness evaluation of the projected system in T1-weighted CE-MRI images. The CNN and KELM(KE-CNN) combination are contrasted with Support Vector Machine, Radial Base Function, and some other classifiers which reveal improved outcomes. Gopi et al. [12] The research work shows a better clustering results by considering the two popular types of tumors such as as benign and malignant for classification through clustering. Feature extraction has been accomplished by GLCM technique and image segmentation by utilizing FCM, Fast FCM and k-means algorithm. The proposed PSO based RBFN model has shown the potentiality of clustering of the tumor. The automatic detection and classification using the proposed RBFNN model with PSO-WCA training is the main contribution of the research work. The feature variance playedavital rolein clustering classification in comparison to the other features. The feature variance have given adequate classification results with kurtosis with variance, variance with skewness, and variance with energy. The proposed RBFNN with PSO-WCA model has been assigned for the classification and the results were compared with some approaches like PSO-RBFNN,WCA-RBFNN etc .Sivasai et al. [13] 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 adaptive bilateral filter (ABF) for removal of the noises that are present in an MR image. This was followed by the binary threasholding and Fuzzy Recurrent Neural Network (FR-Net) 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. In our paper, we proposed a tumor detection system that presented good accuracy and less computational time when compared to its contour parts. The proposed system begins with reading the MRI image from the dataset, and then image pre-processing is done by using filtering techniques such as adaptive bilateral for removal of noise pixels present in the original brain tumor image.

III.METHODOLOGY

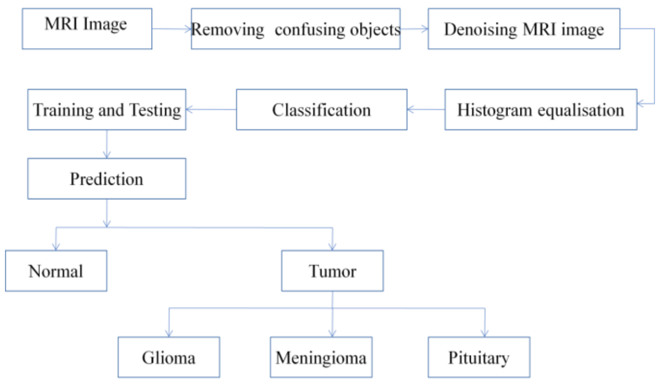


Figure 1.Block Diagram

A. THE PROPOSED PRE-PROCESSING APPROACH

In the classification challenge for detecting the brain tumor in MRI images, the identification of a correct pattern is the main key in the classification process. Many issues in the MRI images face the classification models, which mislearning can happen and leads award downgrading the classification accuracy. So, we proposed a three-step pre-processing approach.

1) REMOVING THE CONFUSING OBJECTS

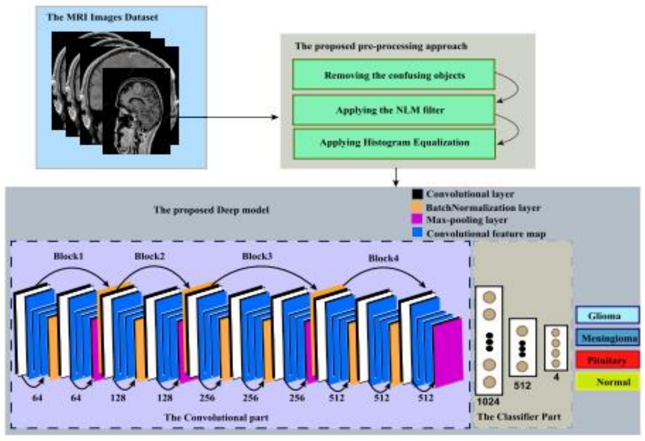
Confusing objects such as texts and black areas on the right and left corners have been removed by cropping 100 pixels from each side of the image to get the exact brain object .

2) DENOISING THE MRI IMAGES

Non-local mean algorithm (NLM)deal with noise efficiently in MRI images. The noise in these images lead to learning undesirable patterns consequently, downgrading the classification accuracy. The NLM algorithm greatly enhances the quality of the MRI images as compared with Gaussian and Median algorithms according to the blind reference less image spatial evaluator (BRISQUE) .

3)HISTOGRAM EQUALIZATION

Histogram Equalization extremely enhances the contrast in the MRI images. Moreover, it allows the detecting small details by setting regions lower contrast with appropriate contrast. It accomplishes this process by performing a separation to the most frequent intensity values. It also clears up the interference of the most frequent patterns in the MRI images .



