

# Detection of Brain Tumor using Convolutional Neural Network

Yash Kulkarni\*

*Zeal College of Engineering and Research, Narhe, Pune.*

Archana Karlekar†

*Zeal College of Engineering and Research, Narhe, Pune.*

Akshata Kulkarni‡

*Zeal College of Engineering and Research, Narhe, Pune.*

## I. Introduction

In the project we are going to diagnose the brain MR images and suggesting the user that whether the given MR image has the tumor or not and classifying the image as Benign or malignant.

For implementing this we have used Convolutional Neural Network (CNN) for classification of the images in the 2 main classes i.e. Benign and malignant. We will give the input image in the Nifty format (.nii) our software will convert that file into a simple image (.png) and perform classification on that image using CNN. We have trained our CNN classifier on our training data set and saved that classifier using .h5 file so that it will not require retraining the classifier repeatedly for each image, this will reduce the time of execution of the algorithm.

### A. CNN(convolutional Neural Network):

CNN is same as of deep neural network with some extra layers like convolutional layer, pooling layer, flattening layer. These extra layers are used for extracting features from the images this reduces manual segmentation of the images. In CNN the relationship between neighboring pixels is maintained which makes it more accurate than any other image classifier. For learning the square plots are used which are known as filters discussed in next section of the paper.

### B. Brief intro of Brain Tumors:

Brain Tumor is the abnormal growth of brain cells. Depending on the various factors like type of the tumor, size of the tumor, its location and its growth rate threat of the tumor can be determined. Two main types of Tumors are :

- Cancerous (Malignant)
- Non-cancerous (Benign)

Benign Tumors are also known as Low Grade Glioma (LGG), non-cancerous brain tumor, growth of it is slow and it cannot invade normal tissue. Whereas Malignant Tumors are also known High Grade Glioma (HGG), Cancerous brain tumor, growth of it is very rapidly and it invade the surrounding normal cells and tissues.

## II. Literature Review

### A. Paper-1

**Title:-Classification using deep learning neural networks for brain tumors [3]** The methodology for classification of brain tumor using DNN includes following four main steps.

#### 1. Dataset acquisition:-

Brain MRIs are used as dataset. According to WHO(world Health Organisation) there is more than 120 types of brain tumors which can be differ in size and location along with this tumours can be originated from different part having different characteristics of tumour tissues. In this paper three types of malignant brain tumour types are considered :

---

\*Student in ZCOER, yash.kulkarni9597@gmail.com

†Student in ZCOER

‡Student in ZCOER

- 1) Glioblastoma
- 2) Sarcoma
- 3) Metastatic bronchogenic carcinoma

## 2. *Image Segmentation:-*

For image segmentation FCM(Fuzzy c-means) is used. Image segmentation is used to separate different normal brain tissue from brain tumor tissue. (Fuzzy C means)FCM is used to segment the brain MRI into 5 parts.

## 3. *Feature extraction and reduction*

Features are extracted using DWT(discrete wavelet transform) and reduction is performed using PCA(Principle component analysis). After segmentation, using DWT features of the segmented images is extracted. They have used a three level haar wavelet decomposition to extract 1024 (32\*32) features. As this number is not large as compared to CNN feature maps Principle Component Analysis is used.

## 4. *Classification using DNN*

After the feature extraction, classification is performed on resulted feature vector. seven fold cross validation technique is used for classification to build and train the DNN of 7 hidden layer structure.

### **Disadvantages:-**

- 1) .Require more hardware specification and take more time to process large size images like(256\*256)
- 2) Require separate method for segmenation and feature extraction.

## **B. Paper-2**

**Title:-Methods for interpreting and understanding DNN(Deep Neural Network) [4]** In this paper they have explained the problem of interpreting DNN. Machine learning techniques are used in applications like classification, NLP (natural language processing). So interpretability has become more important in application like medicines. As in such applications, dependency of the model on the exact feature must be guaranteed.

