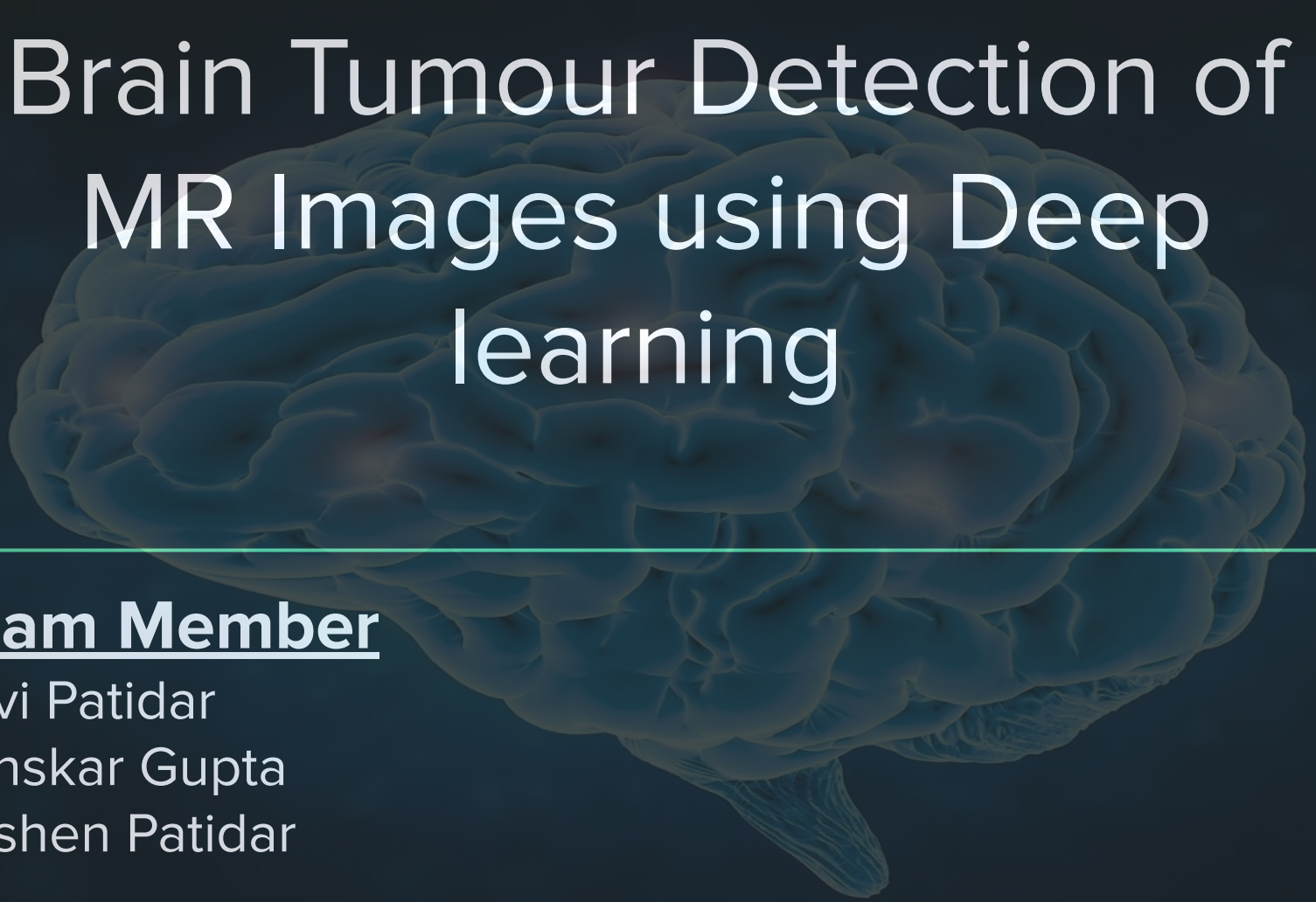
Brain Tumour Detection of MR Images using Deep learning



Team Member

Ravi Patidar

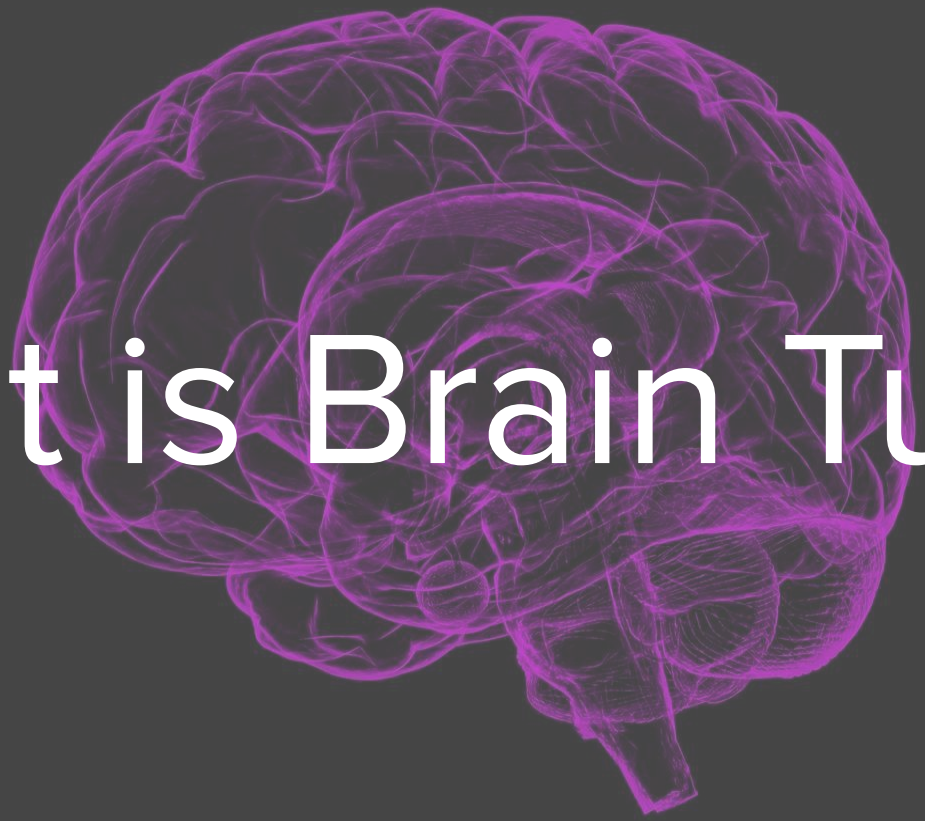
Sanskar Gupta

Sushen Patidar

Problem Statement :

- A brain tumor is one of the major causes of death among other types of cancers.
- Proper and timely diagnosis can prevent the life of a person to some extent.
- We used Magnetic resonance (MR) imaging for this project. Magnetic resonance (MR) imaging is widely used by physicians in order to determine the existence of tumors or the specification of the tumors.
- The qualification of brain cancer treatment depends on the physician’s experience and knowledge.