B.DATASET

The dataset that has been used in the experiments and test formed based on Sartaj brain MRI images dataset [14] and the Navoneel brain tumor dataset . The used dataset [15] contains two types of MRI[16] images: T1-weighted and T2 weighted. T1 weighted images are produced using short time to echo (TE) and repetition time (RT) constraints, which are 14 and 500 milliseconds, respectively. T2-weighted images are produced using longer TE and RT constraints, which are 90 and 4000 milliseconds, respectively. The dataset has been divided into three folders (Training, Testing and Validating), with sub-folders for each class (GLIOMA, MENINGIOMA, NO TUMOR and PILUITARY). There're 3394 MRI images organized into four classes of GLIOMA (934), MENINGIOMA (945),NO-TUMOR (606) and PILUITARY (909). The training folder contains 826 images for GLIOMA, 822 images for MENINGIOMA, 827 images for PILUITARY.The used dataset details:493 images for NO-TUMOR. The testing folder contains 100 images for GLIOMA, 115 images for MENINGIOMA, 105 images for PILUITARY and 74 images for NO-TUMOR .The validating folder contains 8 images for GLIOMA, 8 images for MENINGIOMA, 8 images for PILUITARY and 8 images for NO-TUMOR available on [17].

C.THE PROPOSED MODEL

This paper propose a combination architecture of RNN and RBFN. Recurrent Neural Network(RNN) are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

Radial basis function (RBF) networks are a commonly used type of artificial neural network for function approximation problems. Radial basis function networks are distinguished from other neural networks due to their universal approximation and faster learning speed. An RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer. Each of these layers has different tasks .

This proposed model resolves many issues such as decreasing the overfitting ,slow learning rates and lack of training accuracy due to the batch normalization operation .

IV.CONCLUSION

As of now, a brain tumor has considered as one among the vulnerable life-threatening diseases .Anomalous cells of the human brain have gathered as cluster and surround the inner part, namely brain tumor. The patients are needed for segmenting and identifying the tumor in a perfect and strong manner in terms of diagnosis. Hence a deep convolution neural network architecture is proposed for glioma, meningioma and pituitary brain diseases detection with an objective of high classification accuracy within a short time. first, a proper brain tumor dataset for efficiently performing the training and testing process. Second, a threes tep pre-processing approach was removing the confusing variables, denoising the MRI images and enhancing the contrast of these images. This approach positively and directly reflected on all of the explored models. Third, a training strategy includes training our model on the desirable patterns from scratch. Fourth, we hired our model to extract the MRI images features and efficiently classify them. We evaluate the proposed model on a dataset with 394 MRI images. The proposed model accomplished an accuracy of 97.72% overall, 99% in detecting glioma, 98.26% in detecting meningioma, 95.95% in detecting pituitary and 97.14% in detecting normal images. In real practice, the proposed model can be considered as an automated computer-aided detector tool to timely detect brain abnormalities in MRI images with high accuracy.

REFERENCES

[1] N. B. Bahadure, A. K. Ray, and H. P. Thethi, ‘‘Image analysis for MRI based brain tumor detection and feature extraction using biologically inspired BWT and SVM,’’ Int. J. Biomed. Imag., vol. 2017, pp. 1–12,Mar.2017,doi:10.1155/2017/9749108.

[2] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, ‘‘Brain tumor segmentation using convolutional neural networks in MRI images,’’ IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1240–1251, May 2016, doi:10.1109/TMI.2016.2538465.

[3] World Health Organization. Accessed: Jun. 10, 2021. [Online]. Available:https://www.who.int.

[4] E. El-Dahshan, H. Mohsen, K. Revett, and A. Salem, ‘‘Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm,’’ Expert Syst. Appl., vol. 41, no. 11, pp. 5526–5545, 2014, doi:10.1016/j.eswa.2014.01.021.

[5] G. Vishnuvarthanan, M. P. Rajasekaran, P. Subbaraj, and A. Vishnuvarthanan, ‘‘An unsupervised learning method with a clustering approach for tumor identification and tissue segmentation in magnetic resonance brain images,’’ Appl. Soft Comput., vol. 38, pp. 190–212,Jan. 2016, doi: 10.1016/j.asoc.2015.09.016.

[6] N. Dhanachandra, K. Manglem, and Y. J. Chanu, ‘‘Image segmentation using K-means clustering algorithm and subtractive clustering algorithm,’’ Proc. Comput. Sci., vol. 54, pp. 764–771, Jan. 2015, doi:10.1016/j.procs.2015.06.090.

[7] N. V. Shree and T. N. R. Kumar, ‘‘Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network,’’ Brain Informat., vol. 5, no. 1, pp. 23–30, Mar. 2018, doi:10.1007/s40708-017-0075-5.

[8] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja,‘‘Segmentation, feature extraction, and multiclass brain tumor classification,’’ J. Digit. Imag., vol. 26, no. 6, pp. 1141–1150, Dec. 2013, doi:10.1007/s10278-013-9600-0.

