Brain Tumor Detection Using Deep Learning.

Sudarshan Kumar
School of Computing
DIT University, Dehradun
Uttarakhand, India.
kumaranshu8092@gmail.com

Aparna Agarwal
School of Computing
DIT University, Dehradun
Uttarakhand, India.
agarwalaparna2001@gmail.com

Aditi Raj
School of Computing
DIT University, Dehradun
Uttarakhand, India.
aditiraj0308@gmail.com

Dr Arshad Husain
School of Computing
DIT University, Dehradun
Uttarakhand, India.
arshad.husain@dituniversity.edu.in

ABSTRACT

Early diagnosis and evaluation of a disease's stage is crucial for the successful treatment of any condition. The classification and diagnosis of the disease are crucial for everyone's health. Decision-making, analysis, and measurement are all part of the modern diagnosis. Medical decision assistance relies heavily on computers. Because brain cancer is frequently not discovered until it is too late for a prognosis, it is one of the most challenging fatal diseases to treat. Because of its efficiency and safety, MRI is very narrowly focused. Early diagnosis can greatly influence the prognosis of brain tumors. Brain tumors can be of two different sorts. The first type of tumor is benign, or non-cancerous, and the second type is malignant, or cancerous.The process of developing a treatment plan and evaluating a tumor depends heavily on the classification of the tumor. Because each of the four brain imaging modalities, T1, T1c, T2, and FLAIR, provide unique and significant information about each region of the tumor, many contemporary techniques have used these four modalities.. While many of them showed promise in their segmentation results on the BRATS 2018 dataset, they all have complex structures that require additional training and testing time. In this paper, we have trained the images on the BR35H Brain tumor dataset available on Kaggle. Convolutional Neural Network architecture was proposed to develop a model with better accuracy and loss time than other individual research. We also discuss a two-phase training method that enables us to address issues with the imbalance of tumor labeling. In the first phase, we will be training the images to detect the tumor and record the accuracy of the model. We will further be working on modifying the model to classify the types of tumor that may be present in the brain.

I. INTRODUCTION

In the past years due to significant advancements in the area of AI, machine learning and various deep learning techniques were introduced to help doctors diagnose diseases in the early stage easily. Many computer-aided technologies use Medical Image processing techniques to identify such diseases efficiently and reliably. Brain Tumor is also one of such diseases which needs to be diagnosed in the early stage so that the probability of saving the patient's life increases.The human brain comprises various types of cells and all cells have their own functionality. It receives signals from the body's sensory organs with the help of the nervous system and gives the information to the muscles on what work to do. So if the brain stopped working properly then it will have a very huge impact on the body of a human and can lead to death.In India alone each year, brain tumor are discovered in more than 28,000 individuals and more than 24,000 of them pass away, as stated by the International Association of Cancer Registries (IARC) [19]. According to a different study, brain tumors are responsible for about 5,250 annual deaths in the United Kingdom.

A brain tumor happens when a tissue begins developing abnormally. The brain tumor side effects and signs depend largely on its size, area and rate of development. The brain tumors can either be noncancerous or cancerous. Majorly there are two kinds of brain tumors. One that begins in the brain is known as a primary brain tumor and the other that begins in other body parts and spreads to the brain is known as secondary brain tumors. Primary brain tumors have their name according to the type of cell involved – Gliomas, Meningiomas, Medulloblastomas etc. Gliomas are brain tumors that start in the brain or spinal cord; meningiomas are tumors that start in the membranes that surround the spinal cord and brain; and the most common type of brain tumor in children is medulloblastoma. . It begins in the brain's lower back and spreads throughout the spinal fluid. Brain tumor detection is a very hard and complicated job because tumors generally have different sizes, shapes and locations. Brain tumor diagnosis in the very early stage of a tumor is difficult because it is very hard to accurately measure the size, area and resolution of the brain tumor. Most of the time the reason for brain tumor is unknown but there are many factors which can result in the development of brain tumor such as age, gender, race and ethnicity and many more.

Magnetic resonance imaging (MRI) and computed tomography (CT) scans are the well-known and frequently utilized procedures for detecting brain tumors. MRI uses radio waves and magnets to view objects inside the human body whereas a CT scan is a type of X-ray that involves X-ray machines. An MRI is highly capable at capturing images that help doctors to determine if there is any abnormal behavior in the tissues.It is a widely used non-invasive imaging method that offers sensitive tissue contrast. The capacity of MRI to tolerate tissue that is frequently normalized can allow imaging structures of interest in human brain tumors. When it comes to manually segmenting brain MRI images, researchers have recently encountered a challenging and challenging problem. A classification system must be used to appropriately classify tumors and the locations of those tumors within the brain.Depending on the desired tissue type, multiple MRI modalities such as T1, T2, and Fluid Attenuation Recovery (FLAIR) are utilized to identify brain tissue . Due to its high contrast in soft tissue in humans, MRI pictures provide better and more meaningful data than other imaging techniques such as CT scan in the field of Medical Detection System. Artificial Intelligence is witnessing a significant amount of growth in minimizing the gap between human capabilities and machine capabilities. One of the domains in artificial intelligence, Computer Vision, is also advancing in this area. Brain tumors can be recognised by computer vision whether they are present or absent. Additionally, using patient medical data and machine learning technologies, it is now possible to forecast the likelihood of tumor occurrences in the future. Many works have been done in the area of deep learning particularly over one algorithm – Convolutional Neural Networks. A deep learning system called CNN uses an image as input and prioritizes different parts of the image, allowing it to distinguish between them. CNN is getting recognition in recent times because of its ability to extract features automatically.