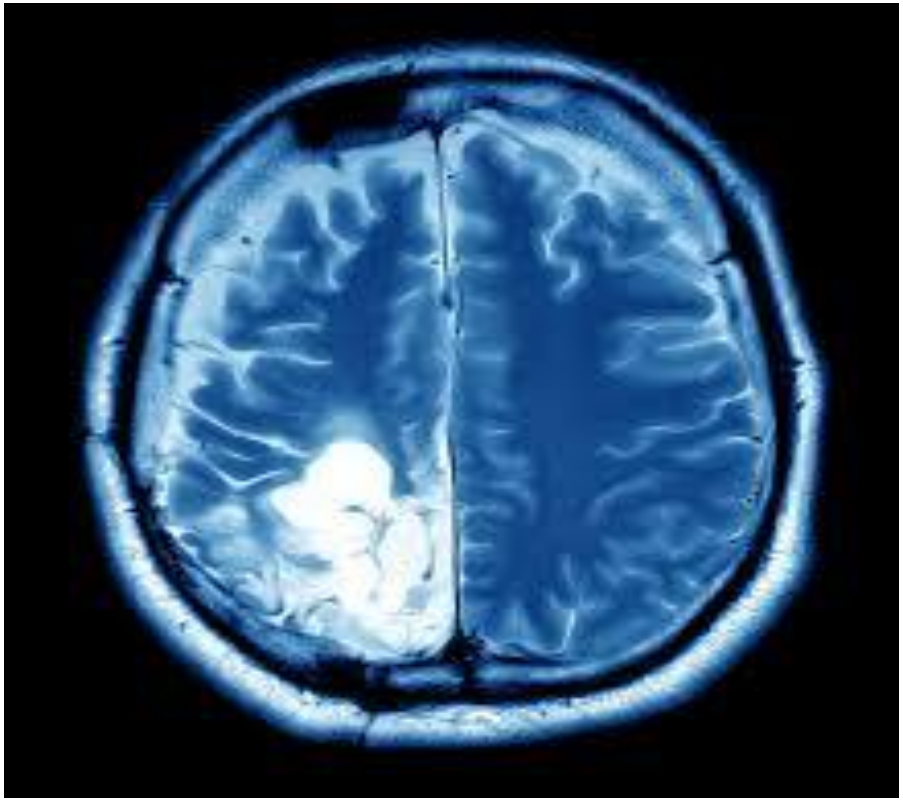
What is Brain Tumour?



A brain tumor is a collection, or mass, of abnormal cells in your brain.

Brain tumors can be cancerous (malignant) or noncancerous (benign).

When benign or malignant tumors grow, they can cause the pressure inside your skull to increase. This can cause brain damage, and it can be life-threatening.



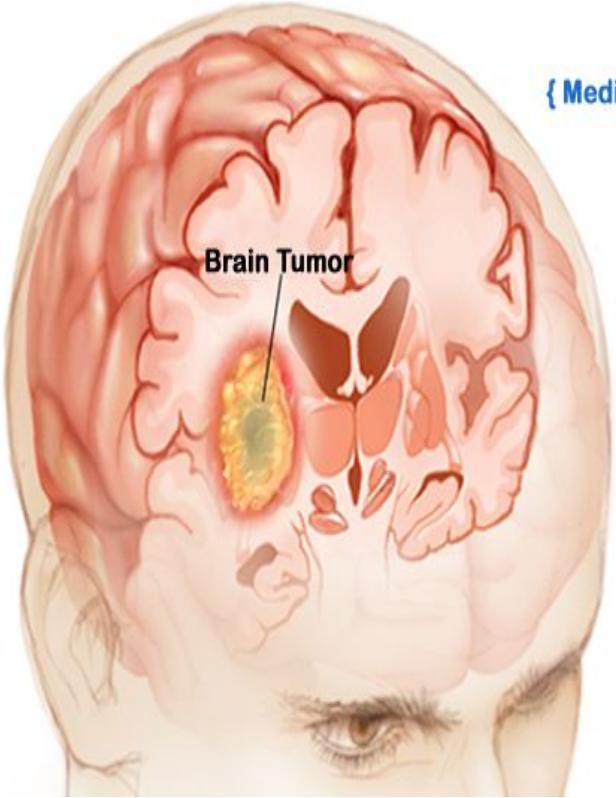


Image of Brain Tumour

Brain Tumour Categories?

Brain tumors are categorized as primary or secondary.

- A primary brain tumor originates in your brain. Many primary brain tumors are benign.
- A secondary brain tumor, also known as a metastatic brain tumor, occurs when cancer cells spread to your brain from another organ, such as your lung or breast.

Development Tools and Technology:

1. Google Colab
2. Jupyter Notebook
3. Django Framework
4. Pycharm

Requirements :-

1. Functional Requirements:-
    1. Upload MRI image
    2. Preprocessing and Classification will perform in backend.

2. Non-Functional Requirements
    1. User friendly UI/UX
    2. Easy for radiologist and neurologist.
    3. Fast response
    5.Automated

Solution :

- We have proposed an automated reliable system for the diagnosis of the brain tumor.The proposed system is a system for brain tumor diagnosis.
- First, noise removal is performed as the pre-processing step on the brain MR images.
- Texture features are extracted from these noise-free brain MR images.
- The next phase of the proposed system is Self-organizing Mapping based feature training and followed by the CNN based classification that is based on these extracted features. More than 85% of accuracy is achieved by the classification phase.

- The results of the proposed technique show that tumor images are recognized quite accurately.
- This technique has been tested against the datasets of different patients received from a medical organization.
- There are different stages:

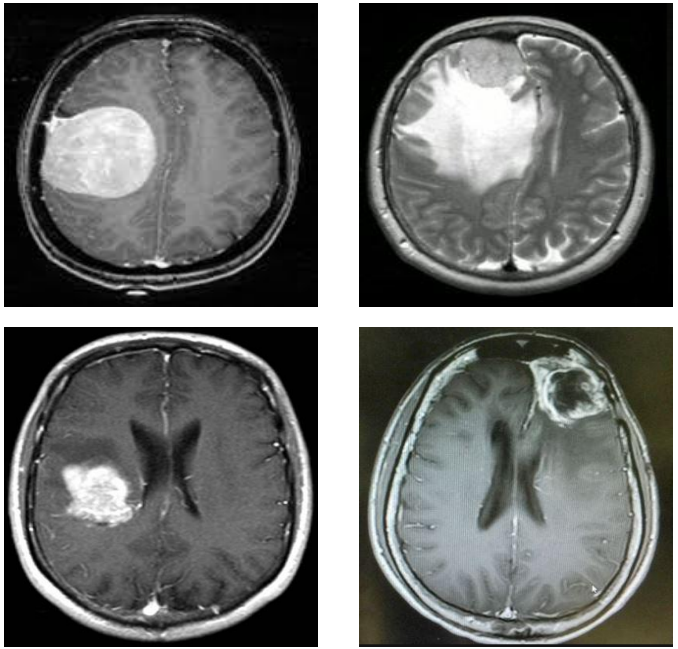
  a. Pre-processing,
  b. Image acquisition,
  c. Feature extraction &
  d. Classification using a Convolutional neural network
- GUI, we used the Django framework and HTML/CSS.

About Dataset

- A brain **MRI images dataset** founded on Kaggle. To find dataset click here.
- The dataset contains 2 folders:
  1. **Yes** :155 Brain MRI Images that are tumorous (malignant)
  2. **No** :98 Brain MRI Images that are non-tumorous(benign).
- Yes and No which contains 253 Brain MRI Images. Meaning that **61%** (155 images) of the data are positive examples and **39%** (98 images) are negative.