[9] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille,‘‘Efficient multilevel brain tumor segmentation with integrated Bayesian model classification,’’ IEEE Trans. Med. Imag., vol. 27, no. 5, pp. 629–640,May 2008, doi: 10.1109/TMI.2007.912817.

[10] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, ‘‘Brain tumor classification using convolutional neural network,’’ in World Congress on Medical Physics and Biomedical Engineering. Singapore: Springer, 2018, pp. 183–189.

[11] A. Pashaei, H. Sajedi, and N. Jazayeri, ‘‘Brain tumor classification via convolutional neural network and extreme learning machines,’’ in Proc.8th Int. Conf. Comput. Knowl. Eng. (ICCKE), Oct. 2018, pp. 314–319.

[12] T. Gopi Krishna a*, Satyasis Mishra b, Sunita Satapathy c, K. V. N. Sunitha d and Mohamed A. Abdelhadi e,"Classification of Brain Tumor Types on Magnetic Resonance Images Using Hybrid Deep Learning Approach with Radial Basis Function Neural Network",DOI: 10.9734/bpi/ramrcs/v7/1579B.

[13] Jalluri Gnana SivaSai, P. Naga Srinivasu, Munjila Naga Sindhuri,Kola Rohitha, and Sreesailam Deepika,"An Automated Segmentation of Brain MR Image Through Fuzzy Recurrent Neural Network", https://doi.org/10.1007/978-981-15-5495-7_9.

[14] Sartaj. Brain Tumor Classification (MRI) Dataset.Accessed: Jun. 10, 2021. [Online]. Available: https://www.kaggle.com.

[15] N. Chakrabarty. Brain MRI Images for Brain Tumor Detection Dataset.Accessed: Jun. 10, 2021. [Online]. Available: https://www.kaggle.com.

[16] D. C. Preston. Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics. Accessed: Dec. 31, 2021. [Online]. Available: https://case.edu/med/neurology/NR/MRI%20Basics.htm.

[17] https://figshare.com/articles/dataset/brain_tumor_dataset/1512427.

[18] S. Ioffe and C. Szegedy, ‘‘Batch normalization: Accelerating deep network training by reducing internal covariate shift,’’ 2015, arXiv:1502.03167.

International Journal on Emerging Research Areas (IJERA) *ISSN:2230-9993*

DOI:

BRAIN TUMOR ANALYSIS AND DETECTION USING DEEP LEARNING TECHNIQUES

Dr.Sreeraj R
Professor of Computer Science And Engineering
Universal Engineering College, Kerala, India
sreerajr@gmail.com

Akshay Ravi U
Department of Computer Science And Engineering
Universal Engineering College, Kerala, India
akshayraviu@gmail.com

Megha M R
Department of Computer Science And Engineering
Universal Engineering College, Kerala, India
meghamr2222@gmail.com

Soniya C J
Department of Computer Science And Engineering
Universal Engineering College, Kerala, India
soniyacj21@gmail.com

Varsha Valsan
Department of Computer Science And Engineering
Universal Engineering College, Kerala, India
varshavalsan0002@gmail.com

Abstract— **A brain tumor is an abnormal growth of cells inside the brain or skull, some are benign, others malignant. Diagnosis and treatment will be applied depending on the tumor type, size and location. Therefore, we proposed a modified computer aided diagnosis system that detect tumor conditions, the types of tumors. The Recurrent Neural Network and Convolutional Neural Network (RNN+CNN) architecture is proposed in this research coupled with a three-step preprocessing method to improve the quality of MRI images for use in diagnosing gliomas, meningiomas, and pituitary tumors. The architecture makes advantage of batch normalization for quick training, a greater learning rate, and simple layer weight initialization. 99.27% accuracy is attained overall. The MRI dataset from Kaggle is used to evaluate the recommended model. The proposed system will be developed using python.**

Keywords: Brain tumor, machine learning, deep learning, thresholding, histogram equalization, recurrent neural network, radial basis function networks.

I.INTRODUCTION

The majority of human behaviors, including memory, speech, thought, and leg and arm motions, are controlled by the brain, which is the principal component of the human body. The majority of brain diseases are brought on by aberrant brain cell proliferation, which directly harms the brain's structure and causes brain cancer. Records from the World Health Organization (WHO) show that cancer claimed the lives of nearly 9.6 million people worldwide [1]. Brain cancer is lethal, fast spreading, and dangerous. Furthermore, the complexity of the brain's development presents a significant hurdle, necessitating prompt and correct diagnosis. Better contrast and particular definition can be seen with magnetic resonance imaging (MRI) [2]. To ascertain whether or not there are abnormalities in an MRI image, the method of detecting brain abnormalities is crucial. Deep Learning is being used by researchers in a variety of medical science fields. We are developing a computer-aided method for brain tumor diagnosis using Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) because there is currently no integrated model to produce more accurate results. In the project, an effective deep diagnosis method is proposed (Figure 1). Here are some details regarding the paper contribution:

- As a starting state, a three-step pre-processing procedure is suggested.
- By using this technique, MRI pictures' quality and contrast are improved.
- The combined RNN and CNN architecture is utilized to determine tumor and its types.
- Batch normalization [15] is applied to train the model faster and get a higher learning rate.