II. RELATED WORK

The interpretation and processing of MRI brain tumor images is the most challenging and promising area. A critical step in selecting the appropriate medication for tumor-infected patients at the appropriate time is the use of magnetic resonance imaging (MRI). Different deep learning methods were used for the various studies to automatically segment and find brain tumors. The study looked into how to forecast survival time using deep features that were derived from pre trained CNNs. It offers more proof that domain-specific fine-tuning will enhance performance. The internet offers a standard dataset. Among the many techniques, convolutional neural network (CNN), DL, and neural network-based techniques are the most popular. Many models have been employed in the literature to determine the accurate boundary curves of brain tumors in medical images. This section discusses the various techniques and researches don't in this field. Menze et al. (2014) remarked that over the past few decades, the number of papers on automated brain tumor segmentation has increased dramatically. This finding emphasizes not only the necessity for automatic brain tumor segmentation methods but also demonstrates the ongoing nature of that research. Ramin Ranjbarzadeh [1] used the BRATS 2018 dataset, which is made up of patient clinical data and Multi-Modal MRI imaging and contains a variety of different histological sub-regions with variable levels of prognosis and aggressiveness. To examine the effect of location of brain tumor in four input modalities ,they used a distance wise approach (DWA) and an area-expected approach. Recurrent convolutional neural network (RCNN), a novel deep learning approach suggested by Ramdas Vankdothu [2]. They used a clustering algorithm for segmentation and a gray level co-occurrence algorithm for feature extraction. The suggested approach could identify and classify pathological and normal tissues with 95.17% precision. S. Meenakshi[3] proposed a Deep Neural Network Architecture to classify MRI images using Kaggle Brain Tumor detection 2020 Dataset. They used seven different combinational architectures for classification problems but the proposed DNN - CFIC model gave comparatively higher accuracy. Their work used the classical ANN for classification because there has been a lot of preprocessing done using different units, such as feature extraction and selections, segmentation using a DNN.Amin ul Haq[4] used the brain tumor data set (BTDS) from the general hospital and Nanfang hospital in China in a different study. They employed Transfer Learning (TL) and Data Augmented techniques to improve the predictive accuracy of CNN Model. Sahar Gull [5] used three datasets (BRATS2018, BRATS2019, and BRATS2020) in the proposed model to train and validate it for the most effective identification of brain tumors. They proposed a fully automated methodology for classification and segmentation. Preprocessing, skull stripping, segmentation, post-processing, and classification were all steps in the procedure.On these three datasets, the experimental results of the suggested programme were displayed, and they showed the greatest batch accuracies of 96.50%, 97.92%, and 98.79%. Samuel Teicher [6] proposed that new tumor segmentation algorithms that make use of Random Forests and comparable voxel analysis methods have a strong chance of being adopted by MRI installations in the future. With the help of these algorithms, areas of concern on a scan can be swiftly identified and marked for use in diagnosis. Discriminative models frequently employ a traditional machine-learning pipeline that relies on manually created features. Because the classifier's behavior is independent of the nature of the data, in these methods, the classifier is trained to discriminate between healthy and unhealthy tissues under the assumption that the input features have a high enough discriminative power.[7] This paper used CNN to detect brain tumors in order to include deep learning in their research, and they contrasted the CNN results with those of the traditional method with the highest accuracy (SVM). The best accuracy achieved with a learning rate of 0.001 was 97.87 %. This analysis of the literature demonstrates that there have been many academic papers published on the classification and segmentation of brain tumors.

Some researchers used traditional classifiers, whereas others used different deep learning techniques.Some efforts yielded significant results while using conventional techniques, whereas others did not.But after looking at these studies, we can say that deep learning beats traditional classifiers because it makes better use of network memory and learning mechanisms.

III. PROPOSED METHODOLOGY

1: Proposed 9-layer CNN architecture for the Brain Tumor detection

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 62, 62, 32)	896
activation_10 (Activation)	(None, 62, 62, 32)	0
max_pooling2d_6 (MaxPooling 2D)	(None, 31, 31, 32)	0
conv2d_7 (Conv2D)	(None, 29, 29, 32)	9248
activation_11 (Activation)	(None, 29, 29, 32)	0
max_pooling2d_7 (MaxPooling 2D)	(None, 14, 14, 32)	0
conv2d_8 (Conv2D)	(None, 12, 12, 64)	18496
activation_12 (Activation)	(None, 12, 12, 64)	0
max_pooling2d_8 (MaxPooling 2D)	(None, 6, 6, 64)	0
flatten_2 (Flatten)	(None, 2304)	0
dense_4 (Dense)	(None, 64)	147520
activation_13 (Activation)	(None, 64)	0
dropout_2 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 2)	130
activation_14 (Activation)	(None, 2)	0

Total params: 176,290
Trainable params: 176,290
Non-trainable params: 0

Table – 1 Modified parameter to classify the 2 categories.

Our Dataset has 3000 images in total of which we have done partition 90% and 15% for the training and testing data respectively. Therefore our model is trained on 2700 images and for testing, we have 300 images. The proposed network has many convolution layers of which 3 are conv2D, 2 of which have 32 filters and 1 has 64 filters. All three convolutional layers with 32,32 and 64 filters respectively have 3*3 kernel functions.

The convolutional network uses hierarchical patterns in the image or data and uses small and simple patterns to form more complicated patterns. This hierarchical network forms a connection between all the Convolutional layers, Max-pooling layers and all 5 fully connected layers. The figure shows that the network layers are made up of 3 Conv2D layers and 3 MaxPooling layers. The final MaxPooling layer is then converted into the 1D layer by flattening the functional layer so that further it could be sent to fully connected convolutional layers. To get the classified output we have used the SoftMax Activation function, a total of 64 connected and 5 fully convolution layers were used in this convolutional neural network. In this model, we have used a dropout layer with a 0.5 value to prevent overfitting which is used following the fully connected layer.

In this model, we have used the ReLu Activation function in all the activation except the layer which is fully connected. In the last Fully connected layer we used SoftMax Activation. Adam was used as the optimizer. After 20 epochs, each epoch with 150 iterations with a batch size of 16 and each epoch lasting about 6s our training model was completed.

The learning parameter summary of the proposed model is shown in Table 1, all 4 categories of this network (convolution layer, activation layer, fully connected layer and softmax classifier) of this model are calculated by summing up the value of params in Table 1. The total params are 176,290, where all the parameters are trainable.

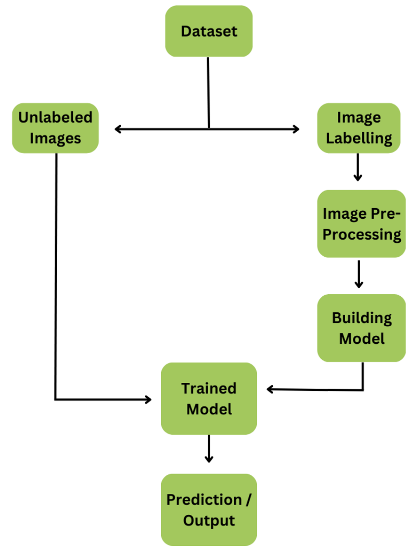


Fig - 9 Layer Model Flowchart

2: Proposed Ensemble deep learning method

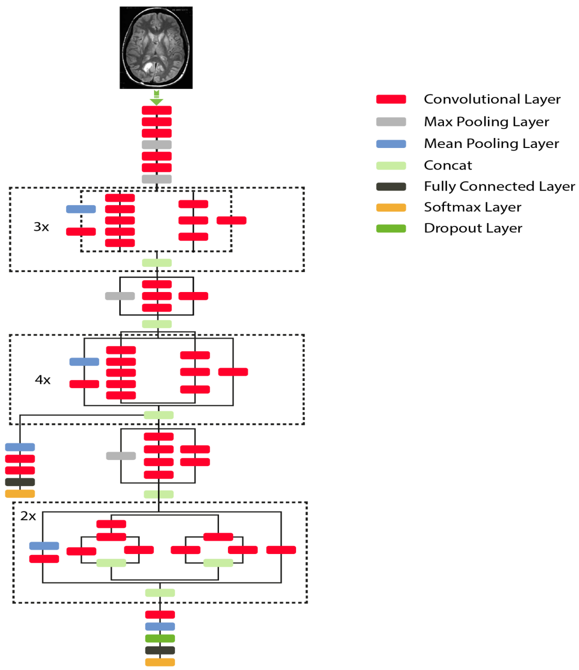
In this model we have used the Ensembled model, in this, we have used 4 pre-trained models that are InceptionV3, ResNet152V2, Xception and DensNet. We have trained on the Google collab with CPU 13.5/15.7 GB of DDR5 RAM and GPU 10.9/15.9 GB on the Tesla K80.

