A

MINOR PROJECT REPORT

on

BRAIN TUMOR DETECTION

BE(IT)-IV Sem
By

JHANSI LAVUDYA(160120737125)

SRAVANTHI VEERAMALLA(160120737136)

Under the guidance of

Dr. D.Lakshmi Srinivasa Reddy

Assistant Professor

AI & DS Department

DEPARTMENT OF INFORMATION TECHNOLOGY
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
(Affiliated to Osmania University; Accredited by NBA(AICTE) and NAAC(UGC), ISO Certified 9001:2015)
KOKAPET(V),GANDIPET(M),RR District HYDERABAD - 75
Website: www.cbit.ac.in
2021-2022

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)

Kokapet (Village), Gandipet, Hyderabad, Telangana-500075. www.cbit.ac.in

ISO Certified 9001:2015

COMMITTED TO RESEARCH, INNOVATION AND EDUCATION 43 years

This is to certify that the project work entitled " BRAIN TUMOR DETECTION" submitted to CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY, in partial fulfillment of the requirements for the completion of Mini Project-II of IV Semester B.E. in Information Technology, during the Academic Year 2021-2022, is a record of original work done by **LAVUDYA JHANSI(160120737125)** and **VEERAMALLA SRAVANTHI(160120737136)** during the period of study in the Department of IT, CBIT, HYDERABAD, under our guidance.

Project Guide	**Head of the Department**
Dr.D.Lakshmi Srinivasa Reddy	**Dr.Rajanikanth Aluvalu**
Assistant Professor, Dept. of AI&DS,	Professor, Dept. of IT,
CBIT, Hyderabad.	CBIT, Hyderabad.

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to **Dr.D.Lakshmi Srinivasa Reddy**, our project guide, for her invaluable guidance and constant support, along with his capable instruction and persistent encouragement.

We are grateful to our Head of Department, **Dr.Rajanikanth Aluvalu**, for her steady support and for the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, **Dr.P.Ravinder Reddy**, as well as the Management of the Institute, for having designed an excellent learning atmosphere.

Our thanks are due to all members of the staff and our lab assistants for providing us with the help required to carry out the groundwork of this project.

ABSTRACT

.A Brain tumor is considered as one of the aggressive diseases, among children and adults. Brain tumors account for 85 to 90 percent of all primary Central Nervous System(CNS) tumors. Every year, around 11,700 people are diagnosed with a brain tumor. The 5-year survival rate for people with a cancerous brain or CNS tumor is approximately 34 percent for men and 36 percent for women. Brain Tumors are classified as: Benign Tumor, Malignant Tumor, Pituitary Tumor, etc. Proper treatment, planning, and accurate diagnostics should be implemented to improve the life expectancy of the patients. The best technique to detect brain tumors is Magnetic Resonance Imaging (MRI). A huge amount of image data is generated through the scans. These images are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties.So we will build a brain tumor classifier using CNN(Convolutional Neural Networks),widely used for image classification for its high accuracy.

CONTENTS

S.No	Topics	Page. No
	List of Figures	5
	List of Tables	-
	Abbreviations	6
	Abstract	3
1	**Introduction**	
	1.1 Motivation	1
	1.2 Objective of the Project	1
2	**Literature Survey**	2
3	**Problem Statement**	3
4	**Methodology**	
	4.1 Existing Methodology	4
	4.2 Proposed Methodology	5
	4.3 Design Of The Project	6
5.	**Implementation**	7-11
6	**Results**	12

7.	**Conclusion**	13
8	**Future Scope**	14
9	**Bibliography**	15
10	**List of figures**	16
11	**List of Abbreviations**	17

1. INTRODUCTION

1.1 Motivation

Early Detection: One of the most important reasons for using deep learning techniques, such as CNNs, for brain tumor detection is that it allows for earlier detection of tumors. Early detection is crucial because it allows for earlier treatment, which can significantly improve patient outcomes.

Accuracy: Another important benefit of using CNNs for brain tumor detection is their high accuracy. CNNs are capable of identifying even small tumors with high accuracy, which can help doctors make more informed decisions about treatment options.

Speed: CNNs can analyze large amounts of medical imaging data quickly and accurately, making them a valuable tool for radiologists and other medical professionals. This can help reduce the time it takes to diagnose brain tumors and start treatment, which is particularly important in cases where time is of the essence.

Cost-Effective: Using CNNs for brain tumor detection can be more cost-effective than other diagnostic methods. This is because it can reduce the need for invasive procedures, such as biopsies, and can help doctors make more accurate diagnoses with fewer diagnostic tests.

Improved Patient Outcomes: Ultimately, the main goal of using CNNs for brain tumor detection is to improve patient outcomes. By detecting brain tumors earlier and with greater accuracy, doctors can provide more effective treatment, which can help patients recover more quickly and with fewer complications.

1.2 Problem Statement

Brain tumors are a leading cause of cancer-related deaths, and early detection is critical for improving patient outcomes. However, accurately identifying brain tumors from medical images such asMRI scans can be challenging, and traditional methods can be time-consuming and prone to error. Therefore, there is a need to develop a deep learning-based approach for brain tumor detection that can improve accuracy and speed up the diagnosis process. The aim of this project is to develop a deep learning algorithm that can accurately detect brain tumors from medical images and classify them.