MRI Images

Tumorous



Non Tumorous

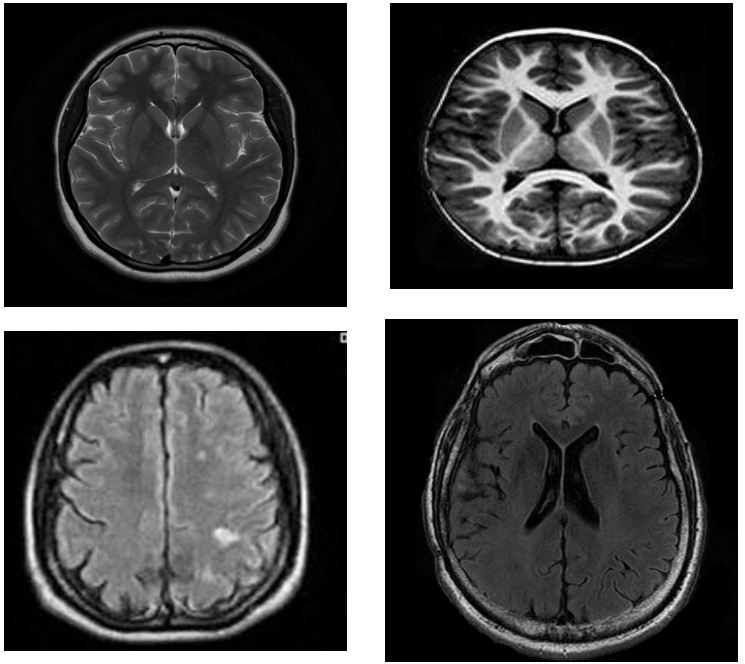
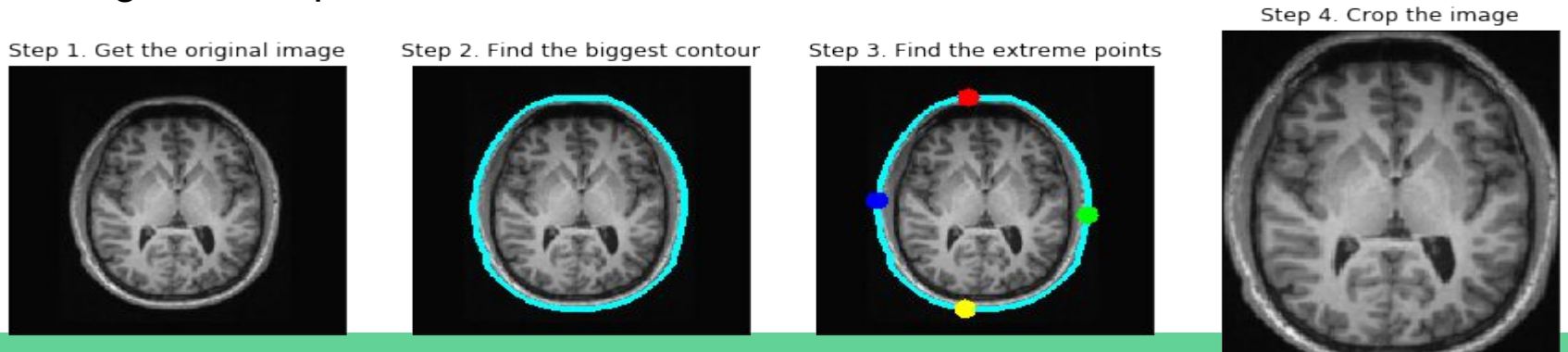


Image Preprocessing

1.Image acquisition :-

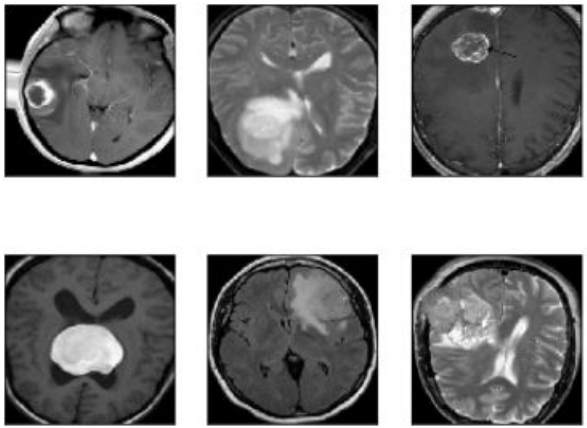
Crop the part of the image that contains only the brain.

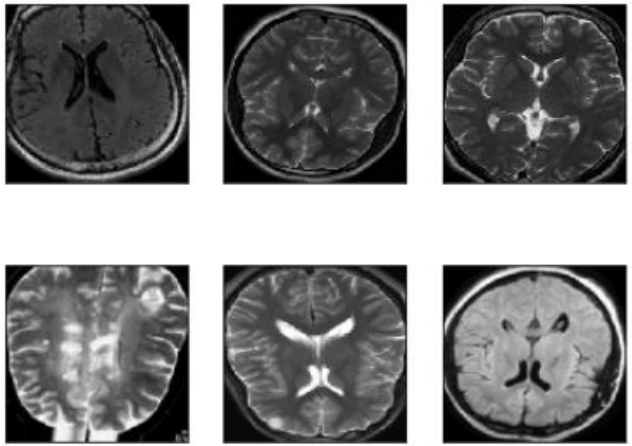
Firstly we find the area which contain only brain part in the image. Than find the extreme top, bottom, left and right extreme points of the brain in the image with OpenCV.



2. Resize the images :-

We converted images to a perticular shape of **(240, 240, 3)=(image_width, image_height, number of channels)**: because images in the dataset come in different sizes. So, all images should have the same shape to feed it as an input to the neural network.





3. Remove noise-

We did this to smooth our image to remove unwanted noise. We used Gaussian blur.

Noise means the pixel in the image show different intensity values instead of true pixel values that are obtained from image

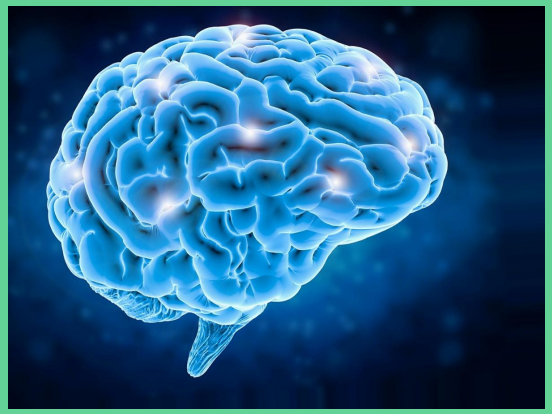
Image Augmentation:

- Data augmentation is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without actually collecting new data.

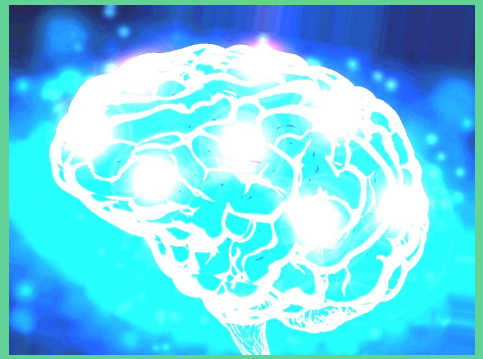
- Since this is a very small dataset, There wasn’t enough examples to train the neural network. And, data augmentation was useful in solving the data imbalance issue.

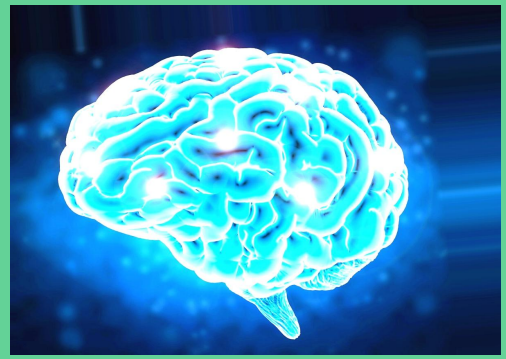
- Before data augmentation, the dataset consisted of:
- **155 positive** and **98 negative** examples, resulting in 253 example images.
- After data augmentation, now the dataset consists of:
- **3875 positive (55%)** and **3136 negative (45%)** examples, resulting in **7011 (100%)** example images.

Sample Of Augmentation Image



Original image





Augmentation Image

What is Artificial Neural Network(ANN)?

Artificial Neural Networks (ANN) are the pieces of a computing system designed to simulate the way the human brain analyzes and processes information.