InceptionV3:

Compared to the inception V1 and V2 models, the inception V3 model has 48 layers, which is a little more.InceptionV3 is the advanced version of Inception V1 which gives better accuracy, higher computational speed, less complexity and much more. In this model, there are 3 optimizers used which are SGD, Momentum and RMSDrop which help us achieve better accuracy in less time.

```
Total params: 22,852,898
Trainable params: 1,050,114
Non-trainable params: 21,802,784
```

Fig 10:-Inception V3 Model Summary



- **InceptionV3 architecture**

In this we have passed the pre-processed images and then we trained the model with a given dataset or pre-processed images after which it has given 98.04%,98.22%, 98.21% training, testing and validation accuracy respectively. We trained the model for 100 epochs with a batch size of 32 and each epoch had 65 iterations and each iteration took around 33s.

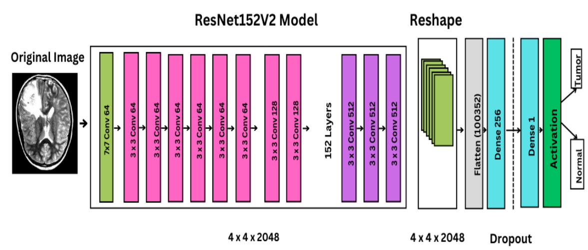
ResNet152V2:

ResNet152V2 is used as a feature extraction model as opposed to the VGG19 - CNN model, as can be seen in the architecture of ResNet152V2. To categorise the image into the right category, this model uses reshape layer, flatten layer, dense layer with 128 neurons, dropout layer, and dense layer with Softmax activation function. The ResNet152V2 has a total of 59,381,762 parameters, that are separated into two categories: trainable params and non-trainable params, which have respective values of 1,050,114 and 58,331,648. The architecture image is provided below with the model summary.

As a pre-trained model, ResNet152V2 provides various benefits, including a faster learning curve and a quicker convergence to high accuracy. It can therefore be added before deep learning models to create models that are effective and trustworthy, leading to improved accuracy.

```
Total params: 59,381,762
Trainable params: 1,050,114
Non-trainable params: 58,331,648
```

Fig 11:-ResNet152V2 Model Summary

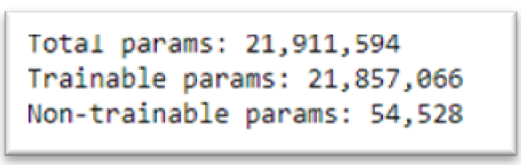


- **ResNet152V2 architecture**

In this we have passed the pre-processed images and then we trained the model with a given dataset or pre-processed images after which it has given 99.57%,98.22% and 98.66% training, testing and validation accuracy respectively. We trained the model for 100 epochs with a batch size of 32 and each epoch had 65 iterations and each iteration took around 47s.

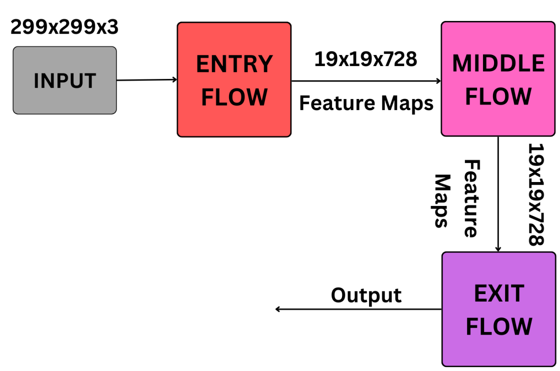
Xception:

The Xception model is also known as the extreme inception as the method applied in the Xception model is the core principle of inception only, which is stretched to its absolute extent. The initial input has been compressed by 1x1 convolutions in the Inception model, and different filter types have been used on each depth space from all of those input spaces. In Xception, only the opposite is true. On the contrary, it applies 1X1 convolution filters individually to each depth map before pulling all of that input space together in one go. That process is almost the same as a depthwise separable convolution technique, which was used in 2014 to build neural networks. There's a further distinction between Inception and Xception. If, after the initial procedure, there is or is not a nonlinearity.



```
Total params: 21,911,594
Trainable params: 21,857,066
Non-trainable params: 54,528
```

Fig 12:-Xception Model Summary



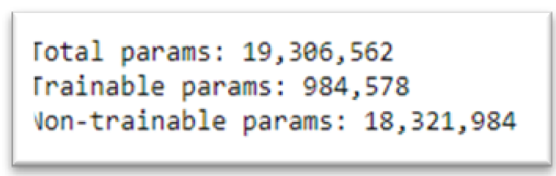
- **Xception architecture**

In this we have passed the pre-processed images and then we trained the model with a given dataset or pre-processed images after which it has given 98.07%,97.33%, 97.77% training, testing and validation accuracy respectively. We have trained the model for 90 epochs with batch size of 32 and each epoch had 65 iterations and each iteration took around 90s.

DenseNet:

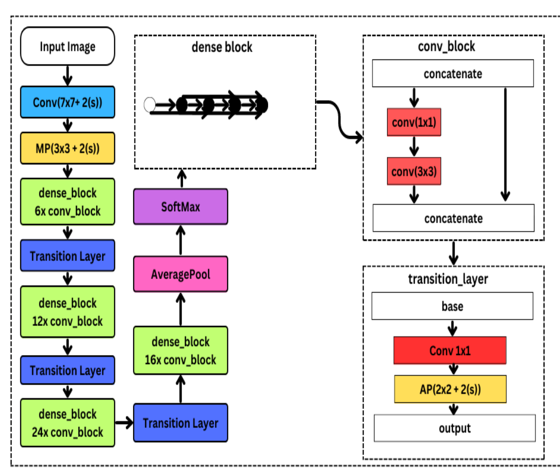
In traditional feed-forward Convolution neural network each convolutional layer receives the output of the previous convolutional layer except the first one and as the layers get deeper and deeper the it causes the problem of some information to vanishing due to this very reason('vanishing gradient') which reduces the efficiency of the model to train.

DenseNets address this issue by adjusting the typical CNN architecture and streamlining the connectivity structure across layers. The term "Densely Connected Convolutional Network" refers to an architecture in which each convolution layer is directly connected to each and every other convolution layer.



```
Total params: 19,306,562
Trainable params: 984,578
Non-trainable params: 18,321,984
```

Fig 13:-DenseNet Model Summary



- **DenseNet architecture**

In this we have passed the pre-processed images and then we trained the model with a given dataset or pre-processed images after which it has given 99.27%,98.66%, 98.88% training, testing and validation accuracy respectively. We trained the model for 90 epochs with batch size of 32 and each epoch had 65 iterations and each iteration took around 49s.

After training all those pre-trained models on our dataset we took them all and made an ensemble to get the higher accuracy we achieved as shown in Table 3.