2 . LITERATURE SURVEY

S.No	Name	Publisher	Methodology
1.	Hybrid Brain Tumor Detection Using SVM Classifier and Deep Learning	S.Manasa,M.Aravindan,S. Anand	Paper proposes hybrid approach that combines deep learning and SVM
2	Deep Learning Using convolution Neural Networks	H.Jalalian,M.Mashohor,R. Mahmud,M.Karim	Deep learning based approach for detecting brain tumor from mri images using CNN
3	Brain Tumor Detection using Image Processing and Machine Learning Techniques	T.Kalim	Using Machine learning libraries
4	Brain tumor detection using deep learning	Avigyan Sinha,Aneesh,Malavia Suresh	Used Convolution Neural Networks
5	CNN Model for brain tumor detection	Nawras Al-Ani,Omran AI-Shamma	Implemented AlexNet,VGG-16,GoogleNet, ResNet-50

3.PROBLEM STATEMENT

Brain tumors are a leading cause of cancer-related deaths, and early detection is critical for improving patient outcomes. However, accurately identifying brain tumors from medical images such asMRI scans can be challenging, and traditional methods can be time-consuming and prone to error. Therefore, there is a need to develop a deep learning-based approach for brain tumor detection that can improve accuracy and speed up the diagnosis process. The aim of this project is to develop a deep learning algorithm that can accurately detect brain tumors from medical images and classify them.

4.METHODOLOGY

4.1.Existing Methodology:

1.Image Acquisition:The acquisition of high-quality medical images is critical to the accuracy and reliability of the deep learning algorithm. The images must be of sufficient resolution and contrast to accurately depict the structures of the brain, including any tumors that may be present. Different imaging modalities may be used depending on the specific needs of the patient and the nature of the suspected tumor.

2.Preprocessing:Pre-processing techniques such as normalization, rescaling, and cropping can be used to standardize the images and improve the accuracy of the deep learning algorithm.

3.Region of Interest Extraction:It refers to the process of identifying and isolating the relevant regions in the medical image that contain the tumor and the surrounding tissues for analysis.

4.Feature Selection:the process of selecting the most informative features from the medical images that are relevant to the task of detecting brain tumors. The goal of feature selection is to reduce the dimensionality of the data and remove irrelevant or redundant features, which can improve the accuracy and efficiency of the deep learning algorithm.

5.Feature Extraction:Extracting relevant features from medical images that can be used to accurately detect and classify brain tumors. The goal of feature extraction is to represent the medical images in a way that captures the most important information while reducing noise and irrelevant information.

6.SVM Classifier : SVM can be used to classify medical images as either containing a tumor or not containing a tumor. The SVM algorithm works by finding a hyperplane that maximally separates the two classes, based on the input features. The hyperplane is chosen in such a way that it maximizes the margin between the two classes, i.e., the distance between the hyperplane and the closest data points in each class.

7.Evaluation:A way to measure the performance and accuracy of the model. Evaluation can help to identify any weaknesses in the model and guide the selection of appropriate hyperparameters and feature selection methods.some common evaluation metrics used are Accuracy ,Sensitivity and Specificity,Receiver Operating Characteristic (ROC) curve,Area Under the Curve (AUC) etc.

8.Testing:It allows the performance of the trained model to be evaluated on new and unseen data. Testing is typically performed using a separate set of data, known as the testing dataset, which is distinct from the training dataset used to train the model.

4.2 Proposed Methodology

1.Data Acquisition: The first step is to acquire the medical imaging data, such as MRI or CT scans, from patients with and without brain tumors.

2.Preprocessing: The acquired images are preprocessed to remove any noise and artifacts that may interfere with the analysis. This may involve techniques such as image denoising, image normalization, and intensity normalization.

3.Image Segmentation:It involves isolating the region of interest (ROI) in the medical images that contains the tumor or suspected tumor and it enables accurate identification of the tumor region in the input medical images.

4.Image Augmentation: It increases the size of the training dataset by creating variations of the original images.It reduces overfitting and improves the generalization performance of the models. Some common techniques for image augmentation are like Rotation,Training,scaling,Flipping etc.

5.Convolution Neural Network:CNNs can be used to classify medical images into those that contain a tumor and those that do not, as well as to segment the tumor region from the surrounding healthy tissue. Here are some of the key features of CNNs that make them effective: convolution layer,pooling layers,Activation Function etc.

6.Model Evaluation: it allows us to assess the performance of the trained models and compare them against each other. Here are some common metrics used for evaluating the performance are Accuracy,Precision,Recall,Area under curve etc.

7.Testing:Testing can be performed on a separate set of images that were not used during the training phase. This phase allows us to evaluate the performance of the model on new, unseen data and to validate that the model can generalize well to new images.The Accuracy of 96.5% is calculated.

