Data and sources of Data:

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\_Bra

TS\_2018\_Data\_Validation.zip”.

In total, we have downloaded 1421 magnetic resonance images (MRI) out which 1046 are HGG and 345 are LGG images.

Data pre-processing:

As the downloaded data was not in the proper format as required for classification, we arranged the data as per requirement.

The images in the dataset was in nifty format so we converted that all images into a .png format manually.

After conversion of that all .nii files into .png format we split data into training and testing datasets.

Theoretical framework

In this project, we are going to classify the images given by the user into a cancerous or non-cancerous class for that we have used CNN for classifying the images.

CNN(convolutional Neural Network)

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that we developed for neural networks still apply on CNNs.

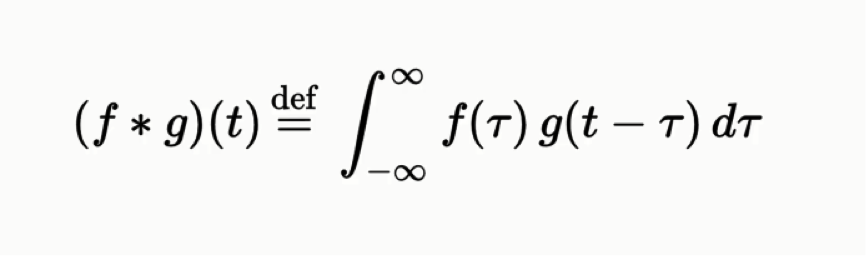
CNNs derive their name from the “convolution” operator. The primary purpose of Convolution in case of CNNs is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

CNN generally works in 4 main stages

1. Convolutional Layer
2. Max pooling layer
3. Flattening layer
4. Full connection layer

Now we will discuss all these stages in detail one by one

1. 1.Convolution Level

**[](http://www.superdatascience.com/wp-content/uploads/2018/08/Convolutional_Neural_Networks_CNN_Step1_Img1.png)**Convolution is a function derived from two given functions by integration, which expresses how the other modifies the shape of one. That can sound baffling as it is, but to make matters worse, we can look at the convolution formula:

Now we will not go deep in math’s here we will go directly into actual convolutional operation, we will take an example, which will provide you with a breakdown of everything you need to know about this process.

Convolution operation works using these 3 elements Input image

Feature Detector

Feature map

Sometimes a 5×5 or a 7×7 matrix is used as a feature detector, but the more conventional one, and that is the one that we will be working with, is a 3×3 matrix. The feature detector is often referred to as a “kernel” or a “filter,”

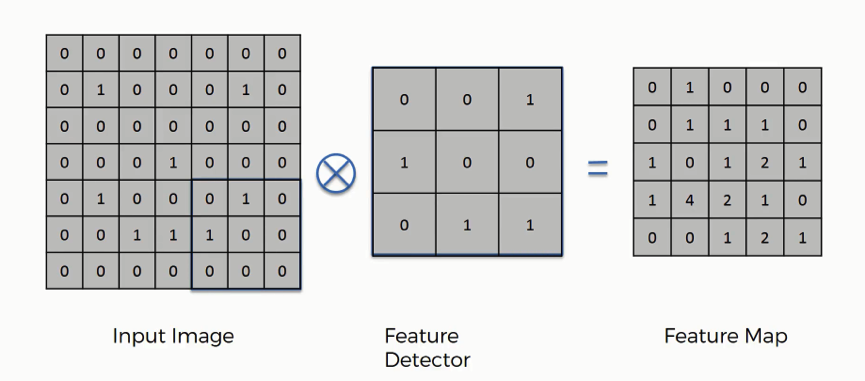


Figure – Convolutional Operation

1.2 MAX POOLING

The purpose of max pooling is enabling the convolutional neural network to detect the required pattern when presented with the image in any manner.

Max pooling is concerned with teaching your convolutional neural network to recognize that despite all of the differences like viewing angle, lightening condition, etc. In order to do that, the network needs to acquire a property that is known as “spatial variance.”

In order to reach the pooling step, we need to have finished the convolution step, which means that we would have a feature map ready.

Types Of Pooling:

There are several types of pooling. These include among others the following:

* Mean pooling
* Max pooling
* Sum pooling

Our main focus here will be max pooling.

1.2.1Pooled Feature Map:

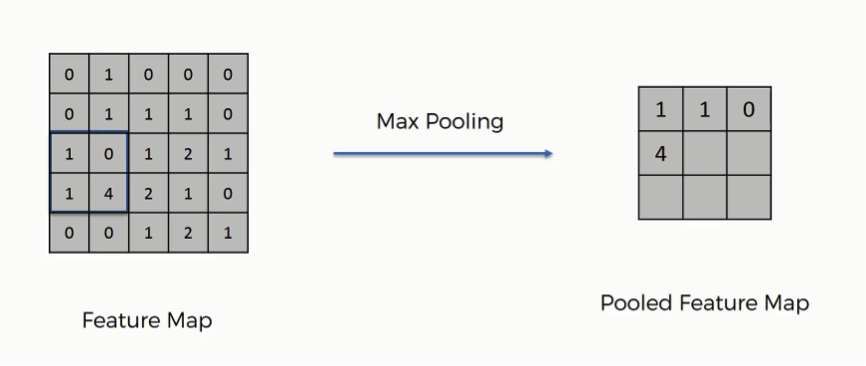
**[](http://www.superdatascience.com/wp-content/uploads/2018/08/Convolutional_Neural_Networks_CNN_Step2_Img3.png)**

Figure – Max Pooling

This time you'll place a 2×2 box at the top-left corner, and move along the row.For every 4 cells your box stands on, you'll find the maximum numerical value and insert it into the pooled feature map. In the figure below, for instance, the box currently contains a group of cells where the maximum value is 4.