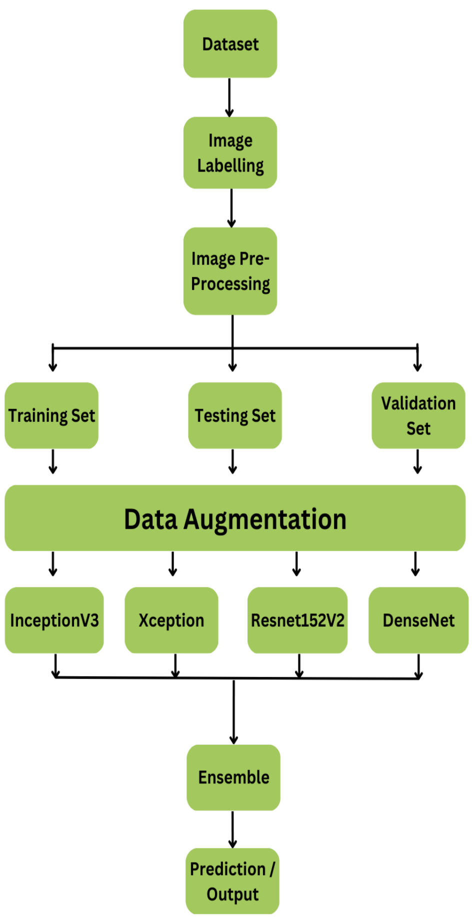


Fig- Ensemble Model FlowChart

IV. CONVOLUTIONAL NEURAL NETWORK

The basic idea of a convolutional Neural Network was introduced by Kunihiko Fukushima in the 1980s. In terms of processing and identifying photos, CNN excels. This structure is composed of convolutional, pooling, and fully connected layers. The fundamental building block of a CNN is its convolutional layers, where filters are applied to extract characteristics from the input image such as edges, textures, and forms. The output of the convolutional layers is then applied to the feature maps after they have been down-sampled, keeping only the most crucial details while decreasing the spatial dimensions. One or more fully connected layers are then applied to the output of the pooling layers to forecast or categorise the image (a convolution layer or a number of convolution layers eliminate distinct features from input by carrying out convolution methods).Each layer may be a set of nonlinear capacities of weighted entireties at assorted orchestrates of spatially close subsets of the past layer's abdicate, which grants the weights to be reused).

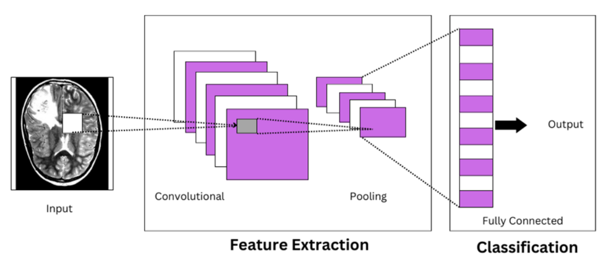


Fig 1-Convolutional Neural Network.

1. CNN VS Simple Neural Network

The architecture of CNNs differs from that of ordinary neural networks.Using a series of hidden layers, ordinary neural networks change an input. Each layer is made up of a group of neurons, and each layer is completely connected to each neuron in the layer before it. Moreover, each layer's neurons operate independently of one another and have no connections between them. A final completely linked layer serves as the output layer, which displays the predictions. To complete images, regular neural networks do not scale well.

CNNs are a little bit unique. The layers are first arranged in width, height, and depth, respectively. Moreover, not all of the neurons in a layer are connected to one another.

Utilizing a previously trained model as the new model for a new task is a technique known as transfer learning. This pre-trained model already knows how to classify images and also has learned general features from images like edges, shapes etc. In this proposed work 4 pre-trained CNN models have been used – InceptionV3, Xception, DenseNet201, and Resnet152V2.

Ensemble learning is a technique of combining different machine learning or deep learning algorithms as sometimes one algorithm is not enough to provide the expected result. Different CNNs can learn different features from the images thus providing different results therefore if we combine all the information obtained from different CNN models and make an ensemble we can get better results as shown through this work.

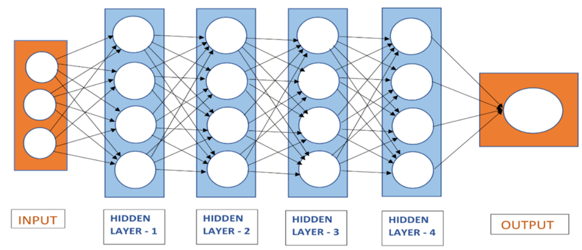


Fig 2:-A Simple Neural Network

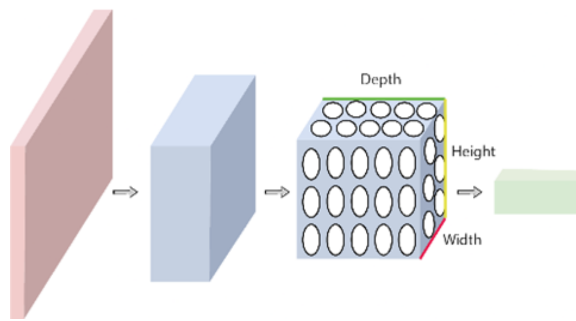


Fig 3:- CNN Background

2. *Layers used to build CNN.*

A fundamental CNN is composed of a number of layers, each of which transforms one volume of activations into another using a differentiable function. Primary layer types include:

- **Input layer:** This is the layer where we supply data for our model. A single image or a series of photos will often constitute the CNN input. The image's original input is found here, with dimensions 32 width,32 height and 3 depth respectively,
- **Convolutional layer:** this extracts the feature from the input dataset. The input

images are processed using a collection of teachable filters known as kernels. Smaller matrices, typically 2 by 2 or 3 by 3, or 5 by 5, make up the filters/kernelsThe dot product between the kernel weight and the pertinent input image patch is calculated as it advances over the input image data. Maps with features are what this layer produces.

- **Activation Layer:** Activation layers introduce nonlinearity to the network by adding an activation function to the output of the preceding layer.It will use the output of the convolution layer as the input for an element-wise activation function. Activation methods that are frequently used include Tanh, Leaky RELU, max(0, x), and RELU. Since the volume is unchanged, the output volume's measurements are 32 x 32 x 12.
- **Pooling layer:** The primary purpose of this layer, which is periodically added to CNNs, is to decrease the volume, which speeds up computation, uses less memory, and prevents overfitting.
- **Flattening layer:** The generated feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they may be passed into a fully connected layer for regression or categorization.
- **Fully Connected Layers:** It computes the final classification or regression task using the input from the preceding layer.
- **Output Layer:** The output from the fully connected layers is then fed into a logistic function for classification tasks, which converts the output of each class into the probability score of each class.

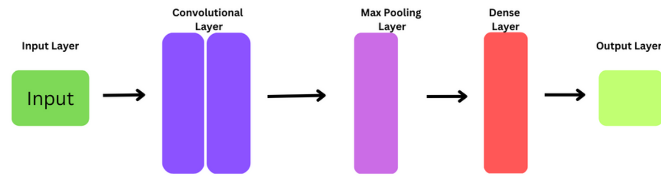


Fig 4:- CNN Layers

3. *Convolution Neural Network Architecture:*

Multiple layers make up a convolutional neural network. Filters are applied by the Convolutional layer to the input image to extract features, the image is down-sampled by the Pooling layer to reduce computation, and the fully connected layer offers the final prediction. Via gradient descent and backpropagation, the network learns the best filters.