4.3 Design Of Project

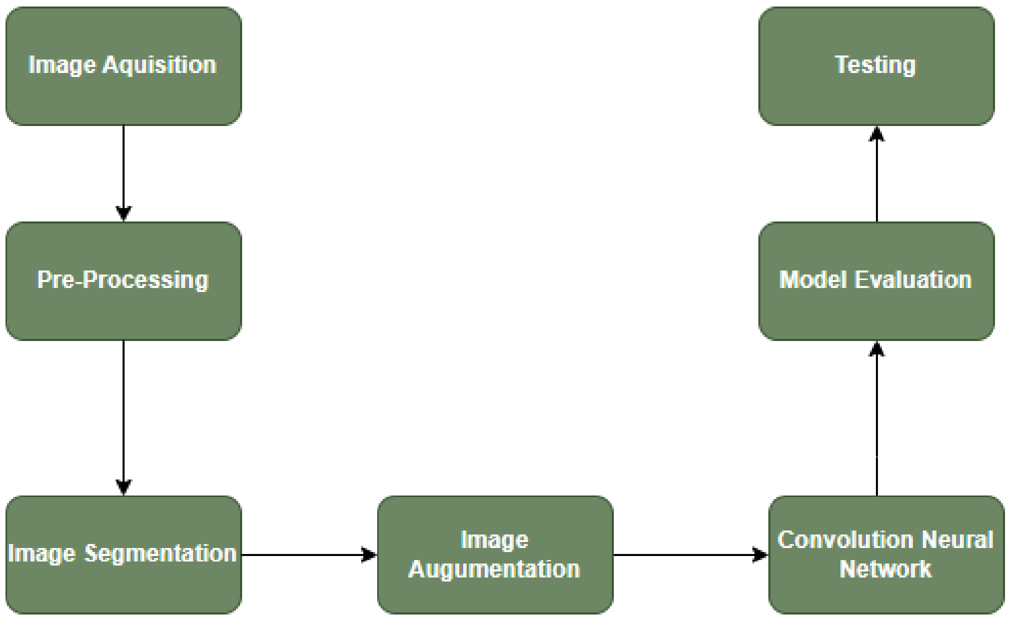


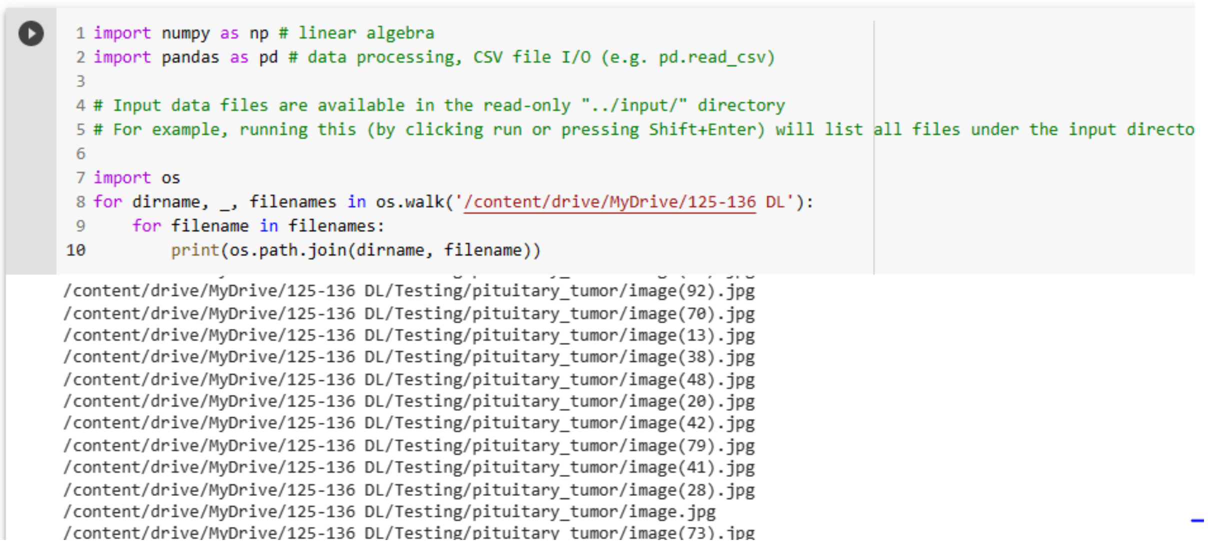
Fig 4.3.1: design of the project

5.IMPLEMENTATION

About the Dataset:

The dataset has two folders testing and training.Each folder has four more sub folders. These folders have MRIs of respective tumor classes.The four folders are: 1.glioma_tumor,2.meningioma_tumor,3.no_tumor,4.pituiatary_tumor.There are totally 3274 images.

Data Collection: The process of gathering relevant data from various sources to be used for training and testing a CNN model. This involves selecting and labeling data that is representative of the problem domain and ensuring that it is large enough to produce meaningful results.

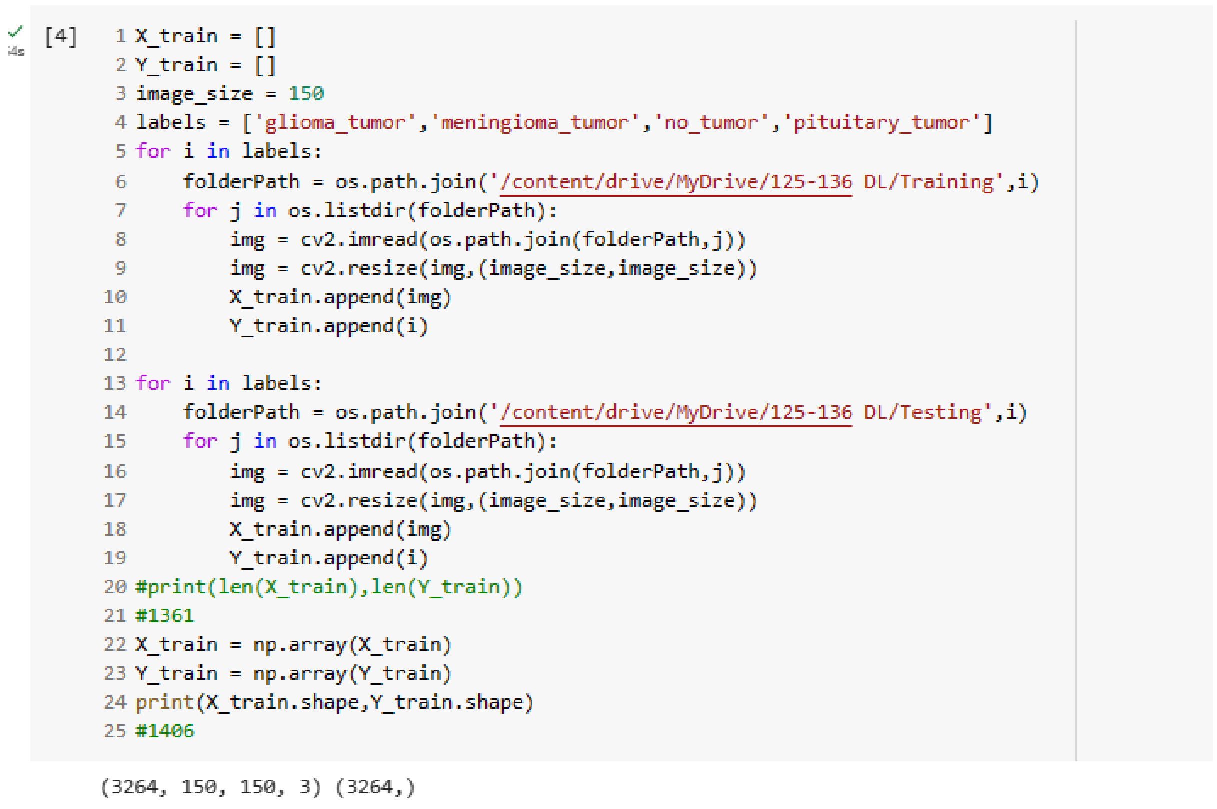


```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directo

import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/125-136 DL'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(92).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(70).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(13).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(38).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(48).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(20).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(42).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(79).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(41).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(28).jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image.jpg
/content/drive/MyDrive/125-136 DL/Testing/pituitary_tumor/image(73).jpg
```