We organized the rest of the paper as follows: First section II discusses the related work, secondly, section III describes the applied methodologies in this paper, thirdly section IV shows the experimental result, then section v presents the concluding remarks.

DOI:

II. LITERATURE REVIEW

There have been a lot of studies and researches recently about finding brain tumors in MRI pictures.This section examines a number of credible works Bahadureetal.[3]. Investigated through Berkeley wavelet transformation (BWT) ba sed brain tumour segmentation to enhance efficiency while redu cing complexity in the medical image segmentation process.Rele vant features are also extracted from each segmented tissue to in crease the support vector machine (SVM) based classifier's accu racy and quality rate.Based on accuracy, sensitivity, specificity, and dice similarity index coefficient, the experimental findings o f the suggested technique have been assessed and validated for p erformance and quality analysis on magnetic resonance brain im ages.The testing results showed 96.51% accuracy, 94.2%specific ity, and 97.72% sensitivity, proving that the suggested technique for distinguishing between normal and pathological tissues from brain MR images works. Pereira et al.[4]explained that Brain tumour segmentation is significant in medical image proce ssing.It is crucial to identify these tumoursearly in order to treat patients.The earlier it is discovered, the better the patient's chanc es of survival.It takes time and is challenging.Enhanced Convolu tional Neural Networks (ECNN) with loss function optimisation using BAT algorithm for automatic segmentation method are su ggested as a potential fix to these issues.Putting forward optimiz ationbased MRI picture segmentation is the desired outcome. Varthanana et al. [5] provided a technique for detecting brain cancers utilizing a new self-organizing map (SOM) and Fuzzy K-Mean (FKM). Their segments outcomes have been verified by knowledgeable radiologists. Unfortunately, their suggested strategy is complicated and laborious with actual useful applications. Dhanachdra et al. [6] suggested a method to enhance the quality of MRI pictures. Their method uses an arbitrary approach to compute the cluster centre’s starting value. To improve the quality of the supplied image, they utilized another contrast stretching algorithm. Although they also employed the K-mean technique, the categorization process was not accurate enough. Varana et al. [7] based on a brain aberrant region, Discrete Wavelet Transform (WDT) was employed. To find brain cancers in the MRI scans, they investigated a Probabilistic Neural Network (PNN). Sachdera et al.[8] suggested a PCA-ANN (Principal Component Analysis-Artificial Neural Network) for the categorization of various types of brain tumors. The ROIs they receive from the Content-Based Contour are numerous (CBAC). The results of their trials revealed a respectable improvement in accuracy, going from 77% without PCA to 91% with PCA. Corso et al.[9] combined a graph-based affinity method with a generative model-based strategy to present an automatic segmentation solution. Using a weighted aggregation approach, their model was added to multi-level segmentation. Abiwinanda et al. [10] used a convolutional neural network to study three prevalent types of brain malignancies, including glioma, meningioma, and pituitary tumors (CNN). From a hidden layer, convolution, max-pooling, and flattening layers are used in turn with the full connection to build CNN. The 3064 T-1 weighted Contrast-Enhanced Magnetic Resonance Imaging (CE-MRI) images used in the CNN training for the analysis of brain tumors are available via figshare Cheng. The basic architecture was used to achieve training accuracy of 98.51% and validation accuracy of 84.19% without the need of any prior region-based segmentation, in contrast to more complex region-based segmentation algorithms, whose accuracy ranges from 71.39% to 94.68% on the same dataset. Pashaei et al.[11]proposed an Extraction of hidden characteristics from images using CNN and the Kernel Extreme Learning Machine was demonstrated (KELM). In T1-weighted CE-MRI pictures, the various types of brain tumors—including meningioma, glioma, and pituitary tumors—are combined to assess the predicted system's performance. When compared to other classifiers that produce better results, such as Support Vector Machine and Radial Base Function, the CNN and KELM(KE-CNN) combo outperforms them. Gopi et al. [12] By using the two common categories of tumors, benign and malignant, for clustering-based categorization, the research work produces improved clustering outcomes. The GLCM approach was used to extract features, while FCM, Fast FCM, and the k-means algorithm was used to segment the images. The suggested PSO-based RBFN model has demonstrated the possibility of tumor clustering. The study work's key contribution is the automatic identification and classification utilizing the suggested RBFNN model with PSO-WCA training. In comparison to the other features, the feature variance played a significant influence in clustering categorization. Kurtosis with variance, skewness with variance, and energy with variation from the feature variance have provided satisfactory classification results. Sivasai et al. [13] suggest Put your attention on the usage of several methods for finding brain cancer utilizing brain MRI. In this study, we used the Adaptive Bilateral Filter (ABF) for pre-processing to get rid of the sounds that are present in an MR image. Then, for accurate tumor region recognition, the binary thresholding and Fuzzy Recurrent Neural Network (FR-Net) segmentation algorithms were used. The inceptionv3 model's deep features[16] are retrieved and us ed t distinguish between pituitary tumours, gliomas, meningio mas, and no tumours using a score vector obtained by softmax and fed into the quantum variational classifier (QVR).The clas sified tumour photos have been sent to the suggested Segnetw ork, where the real infected area is subdivided to assess the tu mour severity level.On three benchmark datasets, including Ka ggle, 2020BRATS, and locally gathered photos,the results of t he research that was reported were reviewed.The proposed mo del's efficacy was demonstrated by the model's detection score s of better than 90%.