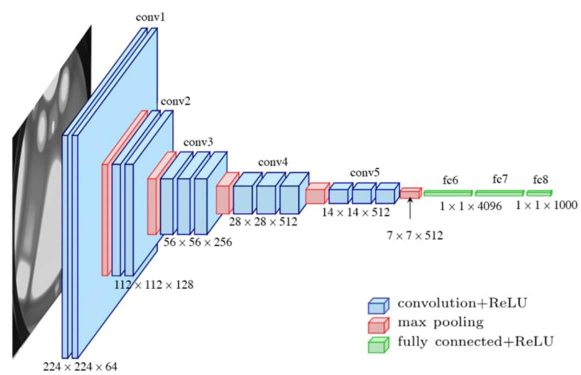


Fig 5:- CNN Architecture

V: OUR CONVOLUTIONAL NEURAL NETWORK (CNN) APPROACH

The objective of this work is to detect and classify brain tumors. In order to achieve this, we are going to utilize Convolution Neural

Networks (CNNs) to form an outfit of deep learning highlights. Models will examine pictures one at a time and recognize the symbol sort. The MRI-based brain tumor pictures will be utilized for preparing and testing reasons. Finally, we will plan a GUI (graphical client interface) website using React and FastApi. Often, brain tumors are mass-like developments of irregular cells within the brain.Some brain tumors are malignant (cancerous), whereas others are noncancerous (kind).

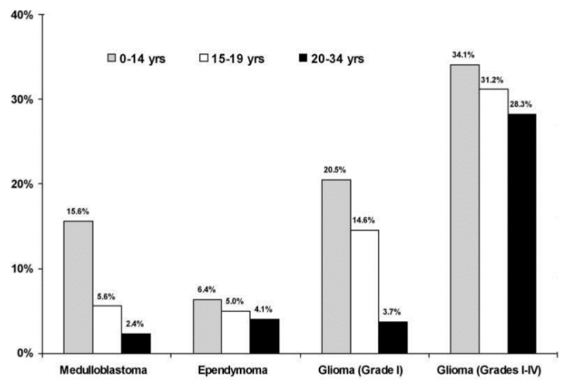


Fig 6:– Distribution of brain tumor types by age

Dissemination of tumors and tumor area by age is an interesting angle of a brain tumor in the study of disease transmission. Certain sorts, such as medulloblastoma, ependymoma, and pilocytic astrocytoma show less increase as they age as shown in the figure above. It is imperative to note that in spite of the huge contrasts in extent between ependymomas and medulloblastomas, spinal ependymomas are among the foremost common among the exceptionally youthful, the pre-adult, and within grown-up, respectively.

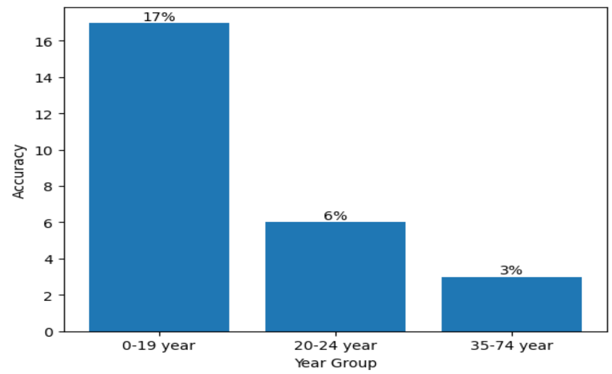


Fig 7:-Proportion of brain tumors involving the cerebellum by age group

As we can see from the above graph, mostly brain tumors occur in children or teenagers so we need to classify them as soon as possible so that treatment can be done on them and we can stop the further growth of the tumor. There are an assessed 20,500 fundamental brain tumors analyzed each year inside the US: 3750 cases happen in individuals age.

1. Architecture of the Model

The convolutional layer serves as the foundation of a CNN architecture . The parameters of its many filters (or kernels) must be learned during training. The size of the filters is typically less than that of the original image. An activation map is created after each filter convolves with the image. A convolution layer modifies the input image in order to extract information from it. During this transformation, the picture is convolved with a kernel (or filter). A hierarchy of features can be created by stacking various layers on top of one another. Each layer extracts the features from its top layers and adds them to the hierarchy to which it is connected.From a stack of input planes as its input, a single convolutional layer produces a number of output planes or feature maps. Filters are applied by the Convolutional layer to the input image to extract features, the image is down-sampled by the Pooling layer to reduce computation, and the fully connected layer offers the final prediction. Via gradient descent and backpropagation, the network learns the best filters.

2. *Functional description of the model*

In this study we have first imported our dataset and library.We read the images and converted them from BGR to RGB. The images present in the dataset were of all different sizes so we converted them to (224,224,1). We have used 3000 images in total. We used a Numpy array to convert the image dataset to an array. We divided our dataset into two categories: trained and tested in a ratio of 70 and 30 percent, where 70% of the images were kept for training and 30% was kept for testing. We normalized the dataset and imported a sequential model (conv2D for generating different layers.)

In this section we will talk about three main topics regarding this project which are :

- Image pre-processing.
- Feature extraction using an ensemble of Convolutional Neural Networks (CNNs)
- ML classifier for brain tumor classification.

2.1 Image pre-processing

The images in the dataset are of different sizes so we have converted them to (224, 244) images size after this we applied Gaussian Blur on the images for the removal of noise from the images and converted all the images to RGB images. We have also done Image data augmentation.

```python
## Image Data Augmentation

train_datagen = ImageDataGenerator(rescale=1/255,
                                   width_shift_range=0.1,
                                   height_shift_range=0.1,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   brightness_range=[0.3,1.5],
                                   horizontal_flip=True,
                                   vertical_flip=True)

val_datagen = image.ImageDataGenerator(rescale = 1./255)

test_datagen = image.ImageDataGenerator(rescale = 1./255)

TRAIN_DIR = BrainTumor_Splited_Data_dir + "/train"
VAL_DIR = BrainTumor_Splited_Data_dir + "/val"
TEST_DIR = BrainTumor_Splited_Data_dir + "/test"
```

Fig 8:-Image Data Augmentation

2.2 Feature extraction using an ensemble of CNNs.

Convolutional Neural Networks have gained a significant amount of attention in the last few years as they can automatically extract features and outperforms any other feature extraction technique when the amount of data is huge. Convolutional channels are applied to the input by the convolutional layer of a CNN in order to compute the amount of neurons that are connected to the input's areas.

$$\begin{array}{|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0\\\hline 0&1&0&0&0&1&0\\\hline 0&0&0&0&0&0&0\\\hline 0&0&0&1&0&0&0\\\hline 0&1&0&0&0&1&0\\\hline 0&0&1&1&1&0&0\\\hline 0&0&0&0&0&0&0\\\hline\end{array} \otimes \begin{array}{|c|c|c|}\hline 0&0&1\\\hline 1&0&0\\\hline 0&1&1\\\hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 0&1&0&0&0\\\hline 0&1&1&1&0\\\hline 1&0&1&2&1\\\hline 1&4&2&1&0\\\hline 0&0&1&2&1\\\hline\end{array}$$

Fig 9:- CNN Feature Extraction

CNN is by and expansive comprised of mainly three units that are being used to extract features from the data then comes the max-pooling or globalaveragepooling2d layer which is being used for dimensionality reduction and then comes the fully connected artificial neural network layer which then uses the extracted features and acts as a classifier to classify the images.