fig 5.1:Data Collection

Data Preprocessing: The process of cleaning, transforming, and organizing raw data to make it suitable for use in a CNN model. This includes operations such as normalization, resizing, and augmentation to improve the quality and quantity of the data.



```
[4]
X_train = []
Y_train = []
image_size = 150
labels = ['glioma_tumor','meningioma_tumor','no_tumor','pituitary_tumor']
for i in labels:
    folderPath = os.path.join('/content/drive/MyDrive/125-136 DL/Training',i)
    for j in os.listdir(folderPath):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        Y_train.append(i)

for i in labels:
    folderPath = os.path.join('/content/drive/MyDrive/125-136 DL/Testing',i)
    for j in os.listdir(folderPath):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        Y_train.append(i)
#print(len(X_train),len(Y_train))
#1361
X_train = np.array(X_train)
Y_train = np.array(Y_train)
print(X_train.shape,Y_train.shape)
#1406
```

```
(3264, 150, 150, 3) (3264,)
```

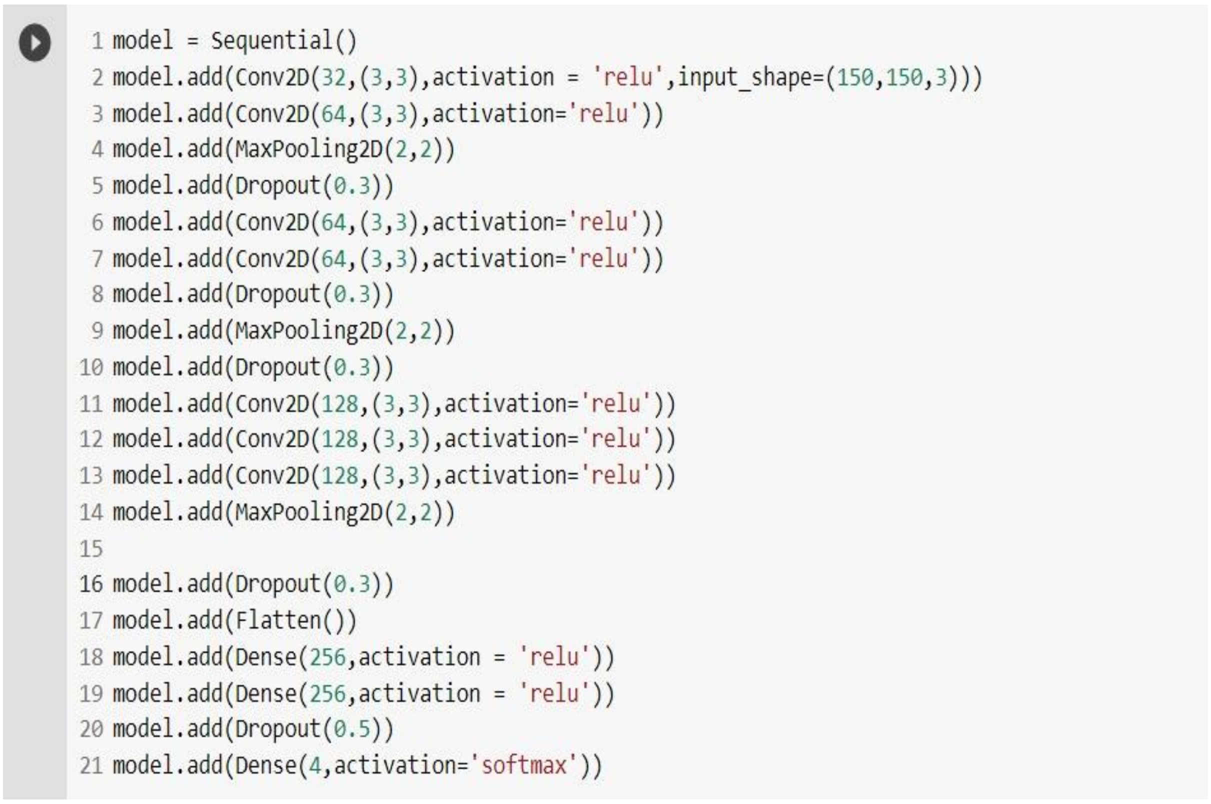
fig 5.2:Data Preprocessing

Building the Model: The process of designing and configuring the architecture of the CNN model. This involves selecting appropriate layers, defining their parameters and connectivity, and specifying the loss function and optimizer to be used during training.