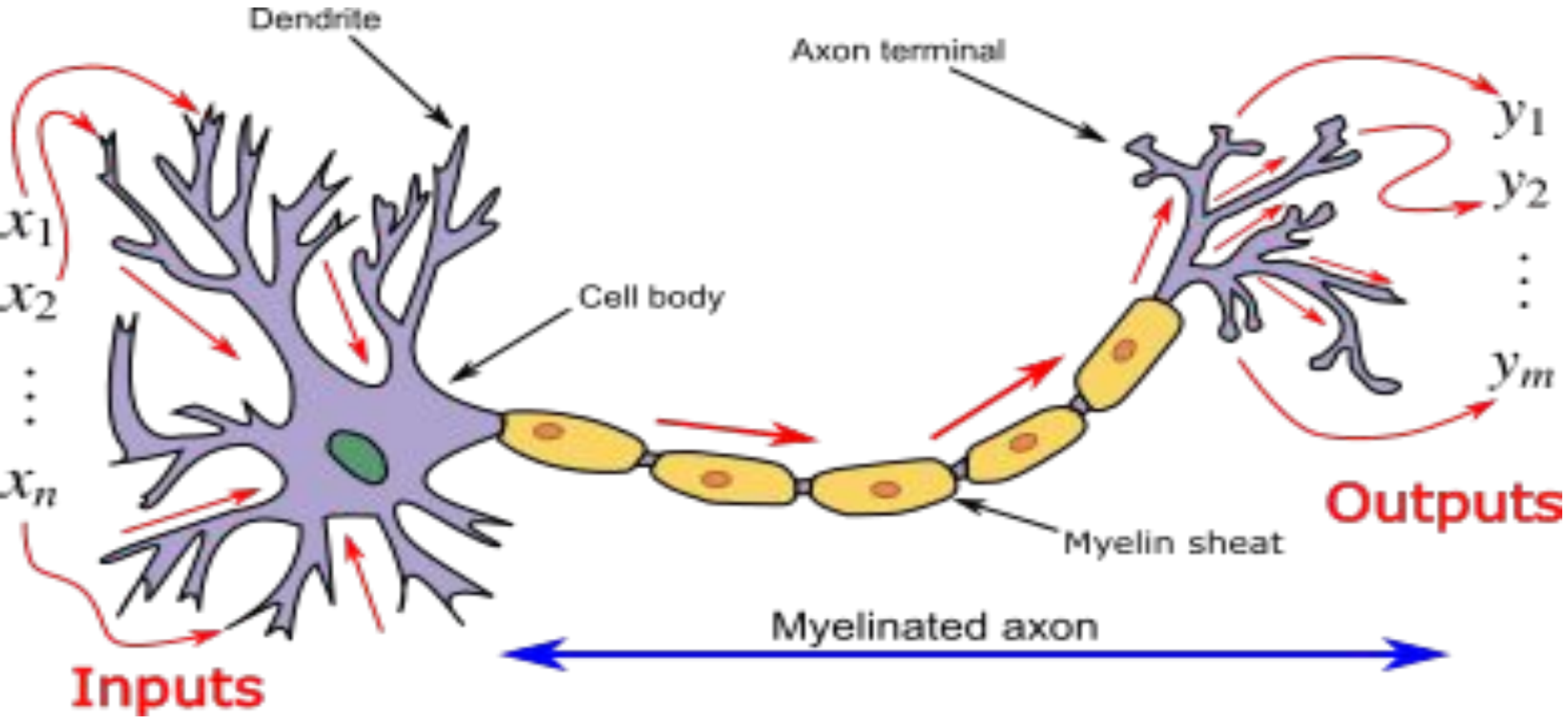
2.Artificial Neural Network

- An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the ANN because a neural network changes - or learns, in a sense - based on that input and output.
- ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found.
- ANN is also known as a neural network.

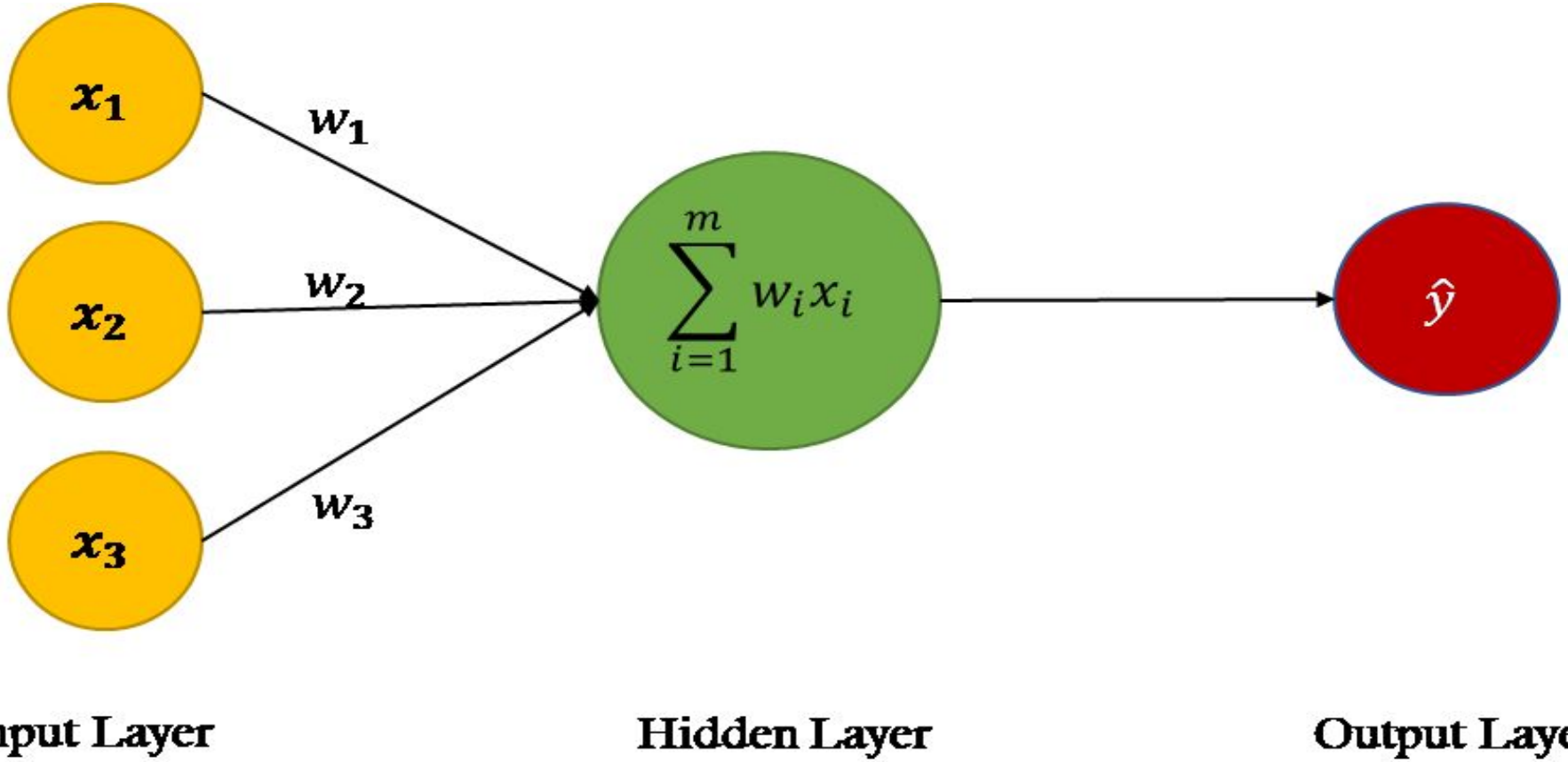
Neural Networks

- A method of computing, based on the interaction of multiple connected processing elements.
- A powerful technique to solve many real world problems.
- The ability to learn from experience in order to improve their performance.
- Ability to deal with incomplete information.

Biological Neuron



Artificial Neurons



Basics Of Neural Network

- Biological approach to AI
- Developed in 1943
- Comprised of one or more layers of neurons
- Several types, we‟ll focus on feed-forward and feedback networks