In this study, we adopted a CNN-based methodology, wherein convolutional layers are utilized to extract features from the images, and the features then sent onto a fully connected neural network that determines the image's class using the features gathered. The pre-trained CNN models used are InceptionV3, Resnet152V2, Densenet201, and Xception.

2.3. Fully Connected Neural Network as a classifier:

Before applying the GlobalAveragePooling2D layer before the classification layers, the retrieved features from the CNNs are first fed into a fully connected NN.

In the classification hidden layer, we employed the RELU activation function, and in the output layer, the softmax activation.The loss function used in this is Adam and the learning rate used is 0.001.

VI. EXPERIMENTS AND RESULTS

Dataset:

Dataset that we have used contains Brain Tumor and Non-Tumor MRI scanned images which are collected from multiple sources like Kaggle, git repositories and Radiopaedia which contains real cases of patients. And the one dataset we used is Br35H which contains 1500 non – Tumor images and 1500 Tumor images which we further divided into Training Dataset, Test Dataset and Validation Datasets 70% 15%, 15% respectively.

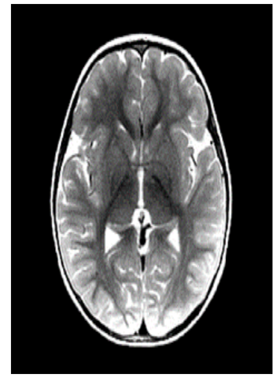
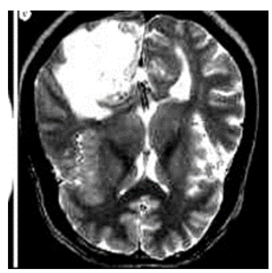
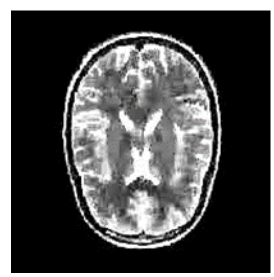
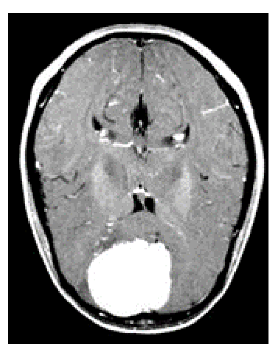
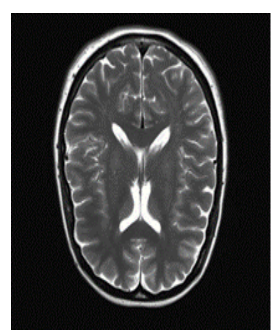
Brain Tumor MRI Images	Normal Brain MRI Images
	
	
	
	

Table 2:- Brain Tumor MRI Images vs Normal Brain MRI Images

In this project, deep neural networks are used to automatically detect brain tumors (CNNs).We have performed this project in python language on jupyter notebook and google colab in which we were assigned Tesla K80 GPU.(After getting the stable accuracy we have compared it with the various Machine Learning(ML) algorithms like KNN, SVM, etc. The suggested brain tumor classification method's effectiveness is determined by taking notice of the training accuracy, validation accuracy, validation loss, and training loss.SVM, KNN, and other ML-based classification methods are used in the current strategies to detect brain tumors. The feature extraction output is required. The classification result is generated based on the feature value we extract, and the accuracy is also determined based on that. The time taken is high, and accuracy is low compared to DNN in SVM-based Tumor detection as shown in the bar graph below.

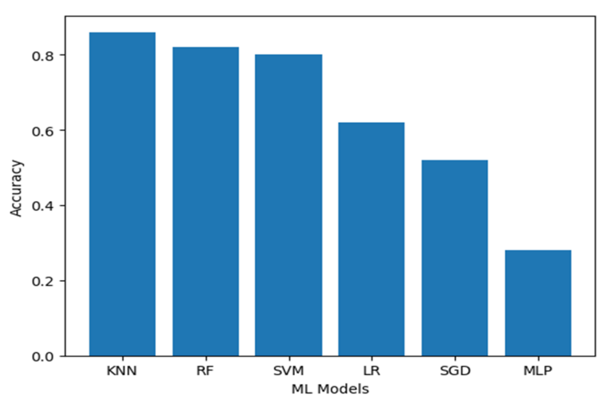


Fig 14:-Comparison Graph

In our work we have trained two models, one with 9-layer architecture and another one is using the Ensemble Model in which we trained 4 pre-build architectures and fed our data to them. We have used Transfer Learning(TL) through which we have achieved the accuracy of 98.04 training accuracy,98.21 % validation accuracy for InceptionV3 and 99.57% training accuracy, 98.66% validation accuracy for Resnet152V2 and 98.07% training accuracy, 97.768% validation accuracy for Xception and 99.27% training accuracy, 98.88% validation accuracy for DenseNet models of which we made ensemble model which have given us the highest accuracy result as 98.66% training, 99.55% validation and 99.95.00% testing accuracy.

The accuracy we achieved by our proposed 9-layer architecture is 99.75% training accuracy and 98.50% validation accuracy as mentioned in the Table no 3 -

Model Name	Training Accuracy	Testing Accuracy	Validation Accuracy
Inception V3	98.04%	98.22%	98.21%
Resnet15 2V2	99.57%	98.22%	98.66%
Xception	98.07%	97.33%	97.77%
DenseNet	99.27%	98.66%	98.88%
Ensemble	98.66%	99.95%	99.55%
9-Layer CNN	99.75%	98.40%	98.50%

Table 3:-Training,Testing and Validation Accuracy of various models

VI.I Our Proposed Ensemble Model:

Accuracy and loss chart in which we have trained on the google collab with CPU 13.5/15.7 GB of DDR5 RAM and GPU 10.9/15.9 GB on

Tesla K80 In this we have trained all the 4 pre-trained models with our data which are:

1. InceptionV3 which we ran for 100 epochs with batch size of 32 and each epoch had 65 iteration and each iteration took around 33s with loss: 0.0561 - acc: 0.9826 - val_loss: 0.0829 - val_acc: 0.9777.

2. Resnet152V2 which we ran for 100 epochs with batch size of 32 and each epochs had 65 iterations and each iteration took around 47s with loss: 0.0381 - acc: 0.9874 - val_loss: 0.0696 - val_acc: 0.9888.

3. Xceptions which we ran for 90 epochs with batch size of 32 and each epoch had 65 iterations and each iteration took around 90s with loss: 0.0393 - acc: 0.9860 - val_loss: 0.1263 - val_acc: 0.9754.

4. DenseNet which we ran for 90 epochs with batch size of 32 and each epochs had 65 iterations and each iteration took around 49s with loss: 0.0285 - acc: 0.9898 - val_loss: 0.0673 - val_acc: 0.9866.