The architecture of our model is:

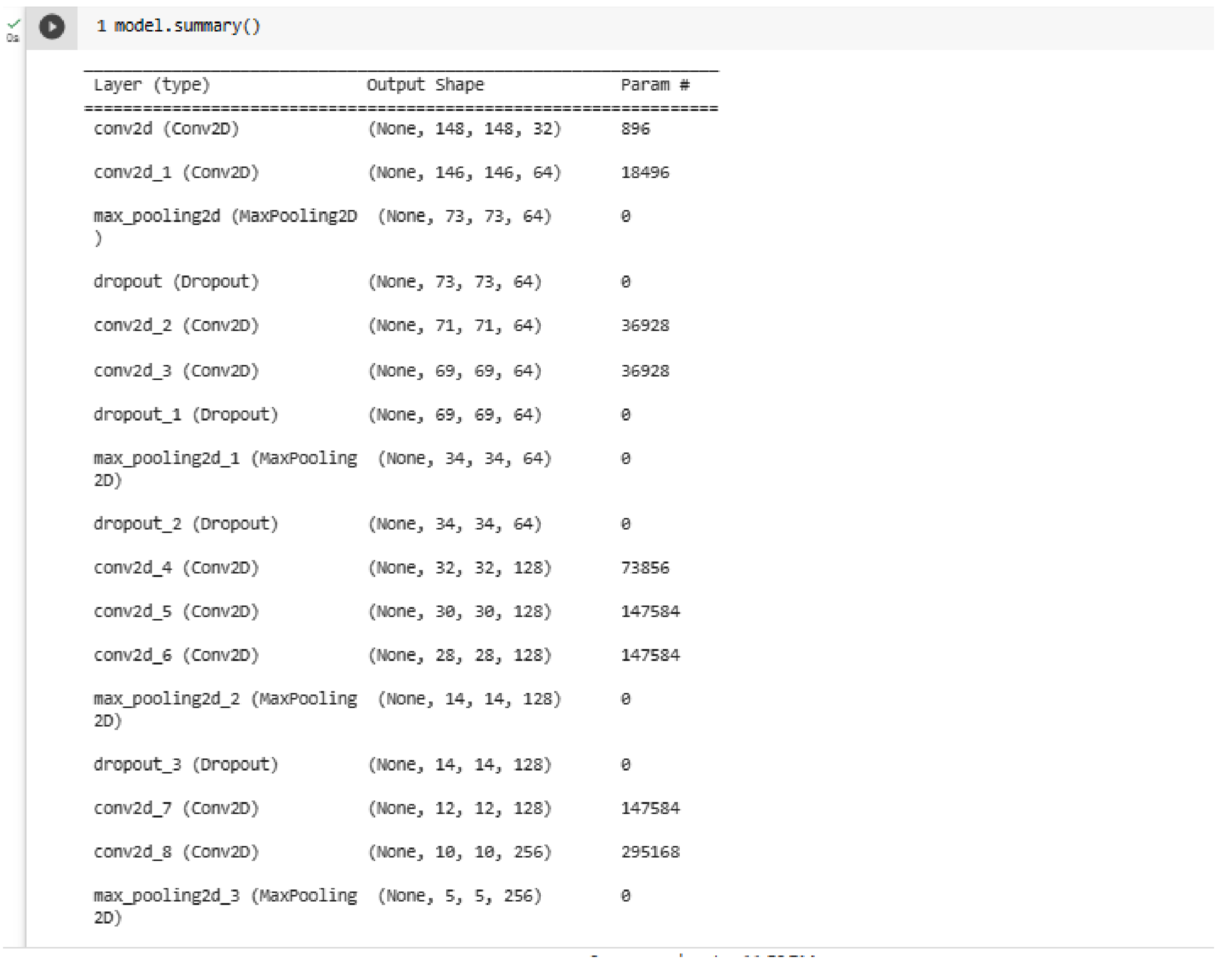
- Conv2D layer (filter=32, kernel_size=(3,3), activation=”relu”)
- Conv2D layer (filter=64, kernel_size=(3,3), activation=”relu”)
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.3)
- Conv2D layer (filter=64, kernel_size=(3,3), activation=”relu”)
- Conv2D layer (filter=64, kernel_size=(3,3), activation=”relu”)
- Dropout layer (rate=0.3)
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.3)
- Conv2D layer (filter=128, kernel_size=(3,3), activation=”relu”)
- Conv2D layer (filter=128, kernel_size=(3,3), activation=”relu”)
- Conv2D layer (filter=128, kernel_size=(3,3), activation=”relu”)
- MaxPool2D layer (pool_size=(2,2))
- Dropout layer (rate=0.3)
- Dense Fully connected layer (256 nodes, activation=”relu”)