### 1. *Interpreting Deep Neural Network model:-*

We studied problem of interpreting concept learned by deep neural network. A deep neural network is an artificial neural network with multiple layers between the input and output layers. The DNN finds correct manipulation to convert the input into output for linear as well as for non-linear relationship. DNN (Deep neural network) is a collection of neurons. These neurons receive the neuron activations as input from the input layers. The neural network forms a complex mapping (non-linear) from input layer to output layer.

- **Activation Maximization**

It is a technique that maximizes the activation of hidden units after giving the desired output class. It is a type of analysis framework which searches for an input pattern that produces a maximum model response for a specific output.

- **LRP LRP** stands for Layer Wise Relevance Propagation. Layer-wise propagation is a backward propagation technique. In LRP technique is based on conservation principle. In neural Network first relevance is concentrated at the o/p node and then iteratively propagate it backward to the input layers.

### 2. *Applications*

- 1) Image Caption Generation
- 2) Object classification in picture
- 3) Automatic Machine Translation

## **C. Paper-3**

**Title:-Visualizing Higher-Layer Feature of a DNN(Deep Neural Network) [5]** Only model definitions and the quantitative analyses is not enough, qualitative comparisons of the solutions learned by different DNN architectures

is also important. In this paper we have studied good qualitative interpretation of high level features represented by various models.

### *1. Models*

In this paper we have studied two models. The first model is a Deep Belief Net (DBN). This model is obtained by training and stacking three layers as Restricted Boltzmann Machines in a greedy manner. The second model is Stacked Denoising Auto-Encoder (SDAE) by Vincent et al.

### *2. Maximizing the activation*

We find the input patterns of bounded norm which maximize the activation of a given hidden unit. As the activation function of a unit in the first layer is a linear function of the input. We can find this for a given unit, the input sample(s) (from either the training or the test set) that give rise to the highest activation of the unit. In this we have studied Sampling from a unit of a Deep Belief Network.

### *3. Sampling a Unit*

The activation maximization method produces features and it decides which examples would “fit” these features; the sampling method produces examples and it lets us decide which features these examples have in common.

**Limitations** one cannot find a simple representation of a higher layer unit as we scale the datasets to larger and larger images i.e DNN cannot process larger size images efficiently.

## **III. Methodology**

Our methodology is based on the CNN architecture for the classification of the images where the classifier identifies the Brain Tumor in MR Images. The proposed methodology for classifying the brain tumors in Brain MR Images is as follows:

- 1) Data set acquisition
- 2) Data preprocessing
- 3) Splitting of data
- 4) Implementation of CNN

### **A. Data set acquisition:**

According to World Health Organization, there are total 120 types of brain tumors out of which we are focusing on the two main types of tumors Low Grade Glioma (LGG) and High Grade Glioma (HGG) which is also known as Benign and Malignant respectively.

For this project, data has collected from BraTS2018 challenge. We have downloaded the data with the ground truth-values from [https://www.cbica.upenn.edu/sbia/Spyridon.Bakas/MICCAI\\_BraTS/2018/MICCAI\\_BraTS\\_2018\\_Data\\_Validation.zip](https://www.cbica.upenn.edu/sbia/Spyridon.Bakas/MICCAI_BraTS/2018/MICCAI_BraTS_2018_Data_Validation.zip) [1].

We have downloaded 285 patients real brain MR Images out of which 210 were HGG and 75 were LGG. All the images were in the sequence flair, t1, t2 and t1ce. These are the types of MR images in medical field depending on the various weighed conditions of imaging. In total, we have downloaded 1421 Brain magnetic resonance images (MRI) out which 1046 are HGG and 345 are LGG images.

### **B. Data pre-processing:**

#### *1. Formation of dataset:*

As dataset is of real Brain MR Images, it was arranged in separate folder for each patient. As the downloaded dataset was in the subfolder format, which is not suitable for the classification so it was necessary to arrange that data in the proper format.

We arranged that data in two main folders namely LGG and HGG.