After training all these models we assembled them and took the best of all accuracy from which we got 98.66% training accuracy, 99.95.0% testing accuracy and 99.55% validation accuracy.

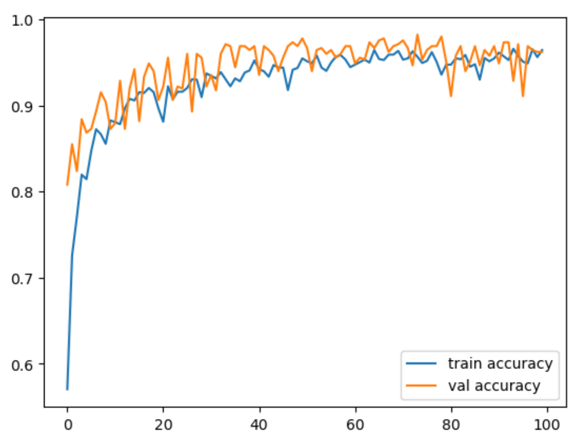


Fig 15:-Inception accuracy

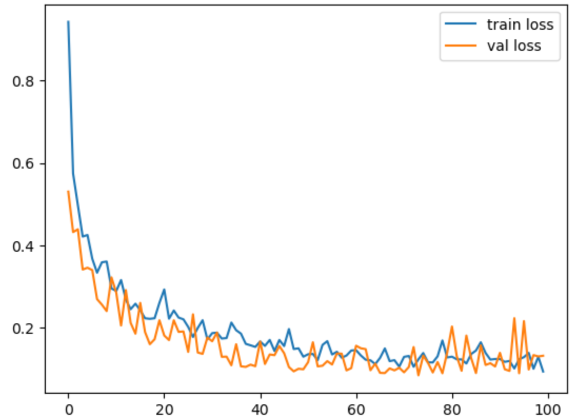


Fig 16:-Inception Loss

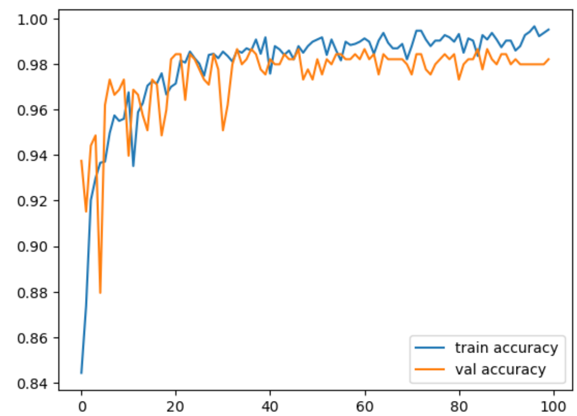


Fig 17:-Resnet Accuracy

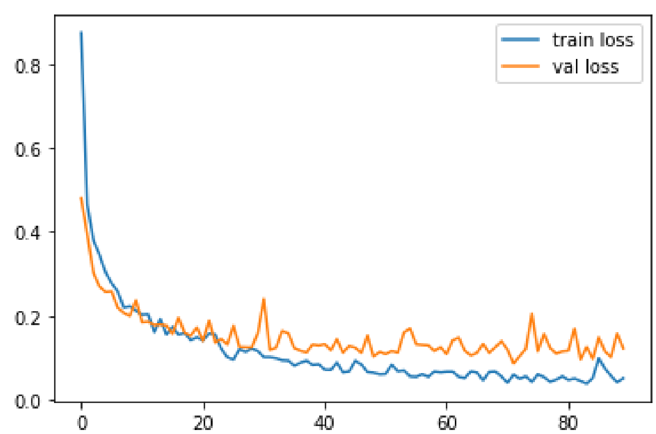


Fig 18:-Resnet Loss

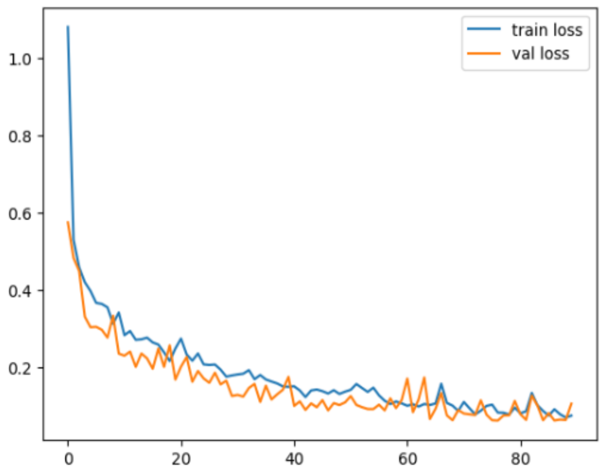


Fig 19:- Xception loss

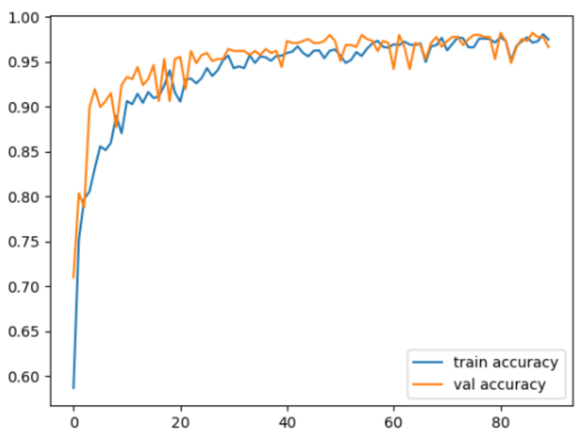


Fig 20:-Xception Accuracy

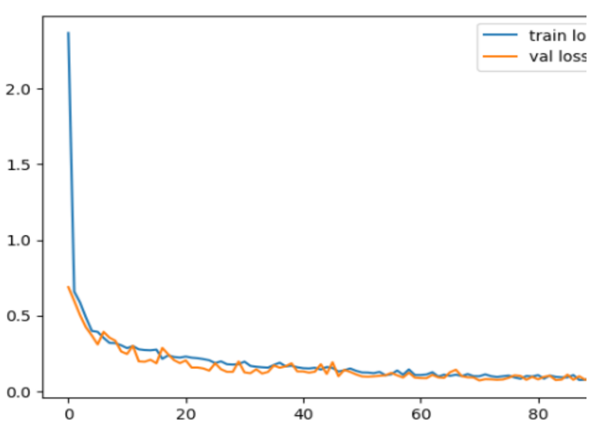


Fig:- 21 DenseNet Loss

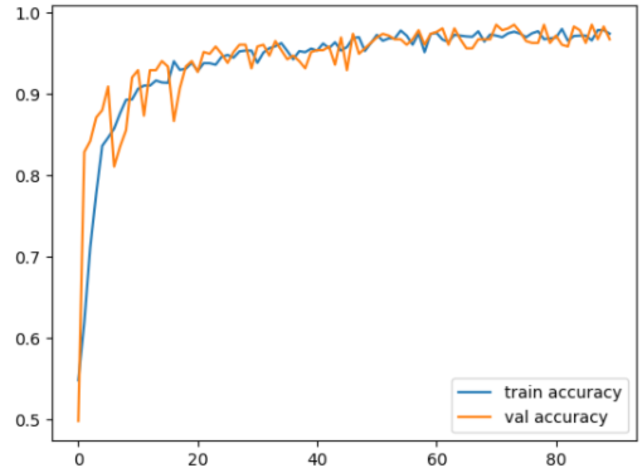


Fig 22:-DenseNet accuracy

VI.II Our Proposed 9-Layer Model:

It has an accuracy and loss chart as shown in the graph which has an acc. of 98.33% and in this model we have included many layers of convolution which involves two Conv2D layers with 32 filters and one Con2D with 64 filters. All three Conv2D layers with 32,32 and 64 filters have the same kernel function size which is 3x3.