- Dense Fully connected layer (256 nodes, activation=”relu”)
- Dropout layer (rate=0.5)
- Dense layer (4 nodes, activation=” softmax”)



```
model = Sequential()
model.add(Conv2D(32,(3,3),activation = 'relu',input_shape=(150,150,3)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Dropout(0.3))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))

model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(256,activation = 'relu'))
model.add(Dense(256,activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(4,activation='softmax'))
```

fig 5.3:Convolution Neural Network

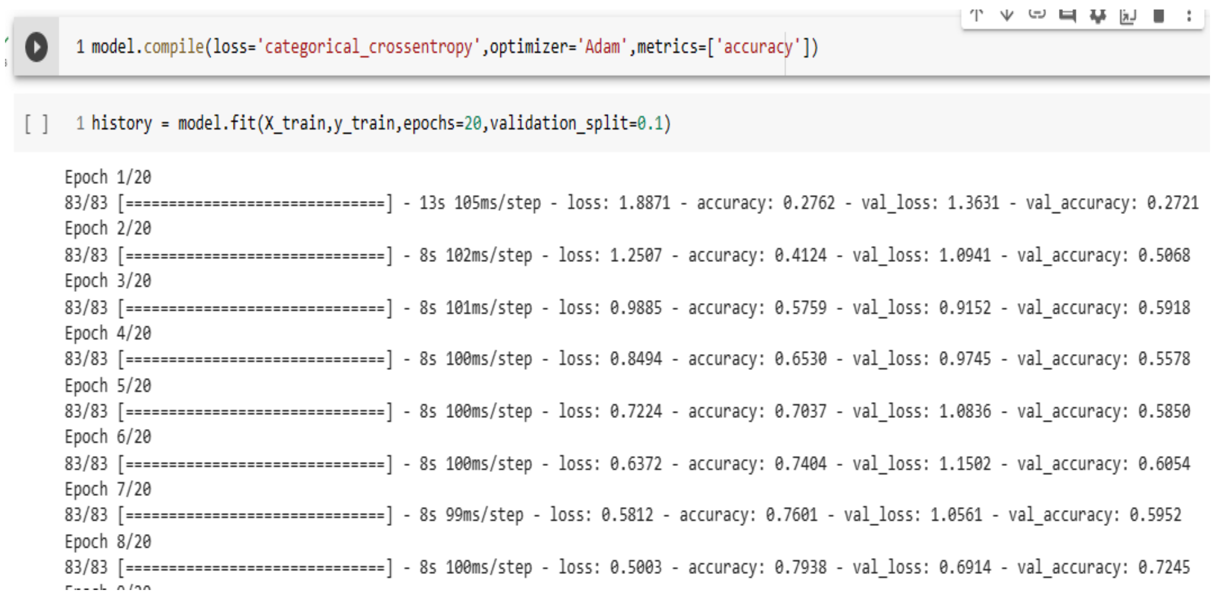


```python
model.summary()
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      896

 conv2d_1 (Conv2D)           (None, 146, 146, 64)      18496

 max_pooling2d (MaxPooling2D  (None, 73, 73, 64)       0
 )

 dropout (Dropout)           (None, 73, 73, 64)        0

 conv2d_2 (Conv2D)           (None, 71, 71, 64)        36928

 conv2d_3 (Conv2D)           (None, 69, 69, 64)        36928

 dropout_1 (Dropout)         (None, 69, 69, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 34, 34, 64)       0
 2D)

 dropout_2 (Dropout)         (None, 34, 34, 64)        0

 conv2d_4 (Conv2D)           (None, 32, 32, 128)       73856

 conv2d_5 (Conv2D)           (None, 30, 30, 128)       147584

 conv2d_6 (Conv2D)           (None, 28, 28, 128)       147584

 max_pooling2d_2 (MaxPooling  (None, 14, 14, 128)      0
 2D)

 dropout_3 (Dropout)         (None, 14, 14, 128)       0

 conv2d_7 (Conv2D)           (None, 12, 12, 128)       147584

 conv2d_8 (Conv2D)           (None, 10, 10, 256)       295168

 max_pooling2d_3 (MaxPooling  (None, 5, 5, 256)        0
 2D)
```

fig 5.4:model summary

Training: The process of optimizing the parameters of the CNN model using the training data. This involves passing batches of input data through the model, computing the loss, and backpropagating the gradients to update the weights.



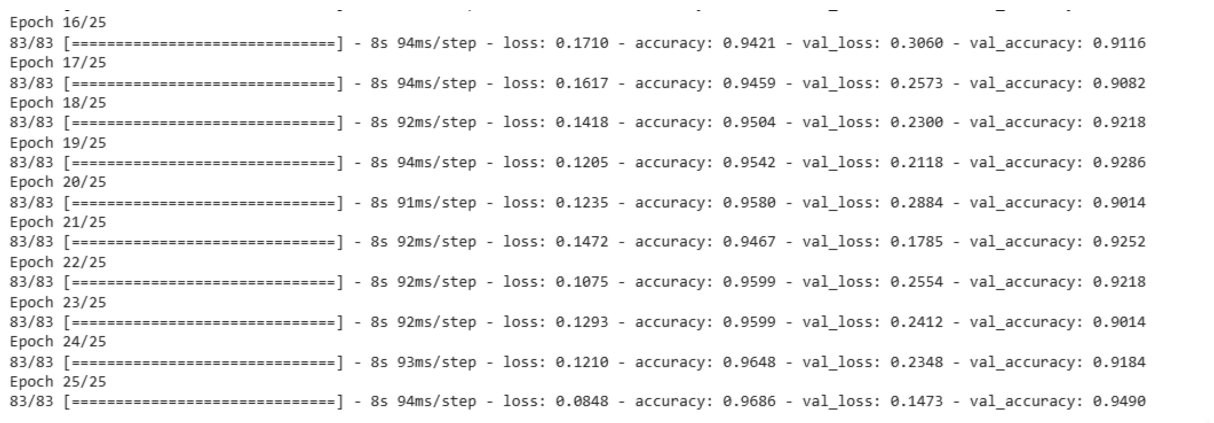
```
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

