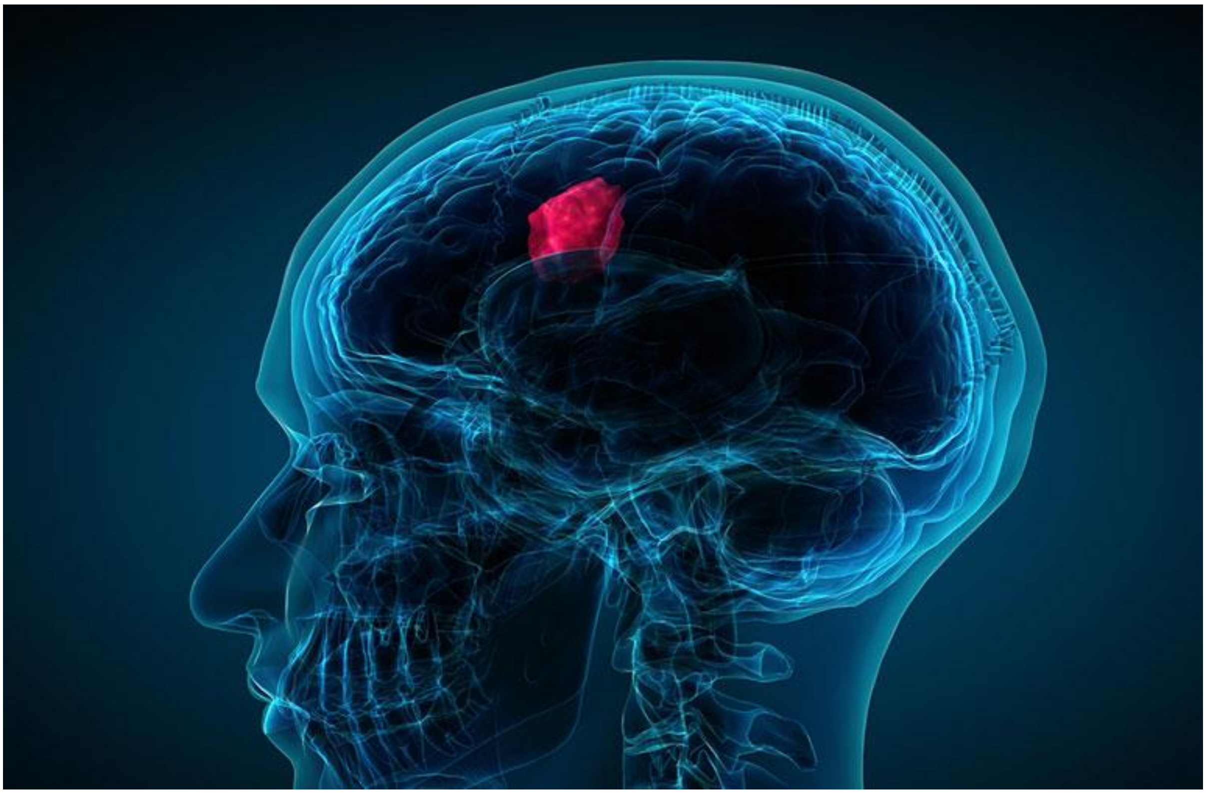
BRAIN TUMOR DETECTION

Subject Name: Neural Networks and Deep Learning
Subject Code: 15CSE380



Project By:
Gayatri K R (BL.EN.U4CSE18035)
B.J. Varshitha (BL.EN.U4CSE18048)
K. Sri Harshini (BL.EN.U4CSE18055)
M. Vaishnavi (BL.EN.U4CSE18063)

Project guide: Ms. Jyotsna C.

Abstract

Tumor is defined as the abnormal growth of the tissues. Brain tumor is an abnormal mass of tissue in which cells grow and multiply uncontrollably, seemingly unchecked by the mechanisms that control normal cells. Brain tumors can be primary or metastatic, and either malignant or benign. A metastatic brain tumor is a cancer that has spread from elsewhere in the body to the brain Epilepsy is a brain disorder in which clusters of nerve cells, or neurons, in the brain sometimes signal abnormally. Neurons normally generate electrochemical impulses that act on other neurons, glands, and muscles to produce human thoughts, feelings, and actions.

Magnetic Resonance Imaging (MRI) is an advanced medical imaging technique used to produce high quality images of the parts contained in the human body MRI imaging is often used when treating brain tumors, ankle, and foot. From these high-resolution images, we can derive detailed anatomical information to examine human brain development and discover abnormalities

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. We choose convolutional neural network-based method over other techniques because it automatically detects the important features without any human supervision and it is also computationally efficient.