In this example, we are using 2-pixel strides. That's why we end up with a 3×3 pooled featured map. Generally, strides of two are most commonly used.

Just like in the convolution step, the creation of the pooled feature map also makes us dispose of unnecessary information or features. In this case, we have lost roughly 75% of the original information found in the feature map since for each 4 pixels in the feature map we ended up with only the maximum value and got rid of the other 3. These are the details that are unnecessary and without which the network can do its job more efficiently.

Again, this is an abstract explanation of the pooling concept without digging into the mathematical and technical aspects of it.

1.3 Flattening

As the name of this step implies, we are literally going to flatten our pooled feature map into a column like in the image below.

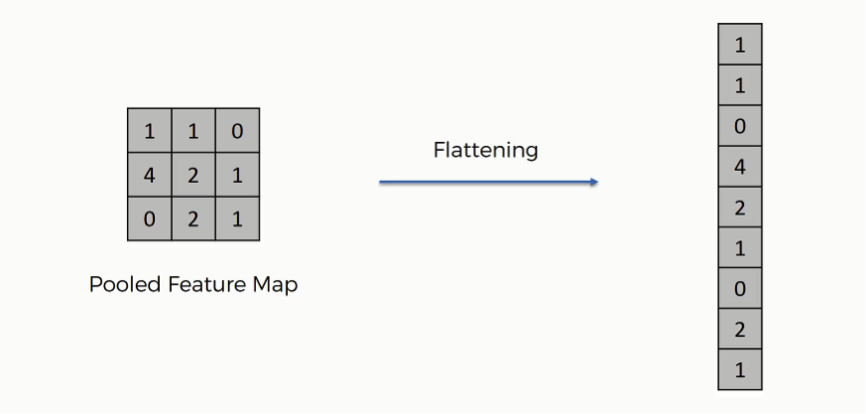
**[](http://www.superdatascience.com/wp-content/uploads/2018/08/Convolutional_Neural_Networks_CNN_Step3_Img1.png)**

Figure-Flattening of pooled feature map

The reason we do this is that we're going to need to insert this data into an artificial neural network later on.

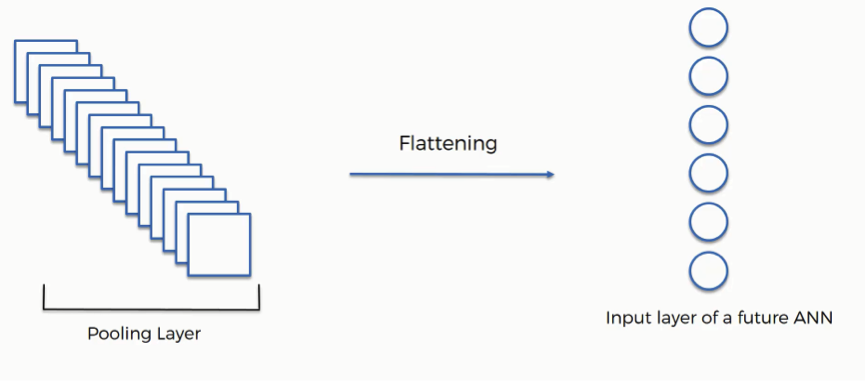
**[](http://www.superdatascience.com/wp-content/uploads/2018/08/Convolutional_Neural_Networks_CNN_Step3_Img2.png)**

Figure – Inserting data into NN

After the flattening step is that you end up with a long vector of input data that you then pass through the artificial neural network to have it processed further.

1.4 FULL CONNECTION

Here's where artificial neural networks and convolutional neural networks collide as we add the former to our latter.

It's here that the process of creating a convolutional neural network begins to take a more complex and sophisticated turn.

As you see from the image below, we have three layers in the full connection step:

1.Input layer 2.Fully-connected layer 3.Output layer

**[](http://www.superdatascience.com/wp-content/uploads/2018/08/Convolutional_Neural_Networks_CNN_Step4_Img1.png)**

Figure – Full connection

* What is aim of this step?

The role of the artificial neural network is to take this data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images, which is the whole purpose from creating a convolutional neural network.

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. Use 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 dataset 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.

Brief intro of Brain Tumors:

Brain tumor is any mass that results from an abnormal and an uncontrolled growth of cells in the brain. Its threat level depends on a combination of factors like the type of tumor, its location, its size and its state of development. Brain Tumors can be

1. cancerous (malignant)
2. non-cancerous (benign)

Benign brain tumors are low grade, non-cancerous brain tumors, which, grow slowly and push aside normal tissue but do not invade the surrounding normal tissue. They are homogeneous, well defined and are known as non- metastatic tumors, because they do not form any secondary tumor. Whereas, malignant brain tumors are cancerous brain tumors, which grow rapidly and invade the surrounding normal tissue. Malignant brain tumors or cancerous brain tumors counted among the most deadly diseases.