The proposed system begins by reading the MRI image from the dataset, followed by a three-step image pre-processing procedure to improve the quality of the MRI image and a reliable combined network (RNN+CNN) for precisely detecting brain tumors.

DOI:

III.METHODOLOGY

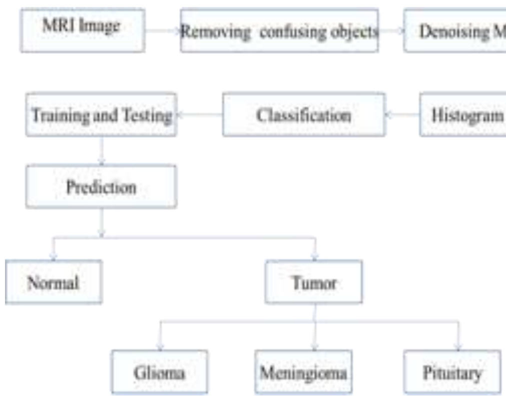


Figure 1. Block Diagram

A. THE PROPOSED PRE-PROCESSING APPROACH

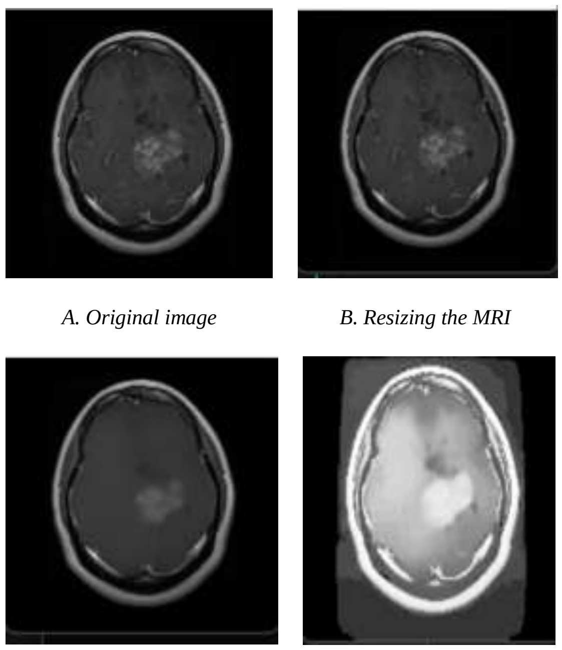
Finding the right pattern is the key to successfully classifying the brain tumor in MRI pictures, according to a classification challenge. The classification models in MRI images must deal with a number of difficulties, which might result in mislearning and lower classification accuracy ratings. Thus, we suggested a three-step pre-processing strategy.

1) REMOVING THE CONFUSING OBJECTS

Then the first step is to remove the confusing objects from images which helps to improve the quality of an image after removing irrelevant image data from image in various applications and domains. Medical images contain lot of irrelevant and unwanted parts in its actual format of the scanned images. To remove such annoying parts in an image, it is required some of the image preprocessing techniques in order for better visualization of the images before finding the diseases in particular. This step provides a fast and accurate means to predict the location of an unwanted object in an image. By subtracting 100 pixels from either side of the image, confusing elements like letters and the dark corners on the right and left have been removed to reveal the precise brain object in figure 2(B).