```
[ ] history = model.fit(X_train,y_train,epochs=20,validation_split=0.1)
```

```
Epoch 1/20
83/83 [==============================] - 13s 105ms/step - loss: 1.8871 - accuracy: 0.2762 - val_loss: 1.3631 - val_accuracy: 0.2721
Epoch 2/20
83/83 [==============================] - 8s 102ms/step - loss: 1.2507 - accuracy: 0.4124 - val_loss: 1.0941 - val_accuracy: 0.5068
Epoch 3/20
83/83 [==============================] - 8s 101ms/step - loss: 0.9885 - accuracy: 0.5759 - val_loss: 0.9152 - val_accuracy: 0.5918
Epoch 4/20
83/83 [==============================] - 8s 100ms/step - loss: 0.8494 - accuracy: 0.6530 - val_loss: 0.9745 - val_accuracy: 0.5578
Epoch 5/20
83/83 [==============================] - 8s 100ms/step - loss: 0.7224 - accuracy: 0.7037 - val_loss: 1.0836 - val_accuracy: 0.5850
Epoch 6/20
83/83 [==============================] - 8s 100ms/step - loss: 0.6372 - accuracy: 0.7404 - val_loss: 1.1502 - val_accuracy: 0.6054
Epoch 7/20
83/83 [==============================] - 8s 99ms/step - loss: 0.5812 - accuracy: 0.7601 - val_loss: 1.0561 - val_accuracy: 0.5952
Epoch 8/20
83/83 [==============================] - 8s 100ms/step - loss: 0.5003 - accuracy: 0.7938 - val_loss: 0.6914 - val_accuracy: 0.7245
```

fig 5.5: training the model

Validation: The process of evaluating the performance of the CNN model on a separate validation dataset during training. This involves monitoring metrics such as accuracy and loss to assess the generalization of the model and avoid overfitting.