## 2. Preprocessing of Images:

The images in the dataset was in nifty format, which is a well-known format for medical imaging. These images are 3 Dimensional images which are not supported by CNN as it works on 2D images only so we converted that all images into a simple 2D format (.png format) manually.

For conversion, we load the MR image first using nilearn library and converted that image into a 2D format using library function. We done this same process on all the images.

## C. Splitting of dataset:

After conversion of that all .nii files into .png format we split data into training and testing sets. As the standard split ratio is 75-80% for training, we split our data into 80% for training and remaining for testing.

While splitting our dataset we avoided the overlapping of the data i.e. we avoided same image in training as well as testing set as it may cause error in accuracy of the classifier.

## D. Implementation of CNN:

After splitting dataset into training and testing sets, we implemented the CNN. We built CNN in python using Keras and Tensorflow at the backend. Each image is given to the classifier with some transformations in the original image for training and testing and the accuracy of the system is observed after each epoch.

CNN can be implemented using 4 main steps:

- 1) Convolution Layer
- 2) Maxpooling Layer
- 3) Flattening Layer
- 4) Total connection

Now we will discuss all these stages in brief

### 1. Convolution Level

Convolution function can be given as the product of the 2 functions as given below:

$$(f * g)(t) =_{def} \int_{-\infty}^{\infty} (T)g(t - T)dT$$

Convolution operation works using these 3 elements

- Input image
- Feature Detector
- Feature map

### Input image:

As humans, computers cannot see the image as it is, images are nothing but a pixel value for the computer. Consider the following image as an example



**Fig. 1 Smiling face**

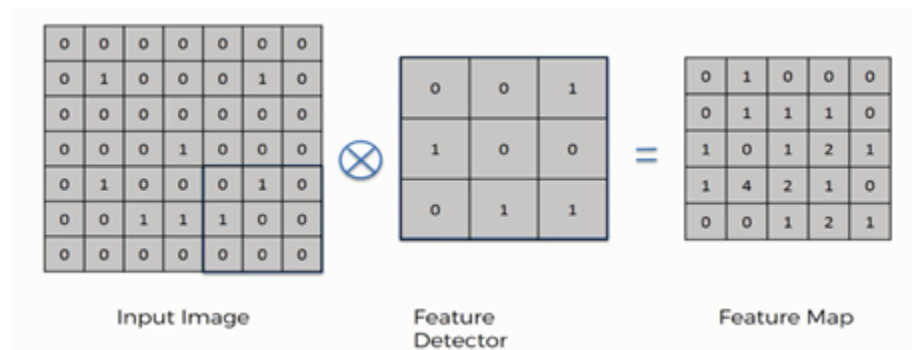
We can see this as a smiling face but computer will see this image as a pixel matrix. Computer will see this image on next page

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

**Fig. 2 Input Image**

With 1 representing value is present and 0 representing absence of the value

**Feature Detector:** It is simply a feature matrix, which compare for a specific feature in the image matrix depending on the image size and task we can create the feature detector for this example we will consider 3\*3 feature detector which is more conventional, it is often referred as filter.



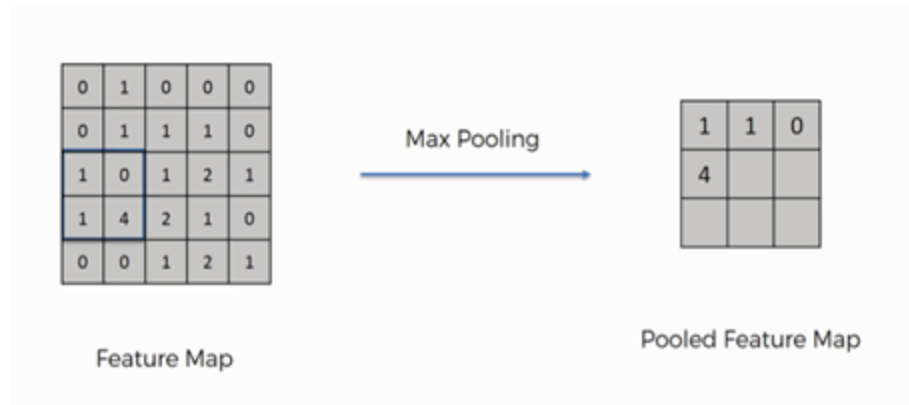
**Fig. 3 Convolutional Operation**

**Feature map:** It is the product of image matrix and feature detector matrix. As shown in the above figure.

## 2. Max pooling

In this step, we enable the convolutional neural network to detect the required pattern from the image when presented in any manner like different viewing angle, lightening condition, etc. In order to perform pooling, we must finish convolution operation first, so that we will have feature map ready.