Convolution Neural Network

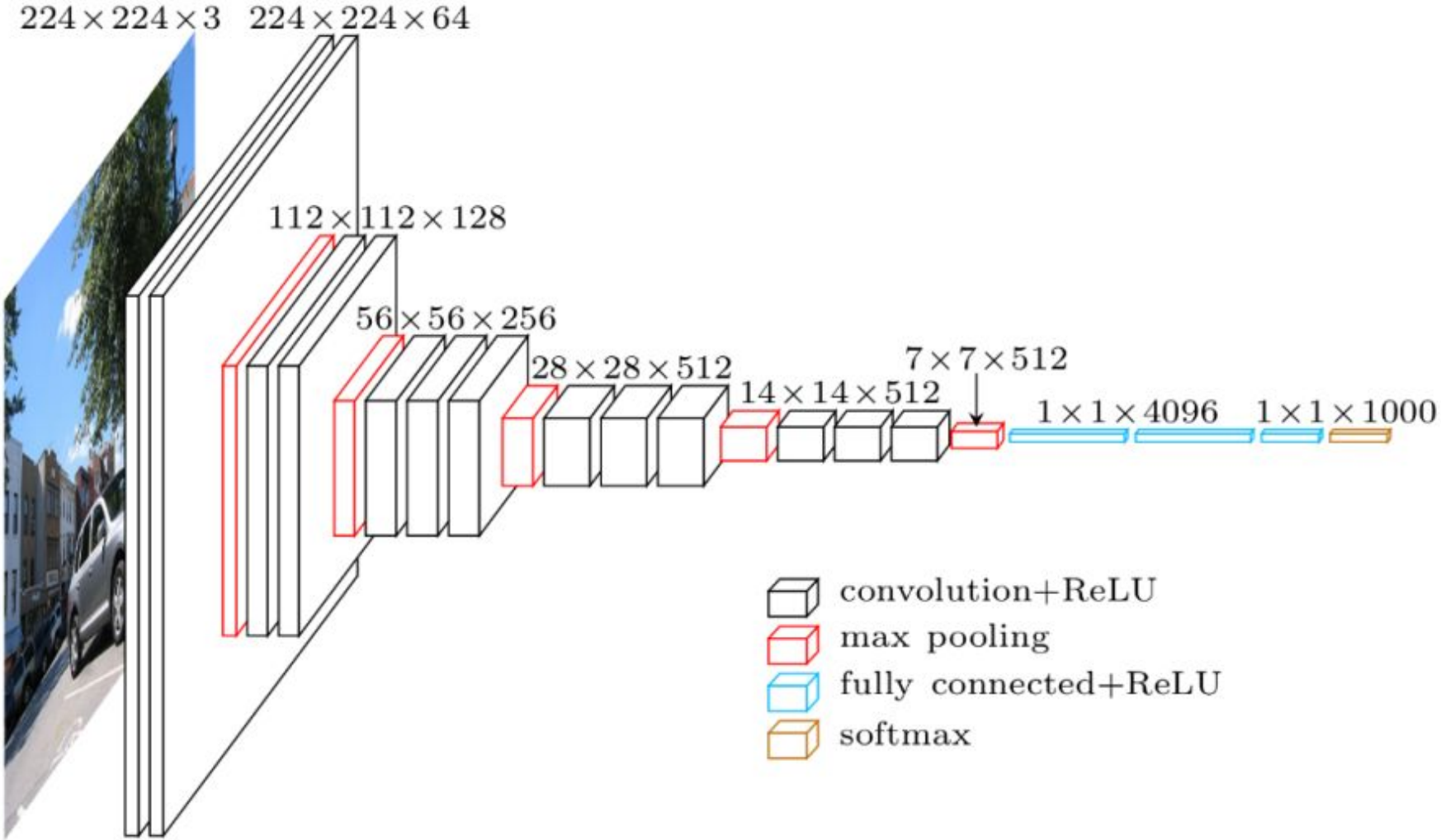
In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

It is similar to the basic neural network. CNN also have learnable parameter like neural network i.e, weights, biases etc.

Why should we use CNN ?

Problem with Feedforward Neural Network

Suppose if we have to work on MNIST dataset, you know each image in MNIST is 28 x 28 x 1(black & white image contains only 1 channel). Total number of neurons in input layer will 28 x 28 = 784, this can be manageable. What if the size of image is 1000 x 1000 which means you need 10^6 neurons in input layer. Oh! This seems a huge number of neurons are required for operation. It is computationally ineffective right. So here comes Convolutional Neural Network or CNN. In simple word what CNN does is, it extract the feature of image and convert it into lower dimension without loosing its characteristics. In the following example you can see that initial the size of the image is 224 x 224 x 3. If you proceed without convolution then you need 224 x 224 x 3 = 100, 352 numbers of neurons in input layer but after applying convolution you input tensor dimension is reduced to 1 x 1 x 1000. It means you only need 1000 neurons in first layer of feedforward neural network.


224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

Layers of CNN

There are five different layers in CNN

- **Input layer**
- **Convo layer (Convo + ReLU)**
- **Pooling layer**
- **Fully connected(FC) layer**
- **Softmax/logistic layer**
- **Output layer**

Layers in CNN

1 Input Layer

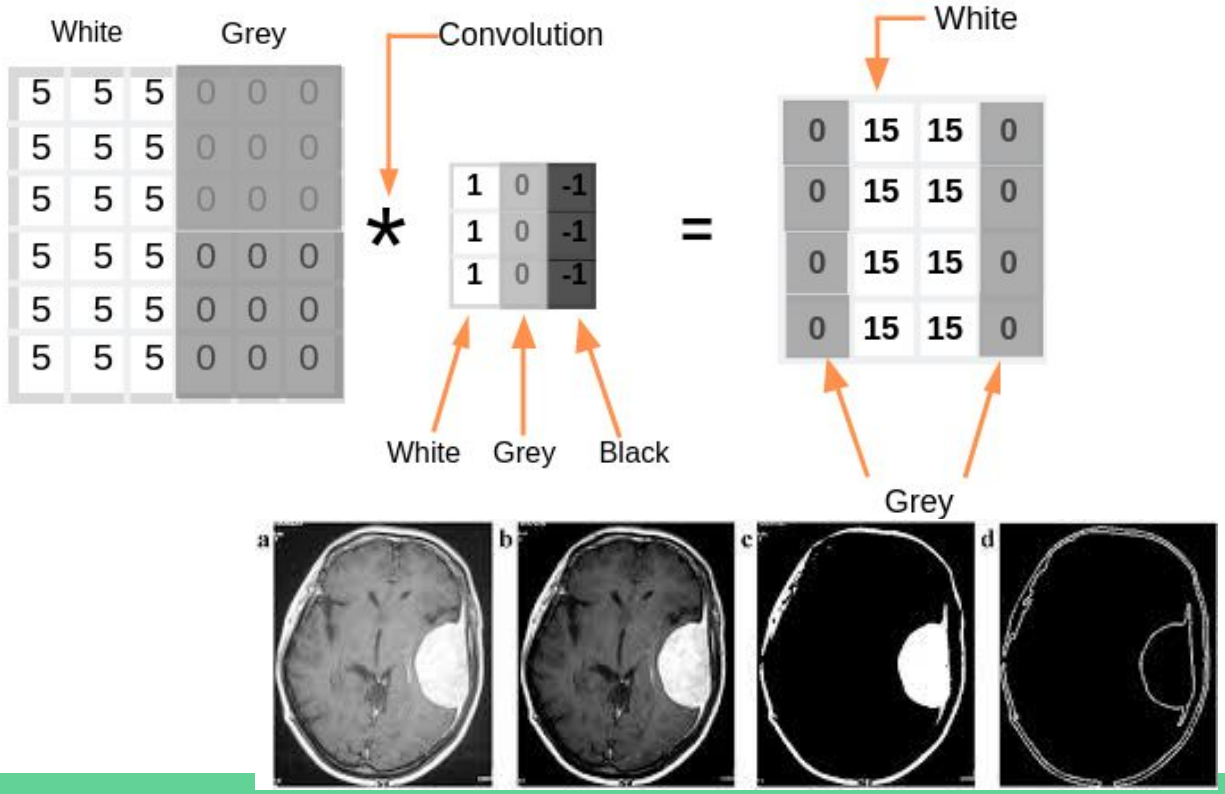
Input layer in CNN should contain image data. Image data is represented by three dimensional matrix as we saw earlier. You need to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, you need to convert it into 784 x 1 before feeding into input. If you have “m” training examples then dimension of input will be (784, m)

2 . Convo layer (Convo + ReLU)

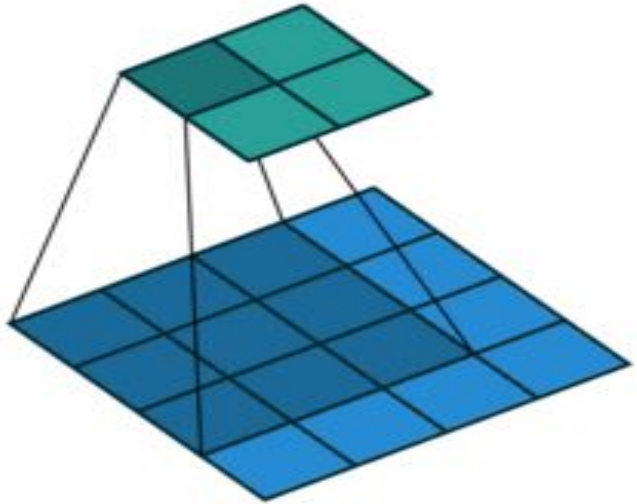
Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer.
First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field(it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then we slide the filter over the next receptive field of the same input image by a Stride and do the same operation again. We will repeat the same process again and again until we go through the whole image. The output will be the input for the next layer.
Convo layer also contains ReLU activation to make all negative value to zero.