2) DENOISING THE MRI IMAGES

Medical images often consist of low-contrast objects corrupted by random noise arising in the image acquisition process. Thus, image denoising is one of the fundamental tasks required by medical imaging analysis. Recently non-local means has been extended to other image processing applications such as de-interlacing, view interpolation, and depth maps regularization. The Non-local Mean Algorithm (NLM) effectively reduces noise in MRI images. The noise in these images causes unfavourable patterns to be learned, which lowers the classification accuracy. The MRI images' quality is significantly improved by the NLM algorithm.



A. Original image *B. Resizing the MRI*

C. *Denoising* *D. Histogram Equalization*

Figure 2. Image Pre-processing

3) HISTOGRAM EQUALIZATION

Histogram Equalization techniques help to enhance the image so that it gives an improved visual quality and a well defined problem. The contrast and brightness is enhanced in such a way that it does not lose its original information and the brightness is preserved. Image histograms are an important tool for inspecting images. They allow you to spot Back Ground and grey value range at a glance. Also clipping and Quantization Noise in image values can be spotted immediately. The contrast in the MRI pictures is greatly enhanced by histogram equalization . Also, by setting zones with lower contrast and suitable contrast, it enables the detection of small details. It completes this procedure by executing a separation to the intensity values with the highest frequency. Also, as seen in figure 3, it removes the interference of the most prevalent patterns in the MRI scans.

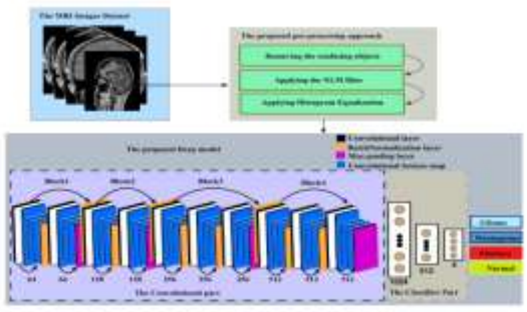


Figure 3. Image Pre-processing

DOI:

B. DATASET

The brain MRI pictures dataset [14] from kaggle served as the foundation for the dataset that was used in the experiments and tests. There are 2870 photos in the input dataset. Glioma 826, meningioma 822, and pituitary 827 are three of the 2475 tumour photos in this dataset, along with 395 non-tumor images. In the original dataset on Kaggle, the non-tumor photos folder's name was "no_tumor." After pre-processing the photos, a 70%–30% split was used to separate the training dataset from the validation dataset. Pre-processing steps included histogram equalization, scaling, denoising, and the removal of distracting elements.

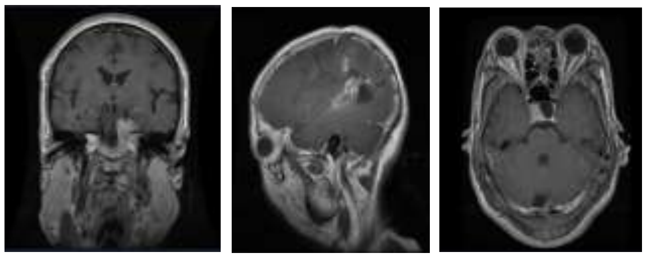


Figure 4. Shows three types of tumors(Meningioma Glioma tumor, pituitary)

C.THE PROPOSED MODEL

In this paper, a hybrid RNN/CNN architecture is proposed. Recurrent Neural Networks (RNNs) are a type of neural network in which the results of one step are fed into the next step's computations. Traditional neural networks have inputs and outputs that are independent of one another, but there is a need to remember the previous words in situations where it is necessary to anticipate the next word in a sentence. As a result, RNN was developed, which utilized a Hidden Layer to resolve this problem. The Hidden state, which retains some information about a sequence, is the primary and most significant characteristic of RNNs.

Convolutional Neural Networks (CNNs, or ConvNets) are a type of artificial neural network (ANN) used most frequently in deep learning to analyze visual data. Based on the shared-weight architecture of the convolution kernels or filters that slide along input features and produce translation-equivariant responses known as feature maps, CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN). Contrary to popular belief, most convolutional neural networks do not translate invariantly because of the down-sampling operation they perform on the input. They have uses in the recognition of images and videos, recommender systems, classification and segmentation of images, analysis of images used in medicine, natural language processing, brain-computer interfaces, and financial time series.

Multilayer perceptrons are modified into CNNs. Fully linked networks, or multilayer perceptrons, are those in which every neuron in one layer is connected to every neuron in the following layer. Due to their "full connectivity," these networks are vulnerable to data overfitting. Regularization techniques that prevent overfitting often involve reducing connectivity or penalising training parameters. By utilizing the hierarchical structure in the data and assembling patterns of increasing complexity utilizing smaller and simpler patterns imprinted in their filters, CNNs use a different strategy for regularization. CNNs are therefore at the lower end of the connectivity and complexity spectrum. This proposed model resolves many issues such as decreasing the overfitting, slow learning rates and lack of training accuracy.