In this we pool out the max values from the feature map using some fixed pixel strides (2\*2 pixel strides are shown in the example below) and creates new pooled feature map.

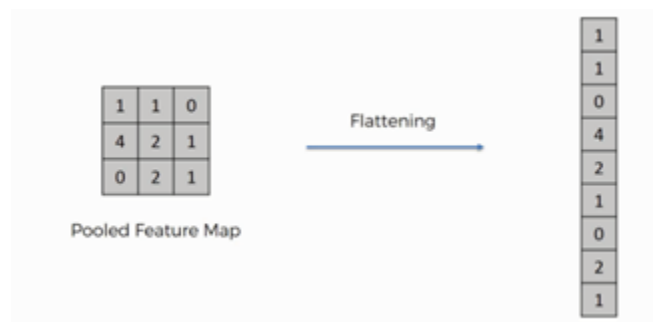


**Fig. 4 Max Pooling**

Similar to convolution step, pooling disposes the unnecessary information or features from the original input and gives the optimized patterns.

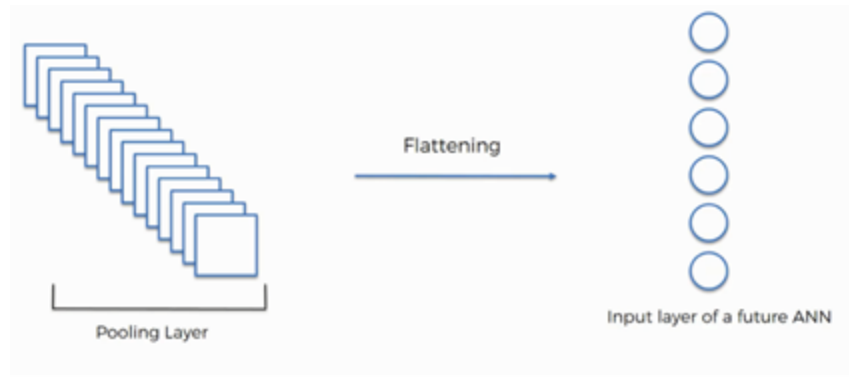
### 3. Flattening

In this step, we convert the pooled feature map into a vector so that we can give it as input to the neural network. The process is shown below:



**Fig. 5 Flattening of pooled feature map**

We perform flattening because we have to load this data into an artificial neural network in the next step. As we have many pooled layers we flatten all these in into a vector and pass this vector as a input to the neural network for further processing.



**Fig. 6 Inserting data into NN**

#### *4. FULL CONNECTION*

This is the step where we connect our convolution layer to our artificial neural network and create the Convolutional Neural Network. Neural network takes this data as input, learns the various features, patterns, attributes and combine them to wider attributes to classify the images.

### **IV. Result**

All the process is done using pure python and python libraries only. We got 99% classification accuracy of our model. It can classify given MR Image as HGG or LGG from the features of the image.

### **V. Conclusion and Future work**

In this project, we had successfully implemented image classification of Brain tumor in two main types as HGG and LGG using CNN. The tumor part can be encircled in the future using open cv and CNN together.

### **References**

- [1] <https://www.med.upenn.edu/sbia/brats2018/data.html>
- [2] <https://www.udemy.com/machinelearning>
- [3] Classification using Deep Learning Neural Networks for Brain Tumors
- [4] Understanding Deep neural network
- [5] Visualizing Higher-Layer Features of a Deep Network