Operation perform in Convo layer

2.1.Edge Detection



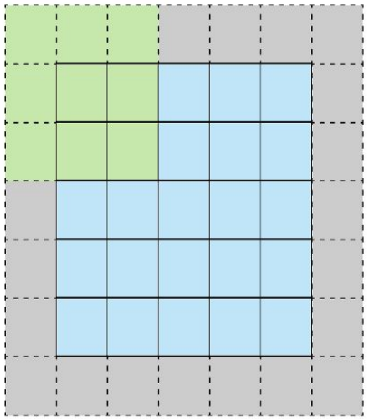
The convolution operation can be visualized in the following way. Here our image dimension is 4 x 4 and filter is 3 x 3, hence we are getting output after convolution is 2 x 2.



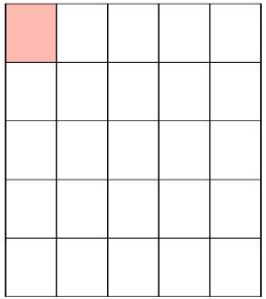
Visualization of convolution

2.2 . Stride and Padding

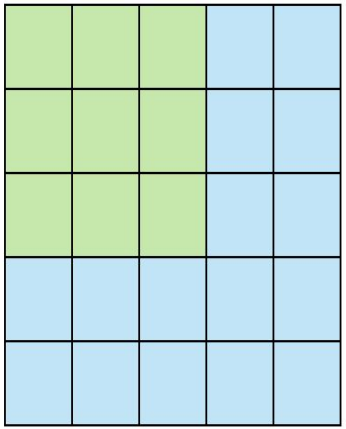
Stride denotes how many steps we are moving in each steps in convolution.By default it is one



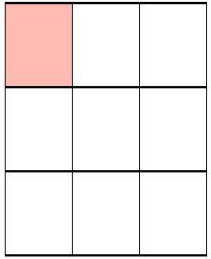
Stride 1 with Padding



Feature Map

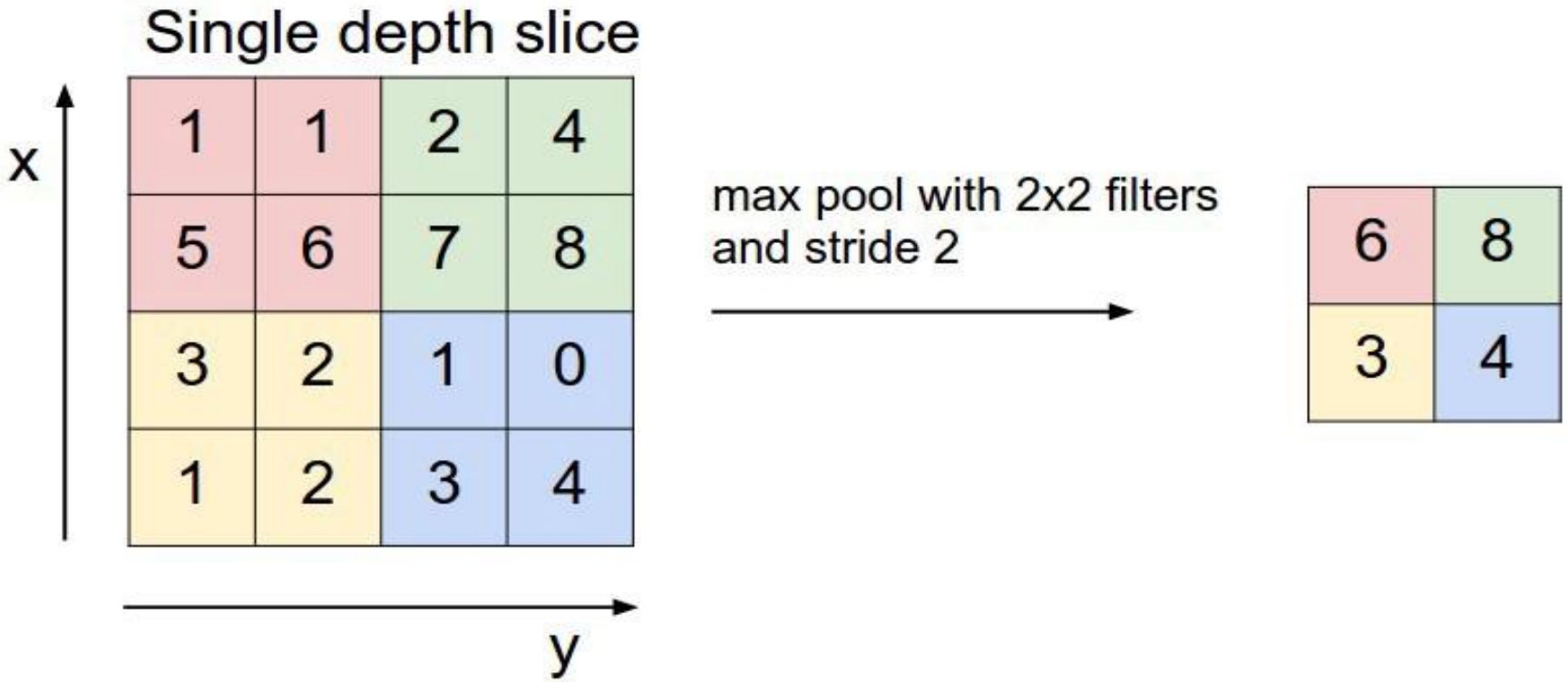


Stride 1



Feature Map

3.Pooling Layer



4. Fully Connected Layer(FC)

Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different category by training.