The convolutional network uses hierarchical patterns in the image or data and uses small and simple patterns to form more complicated patterns. This hierarchical network forms a connection between all the Convolutional layers, Max-pooling layers and all 5 fully connected layers. The figure shows that the network layers are made up of 3 Conv2D layers and 3 MaxPooling layers. The final MaxPooling layer is then converted into the 1D layer by flattening the functional layer so that further it could be sent to fully connected convolutional layers. To get the classified output we have used the SoftMax Activation function, a total 64 connected and 5 fully convolution layers were used in this convolutional neural network. In this model we have used a dropout layer with a 0.5 value to prevent overfitting which we trained for 20 epochs with a batch size of 16 and each epoch were having 150 iterations with loss: 0.0096 - acc: 0.9975 - val_loss: 0.1257 - val_acc: 0.9850, which we trained on local CPU with 5.9/7.9GB space of DDR5 RAM and GPU(Intel UHD Graphics 630) and used Jupyter Notebook for the code writing and model training using python language.

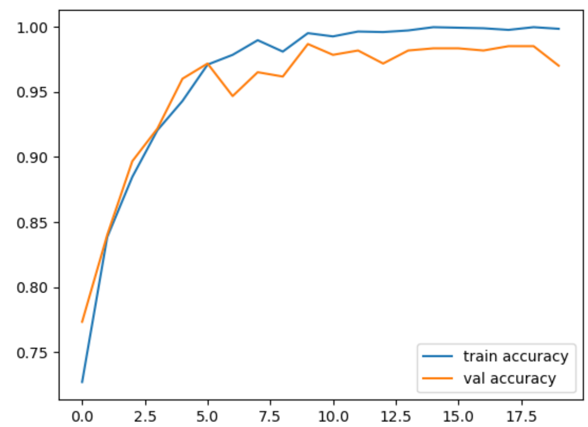


Fig 24:-9-Layer CNN Accuracy

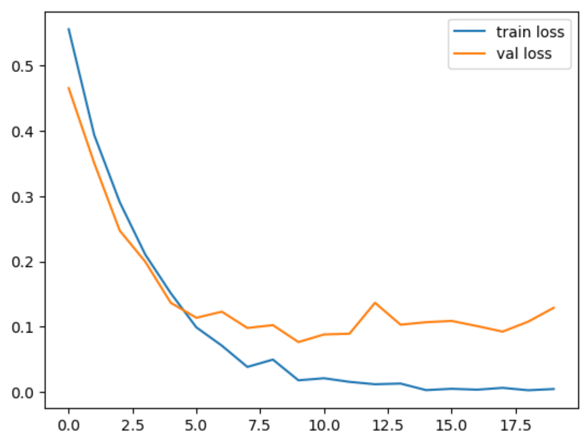


Fig 23:-9-Layer CNN Loss

VII. CONCLUSION

In this cutting-edge world, full of incredible advances, brain tumors could be an exceptionally unsafe malady additionally exceptionally common among individuals. According to statistics from 2018, brain tumors were the 10th most prevalent type of tumor among Indians. This demonstration aims to classify brain tumors to drop the passing rate in India due to brain tumors. This framework will tell the individuals whether they have a brain tumor or not.

Our project consisted of three main steps which are image preprocessing, feature extraction, and then using a classifier for brain tumor classification.The first step includes conversion of the images present in the dataset to (224, 244) images size. After this, we applied Gaussian Blur on the images to remove noise from the images and converted all the images to RGB images.For the next step,we have adopted a CNN-based strategy, features are extracted from the images, and the features are then sent onto a fully connected neural network that classifies the image's class. The pre-trained CNN models utilized in this wander are InceptionV3, Resnet 152V2, Densenet201, and Xception.After this,the features extracted from the CNNs are then fed into the fully connected neural network but before that, we have applied the GlobalAveragePooling2D layer before the classification layers. We used the RELU activation function in the classification hidden layer and softmax activation in the output layer. The loss function used in this is Adam and the learning rate used is 0.001.

We trained our model on a lot of pre-trained models like DenseNet, Exception, and Inception, and then used an Ensemble to improve our accuracy. We achieved 99.95% testing accuracy, 98.66% training accuracy, and 99.55% validation accuracy respectively.

VIII. REFERENCES

1. Malika Bendechache, Maryam Naseri: Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images(2021)
2. Ramdas Vankdothu, Mohd Abdul Hameed: Brain tumor MRI images identification and classification based on the recurrent convolutional neural network(2022)
3. Ketan Kotecha, S. Meenakshi: MRI Brain Tumor Image Classification Using a Combined Feature and Image –Based Classifier(2022)
4. Amin ul Haq, Jian Ping Li: Deep Learning Approach for classification of Brain Tumors using MRI data in the IoT Healthcare department.(2022)
5. Shahzad Akbar, Sahar Gull: Automated Detection of Brain Tumor through Magnetic Resonance Images Using Convolutional Neural Network(2021)
6. Samuel Teicher, Alexander Martinez: Diagnosing and Segmenting Brain Tumors and Phenotypes using MRI Scans(2014)
7. Tonmoy Hussain, Mohsena Ashraf: Brain Tumor Detection using Convolutional Network(2019)
8. N. Varuna Shree and T. N. R. Kumar: Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network(2018)
9. Mohammad Havaei,Axel Dav: Brain Tumor Segmentation with Deep Neural Networks(2017)
10. Md. Saikat Islam Khan, Anichur Rahman: Accurate brain tumor detection using deep convolutional neural network(2022)
11. Tustison, Nicholas J: Advanced Normalization Tools(2009)
12. Narmada M. Balasooriya, Ruwan D. Nawarathna: A sophisticated convolutional neural network model for brain tumor classification(2017)
13. Dinggang Shen, Guorong Wu, Heung -1l Suk: Deep Learning in Medical Image Analysis (2017)
14. Narmada M. Balasooriya, Ruwan D. Nawarathna: A sophisticated convolutional neural network model for brain tumor classification(2017)
15. CDHO Advisory Brain Tumors(2019)
16. Shrutika Santosh Hukkar: Implementation Of image processing for detection of brain tumors(2017)
17. Milica M. Badža ,Marko Č. Barjaktarović: Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network(2020)
18. Hassan Ali Khan1, Wu Jue1: Brain Tumor Classification in MRI image using Convolutional Neural Network(2020)
19. Amit Kumar, Prateek Kumar : A CNN Based Efficient Brain Tumor Detection using MRI (2020)