1) CNN+RNN Architecture

We will employ a CNN combined LSTM based deep learning architecture. The CNN LSTM architecture combines LSTMs for sequence prediction with Convolutional Neural Network (CNN) layers for feature extraction from input data. A CNN LSTM can be created by first adding CNN layers, then LSTM layers, and finally a Dense layer at the output. We must put together the deep learning architecture with assigned loss functions and optimizers after generating the model.

2). Training

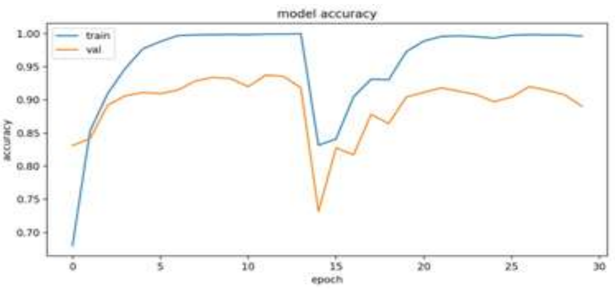
We will use the train set to train our CNN LSTM model and the test set to validate it. Following that, we will calculate the accuracy and save the trained model.

3). Prediction

First, we will read the MRI image using the cv2 library, resize it, perform denoising and histogram equalisation on it, then load the trained model. After loading the model, the pre-processed image will be input to the model, and the model will predict whether the person has a brain tumour (Glioma/Meningioma/Pituitary) or not.

IV. EXPERIMENTAL RESULT

The model has been implemented using Python and Kera library on TensorFlow , Google Colaboratory note along with Github where the used dataset is uploaded . We hired the train part of the used dataset in the training process. Figure 5 shows the training accuracy and the training loss of the proposed model along with the discussed models.



DOI:

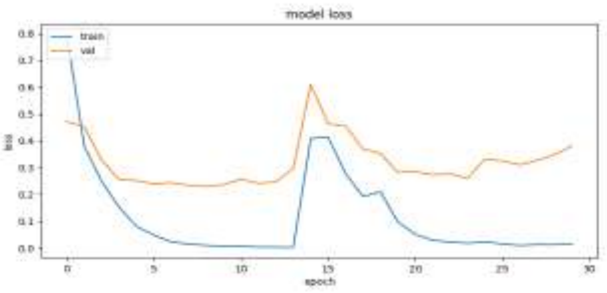


Figure 5: The Training accuracy and Training loss .

As you can see in the diagram, the accuracy increases rapidly in the first five epochs, indicating that the network is learning fast. Generally, if the training data accuracy keeps improving while the validation data accuracy gets worse, if data are encountering overfitting. It indicates that the model is starting to memorize the data. At training, we get 99%, and at validation, we get 93%. The loss on the training set decreases rapidly for the first five epochs. For the test set, the loss does not decrease at the same rate as the training set, but remains almost flat for multiple epochs. This means our model is generalizing well to unseendata.A measurement used to evaluate how well a deep lea rning model fits the training data is called training loss.

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.A multi-class confusion matrix here that is pituitary_tumor, no_tumor,meningioma_tumor,glioma_tumor.The explored models have two main parts in their structures, the convolutional part and the classifier part. The convolutional part extracts the inputted image’s features and the classifier part classifies these features into one of the intended classes.

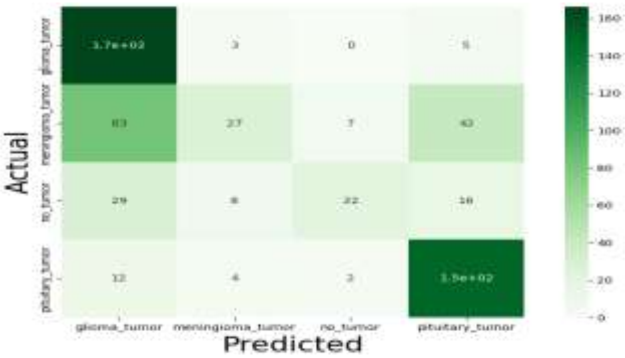


Figure 6. Confusion Matrix

Since "Brain Tumor" has four classes, we must modify the classifier component of these models. Max-pooling layers help speed up diagnosis of MRI images since they reduce the output size of the convolutional layer. However, some MRI picture features may no longer be present as a result of these layers. Confusion matrix shows the actual and predicted outcomes of the input dataset.