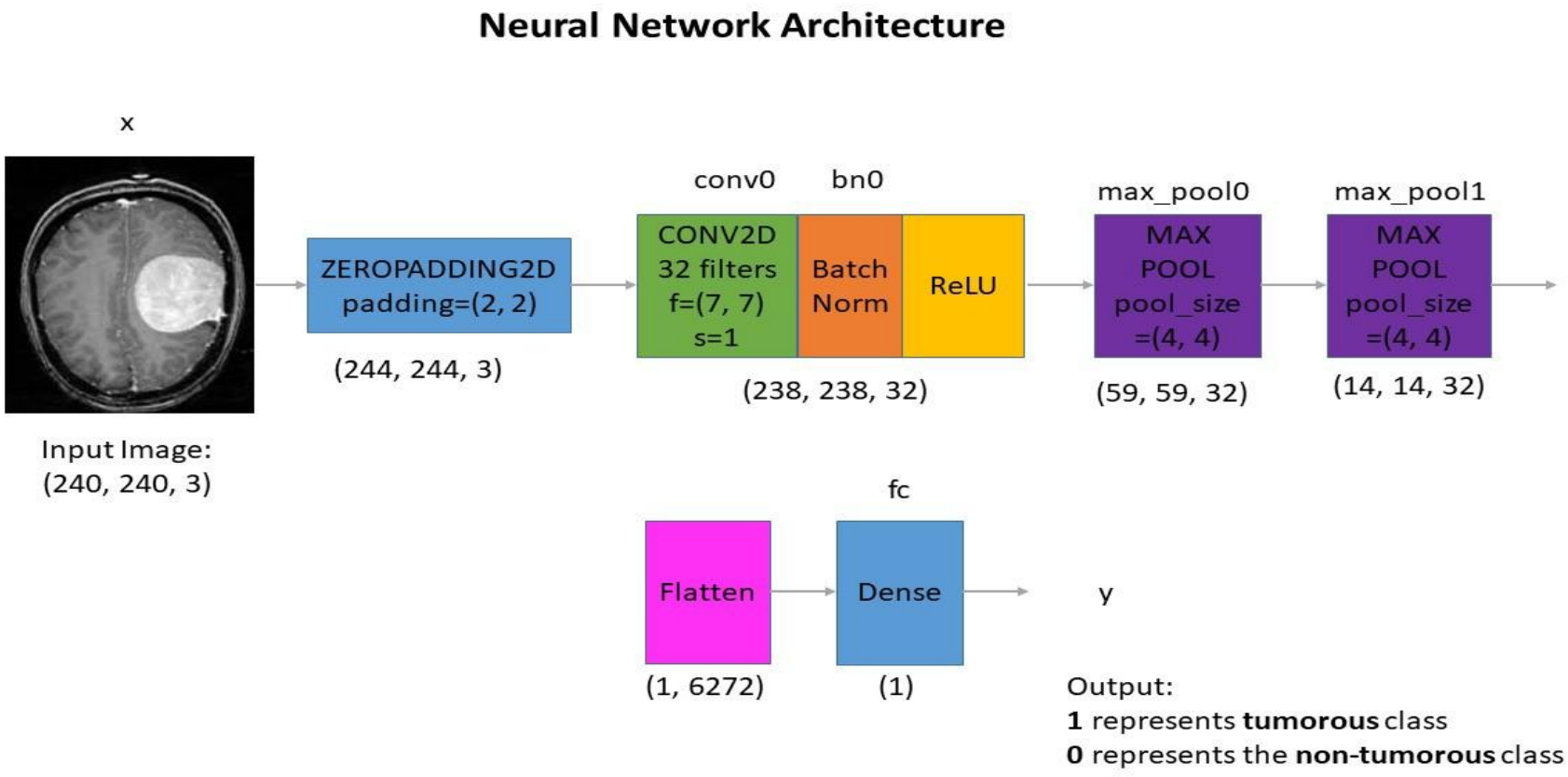
5. Softmax / Logistic Layer

Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multi-classification.

6. Output Layer

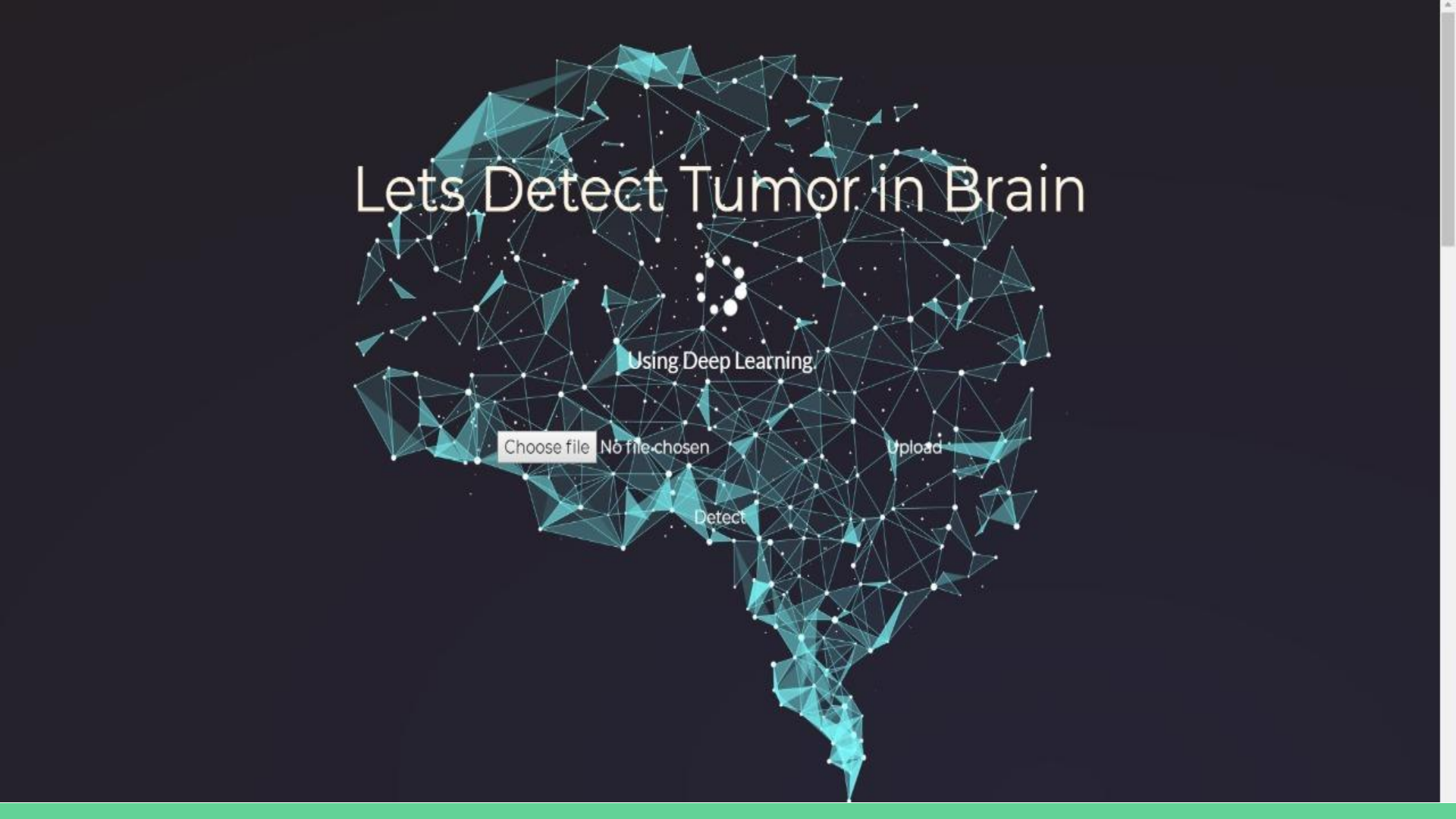
Output layer contains the label which is in the form of one-hot encoded.