```
Epoch 16/25
83/83 [==============================] - 8s 94ms/step - loss: 0.1710 - accuracy: 0.9421 - val_loss: 0.3060 - val_accuracy: 0.9116
Epoch 17/25
83/83 [==============================] - 8s 94ms/step - loss: 0.1617 - accuracy: 0.9459 - val_loss: 0.2573 - val_accuracy: 0.9082
Epoch 18/25
83/83 [==============================] - 8s 92ms/step - loss: 0.1418 - accuracy: 0.9504 - val_loss: 0.2300 - val_accuracy: 0.9218
Epoch 19/25
83/83 [==============================] - 8s 94ms/step - loss: 0.1205 - accuracy: 0.9542 - val_loss: 0.2118 - val_accuracy: 0.9286
Epoch 20/25
83/83 [==============================] - 8s 91ms/step - loss: 0.1235 - accuracy: 0.9580 - val_loss: 0.2884 - val_accuracy: 0.9014
Epoch 21/25
83/83 [==============================] - 8s 92ms/step - loss: 0.1472 - accuracy: 0.9467 - val_loss: 0.1785 - val_accuracy: 0.9252
Epoch 22/25
83/83 [==============================] - 8s 92ms/step - loss: 0.1075 - accuracy: 0.9599 - val_loss: 0.2554 - val_accuracy: 0.9218
Epoch 23/25
83/83 [==============================] - 8s 92ms/step - loss: 0.1293 - accuracy: 0.9599 - val_loss: 0.2412 - val_accuracy: 0.9014
Epoch 24/25
83/83 [==============================] - 8s 93ms/step - loss: 0.1210 - accuracy: 0.9648 - val_loss: 0.2348 - val_accuracy: 0.9184
Epoch 25/25
83/83 [==============================] - 8s 94ms/step - loss: 0.0848 - accuracy: 0.9686 - val_loss: 0.1473 - val_accuracy: 0.9490
```

fig 5.6:validating the model

Testing: The process of evaluating the performance of the CNN model on a separate test dataset after training. This involves passing input data through the model and comparing the predicted outputs to the ground truth labels to compute metrics like accuracy

6.RESULTS

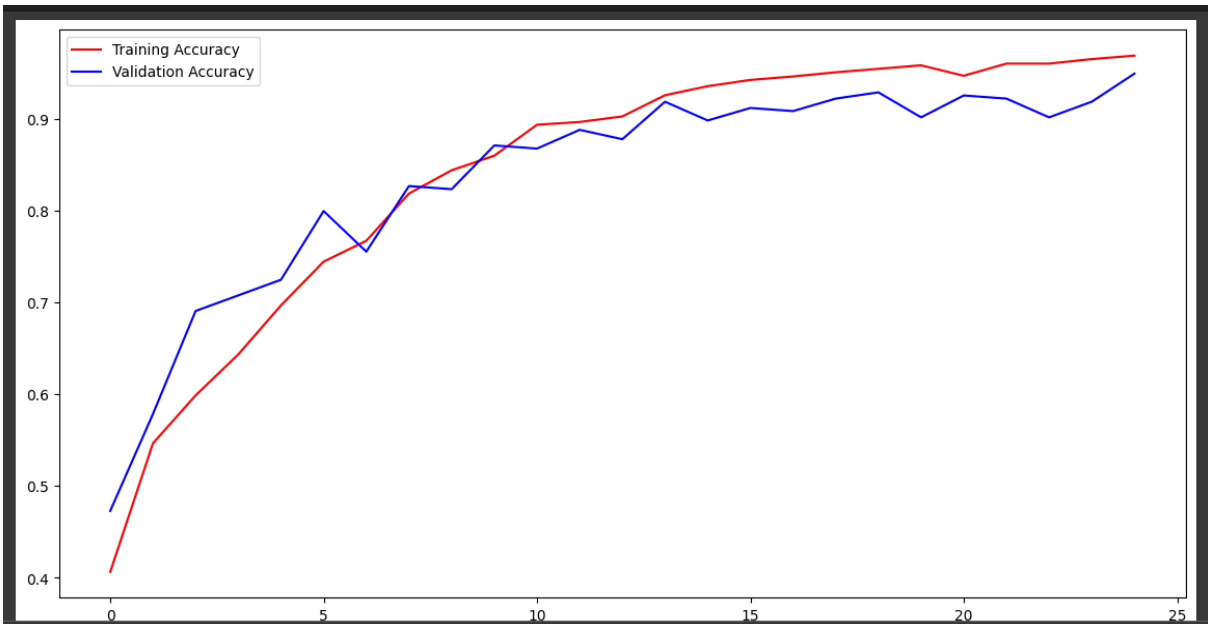


fig 6.1: Accuracy for proposed methodology

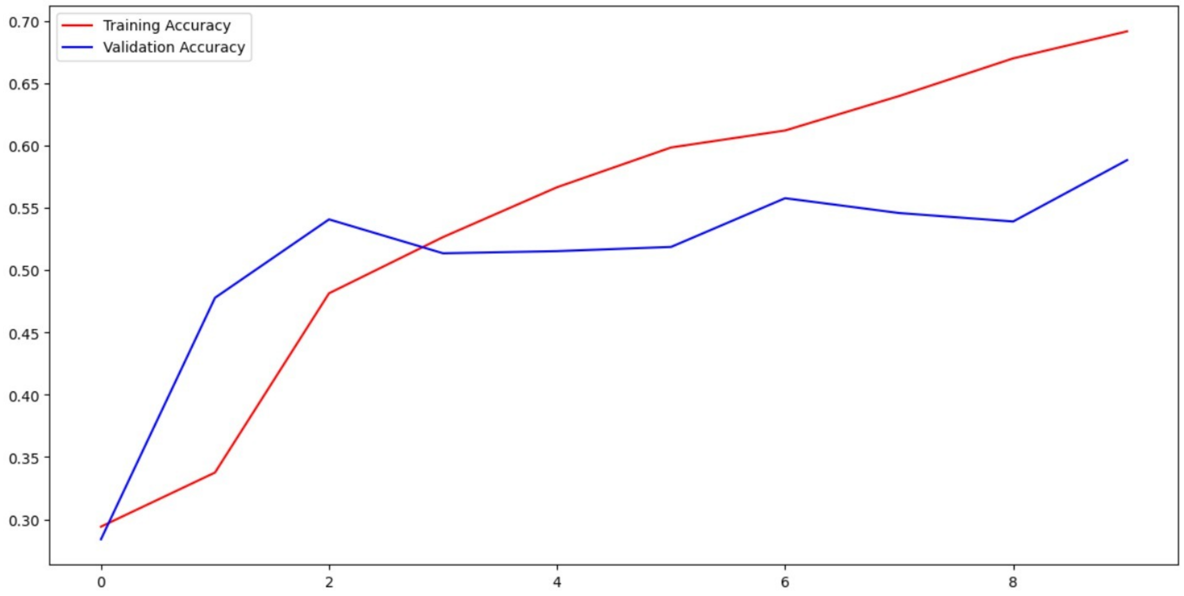


fig 6.2**:** Accuracy in Existing Methodology

7.CONCLUSION

We have developed a new brain tumor segmentation architecture that benefits from the characterization of the four MRI modalities. It means that each modality has unique characteristics to help the network efficiently distinguish between classes. We have demonstrated that working only on a part of the brain image near the tumor tissue allows a CNN model (that is the most popular deep learning architecture) to reach performance close to human observers. Moreover, a simple but efficient cascade CNN model has been proposed to extract both local and global features in two different ways with different sizes of extraction patches. In our method, after extracting the tumor's expected area using a powerful preprocessing approach, those patches are selected to feed the network that their center is located inside this area. Overall, the proposed methodology in brain tumor detection using deep learning has the potential to provide accurate and efficient solutions for the diagnosis and treatment of brain tumors, leading to improved patient outcomes. However, it's important to continue to improve the methodology and validate its performance through further research and testing.

8. FUTURE SCOPE

1.Real-time detection: Deep learning models can be optimized for real-time detection of brain tumors, which can enable faster diagnosis and treatment. This could be especially useful in emergency situations where time is critical.

2.Multimodal image fusion: Deep learning can be used to combine multiple types of medical imaging data, such as MRI, CT, and PET scans, to provide a more comprehensive view of brain tumors. This can improve the accuracy of tumor detection and segmentation.

9.BIBLIOGRAPHY

1.https://www.youtube.com/watch?v=juJYmc4vrWU&feature=youtu.be
2.https://github.com/MohamedAliHabib/Brain-Tumor-Detection/blob/master/Brain%20Tumor%20Detection.ipynb

List of Figures

Figure No	**Name of the Figure**	**Page No**
4.3.1	Design of the project	**6**
5.1	Data preprocessing	7
5.2	Preprocessing	8
5.3	Convolution Neural Network	9
5.4	Model summary	10
5.5	training the model	11
5.6	Validating the model	11
6.1	Accuracy in Proposed methodology	12
6.2	Accuracy in Existing methodology	12

List of Abbreviations

Acronym	**Abbreviation**
CNN	Convolution Neural Network
CNS	Central nervous system
MRI	Magnetic Resonance Images