V.CONCLUSION

A brain tumour is currently thought to be one of the most susceptible life-threatening illnesses. The inner portion of the human brain, a brain tumor, is surrounded by abnormal cells that have gathered into a cluster. In order to diagnose the tumour accurately and effectively, the patients are required for segmenting and finding the tumour. In order to detect glioma, meningioma, and pituitary brain diseases, a deep convolution neural network architecture is suggested with the goal of achieving high classification accuracy quickly. first, a suitable brain tumour dataset for carrying out the training and testing procedure quickly. Second, a three-step pre-processing method was used to clear the MRI images of any distracting factors, denoise them, and improve their contrast. All of the investigated models were favourably and significantly impacted by this method. Third, as part of a training plan, we train our model from scratch on the desired patterns. Fourth, we employed our model to quickly and accurately classify the MRI images based on their features. We test the suggested model using a dataset containing 2870 MRI pictures. The precision of the suggested model was 99%. In actual use, the suggested model can be viewed as an automated computer-aided detector instrument to accurately and promptly identify brain abnormalities in MRI images.

References

[1] *World Health Organization. Accessed: Jun. 10, 2021. [Online]. Available:https://www.who.int.*

[2] *E. El-Dahshan, H. Mohsen, K. Revett, and A. Salem, ‘‘Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm,’’ Expert Syst. Appl., vol. 41, no. 11, pp. 5526–5545, 2014, doi:10.1016/j.eswa.2014.01.021.*

[3] *N. B. Bahadure, A. K. Ray, and H. P. Thethi, ‘‘Image analysis for MRI based brain tumor detection and feature extraction using biologically inspired BWT and SVM,’’ Int. J. Biomed. Imag., vol. 2017, pp. 1–12,Mar.2017,doi:10.1155/2017/9749108.*

[4] *S. Pereira, A. Pinto, V. Alves, and C. A. Silva, ‘‘Brain tumor seg-mentation using convolutional neural networks in MRI images,’’ IEEE Trans. Med. Imag., vol. 35, no. 5, pp. 1240–1251, May 2016, doi:10.1109/TMI.2016.2538465.*

[5] *G. Vishnuvarthanan, M. P. Rajasekaran, P. Subbaraj, and A. Vishnuvarthanan, ‘‘An unsupervised learning method with a clustering approach for tumor*

DOI:

identification and tissue segmentation in magnetic resonance brain images,'' Appl. Soft Comput., vol. 38, pp. 190–212,Jan. 2016, doi: 10.1016/j.asoc.2015.09.016.

[6] N. Dhanachandra, K. Manglem, and Y. J. Chanu, ''Image segmen-tation using K-means clustering algorithm and subtractive clustering algorithm,'' Proc. Comput. Sci., vol. 54, pp. 764–771, Jan. 2015, doi:10.1016/j.procs.2015.06.090.

[7] N. V. Shree and T. N. R. Kumar, ''Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network,'' Brain Informat., vol. 5, no. 1, pp. 23–30, Mar. 2018, doi:10.1007/s40708-017-0075-5.

[8] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja,''Segmentation, feature extraction, and multiclass brain tumor classification,'' J. Digit. Imag., vol. 26, no. 6, pp. 1141–1150, Dec. 2013, doi:10.1007/s10278-013-9600-0.

[9] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille,''Efficient multilevel brain tumor segmentation with integrated Bayesian model classification,'' IEEE Trans. Med. Imag., vol. 27, no. 5, pp. 629–640,May 2008, doi: 10.1109/TMI.2007.912817.

[10] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, ''Brain tumor classification using convolutional neural net-work,'' in World Congress on Medical Physics and Biomedical Engineering. Singapore: Springer, 2018, pp. 183–189.

[11] A. Pashaei, H. Sajedi, and N. Jazayeri, ''Brain tumor classification via convolutional neural network and extreme learning machines,'' in Proc.8th Int. Conf. Comput. Knowl. Eng. (ICCKE), Oct. 2018, pp. 314–319.

[12] T. Gopi Krishna a, Satyasis Mishra b, Sunita Satapathy c, K. V. N. Sunitha d and Mohamed A. Abdelhadi e,"Classification of Brain Tumor Types on Magnetic Resonance Images Using Hybrid Deep Learning Approach with Radial Basis Function Neural Network",DOI: 10.9734/bpi/ramrcs/v7/1579B.*

[13] Jalluri Gnana SivaSai, P. Naga Srinivasu, Munjila Naga Sindhuri,Kola Rohitha, and Sreesailam Deepika,"An Automated Segmentation of Brain MR Image Through Fuzzy Recurrent Neural Network", https://doi.org/10.1007/978-981-15-5495-7_9.

[14] Brain Tumor Classification (MRI) Dataset.Accessed:Available:https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri?resource=download

[15] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' 2015, arXiv:1502.03167.

[16] Javeria Amin ,Saima Jabeen"A New model for brain tumor detection using ensemble transfer lrarning and quantum Variational Classifier.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9023211