Implementation of CNN in our Project

Step 1: Data Visualization

In the first step, we will analyze the MRI data. In this problem, we have a total of 253 MRI images. Out of them, 155 are labelled "yes*"* and the remaining *98* are labelled "no". Now, in a CNN, we have to train a neural network, which can be visualized as a series of algorithms that can recognize the relationships between images in a set of data and can classify them.

Step 2: Data Augmentation

In Data Augmentation, we take a particular MRI image and perform various sorts of image enhancements such as rotate, mirror and flip to get more number of images. We will apply more augmentation to the class with less number of images to get approximately equal number of images to both classes. So after applying Data Augmentation to our dataset, we have 1085 images of “yes” class and 979 images of “no” class.

Step 3: Splitting the data

In the next step, we split our data to training set and test set. 80% (1651 images) of the images will go to the training set, which will be used by our neural network to get trained. The remaining 20% (413 images) will go to the test set, with which we will apply our trained neural network and classify them to check the accuracy of our Neural Network.

Step 4: Building a CNN Model

In this step, we design a neural network using Keras library with various convolutional and pooling layers.

Step 5: Training the CNN Model

We will train the images data for around 200 “epochs”. An epoch can be thought of an iteration in which we feed the training images again and again for the Neural Network to get trained better with the training images. The Accuracy of the training set keeps improving with each iteration. This means that the Neural Network model is able to improve in classifying the image as Tumor or Not a Tumor.

Step 6: Analysis of CNN Model

After the training, we finally plot the “test” and “validation” for all the 200 epochs(iterations)

model =Sequential()
#Created a Sequential neural network object.

model.add(Conv2D(32,kernel_size=(2,2),padding=’same’,input_shape=(128,128,3)))
#Added a Input Layer with required parameters.

model.add(Conv2D(32, kernel_size=(2, 2), activation ='relu', padding = 'Same'))
#Added a Convolutional layer with required parameters.

model.add(BatchNormalization())
#For updating weights as mini-batch before adding another convolutional layer

model.add(MaxPooling2D(pool_size=(2, 2)))
#Added a Max Pooling layer of size 2by2.

model.add(Dropout(0.25))
#Dropped out 25 percent features .

model.add(Conv2D(64, kernel_size = (2,2), activation ='relu', padding = 'Same'))
Added another Convolutional layer with required parameters.

model.add(Flatten())
#Flattened all the features and given as input to the neural network.

model.add(Dense(512, activation='relu'))
#Created first neural network layer with activation function as relu and number of neurons is 512.

model.add(Dropout(0.5))
#Dropped out some features that are not contributing for efficiency.

model.add(Dense(2, activation='softmax'))
#Output layer we used softmax as activation function.

model.compile(loss = "categorical_crossentropy", optimizer='Adamax')
#Finally compiled the network .

history = model.fit(x_train, y_train, epochs = 22, batch_size = 32, verbose = 1,validation_data = (x_test, y_test))
#We trained the model.

Tools

- Keras
- Tensorflow
- Google colab

Dataset

Datasets were taken from Kaggle website https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection

Python Libraries use

- Numpy
- Sklearn
- pandas
- Keras
- OS
- OpenCV
- Seaborn
- Matpltlib

Description of modules used

1. **show_image():** This module is used in Data Visualization where images with tumor and no tumor will be displayed randomly.
2. **augment_data():**This module is used for Data Augmentation, we take a particular MRI image and perform various image enhancements such as rotate, mirror, flip to get more number of images.
3. **names():**This module is used for checking the model.
4. **plot_metrics():**This module is used for plotting losses and accuracy.

Problems Faced

Problem1: Choosing Hyperparameters

Solution: we had to run with different possibilities and finally selected 200 epochs with batch size of 32 to get accuracy of 80% with minimum loss.

Problem2: Choosing Activation Functions

Solution: We tried using Tanh at hidden layer and Sigmoid at output layer and compared accuracies and finally chose ReLu and Softmax.

Result

A Brain Tumor Detection was developed by training a CNN model for detecting the presence of tumor for a given MRI images. A small part of images from the dataset are taken and passed through the data augmentation and preprocessing modules to achieve the expected result within the computational limitations. The model was trained with an accuracy of around 80% (with a considerably better output for 1652 image training set) for the small part of dataset with 200 epochs.

Contribution of each team member

CNN model – Gayatri, Varshitha, Harshini, Vaishnavi
Data visualization – Varshitha B J
Data Augmentation – Sri Harshini K
Data Preprocessing – Gayatri K R
Training and checking the model -Vaishnavi M

Scope of future work

The scope of project could be solved by adding more variety to the MRI images to the dataset. Future work can try to overcome this lack of computational power for improved and better accuracy and reducing the validation losses.

Whether you will extend the project

This project can further be implemented by adding extra feature which can tell the type of tumor also.