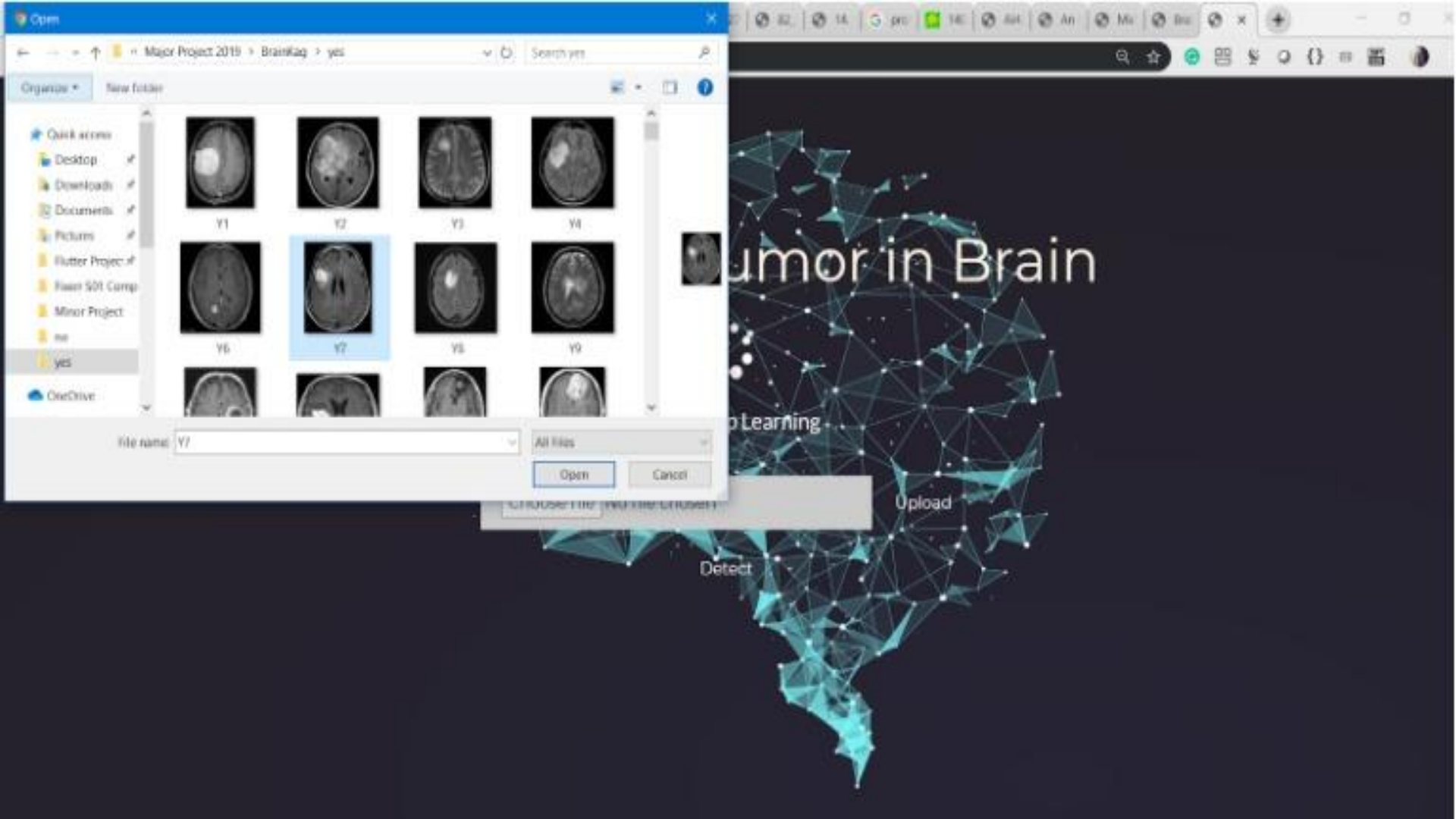
CNN


Neural Network Architecture
x
Input Image:
(240, 240, 3)
ZEROPADDING2D
padding=(2, 2)
(244, 244, 3)
conv0
bn0
CONV2D
32 filters
f=(7, 7)
s=1
Batch
Norm
ReLU
(238, 238, 32)
max_pool0
MAX
POOL
pool_size
=(4, 4)
(59, 59, 32)
max_pool1
MAX
POOL
pool_size
=(4, 4)
(14, 14, 32)
fc
Flatten
(1, 6272)
Dense
(1)
y
Output:
1 represents tumorous class
0 represents the non-tumorous class

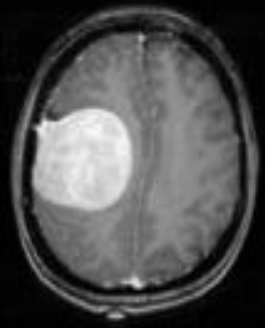
User Interface:-

- We designed a simple website for neurologist , which classify the MRI image on the basis of presence of tumor.
- the user selects an image or file with DICOM or JPG Extention and upload it into the database. In the following, when the user clicks on the Detect button then a pop-up shows the classification with the uploaded image. For this GUI, we used the Django framework and HTML/CSS.


Lets Detect Tumor in Brain
Using Deep Learning
Choose file
No file chosen
Upload
Detect


Open
Major Project 2019 › BrainKaq › yes
Search yes
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
Flutter Project
Minor Project
yes
OneDrive
Y1
Y2
Y3
Y4
Y6
Y7
Y8
Y9
File name: Y7
All Files
Open
Cancel
umor in Brain
Learning
Upload
Detect

Oh No! Brain Tumor Detected

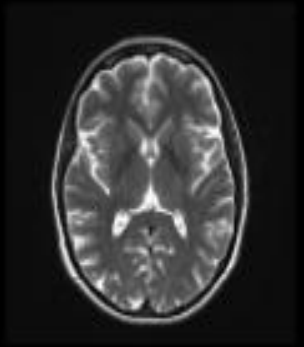


x

Detect

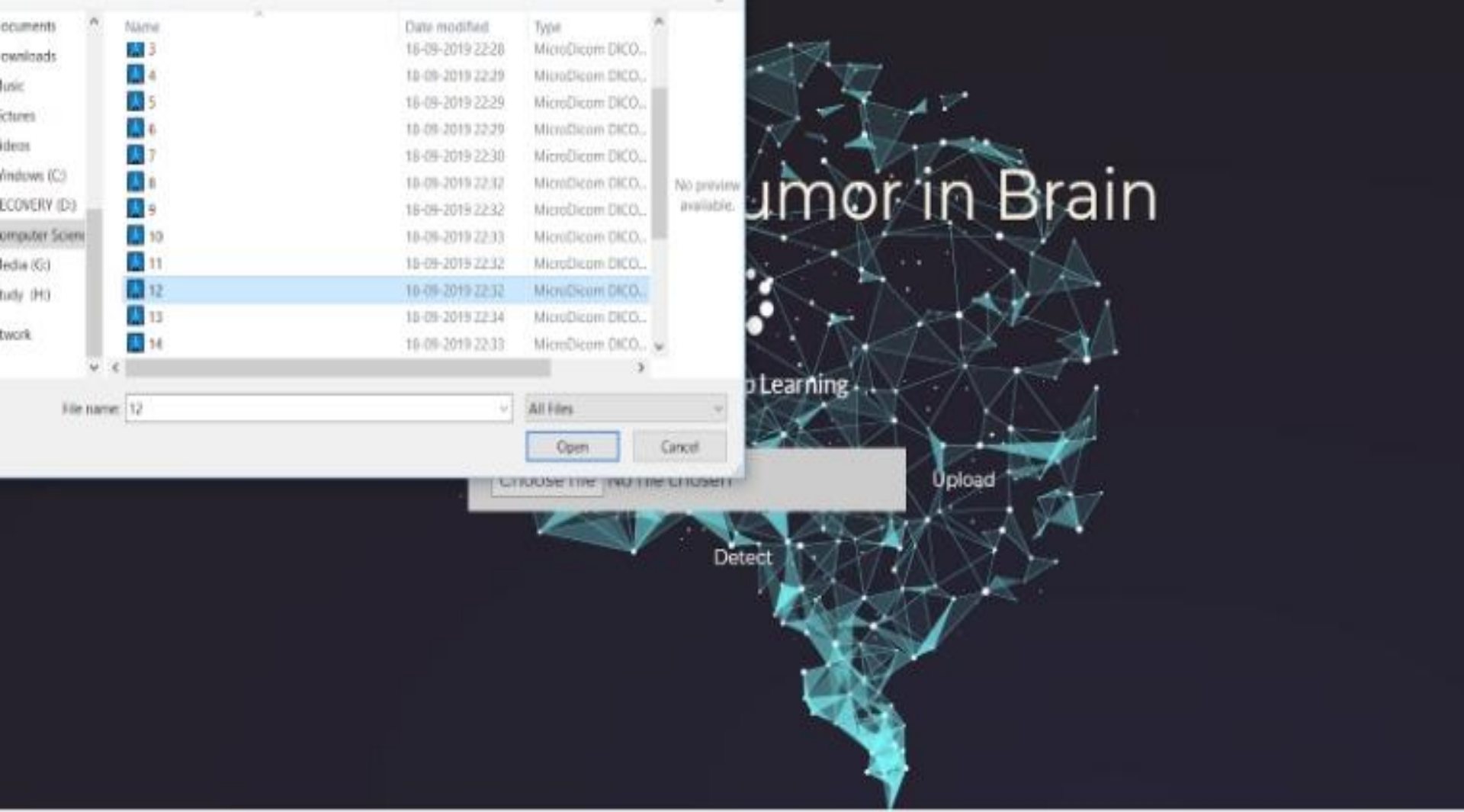
Congratulations! Tumor is a myth

for you



x

Detect


Name
Date modified
Type
No preview available.
File name: 12
All Files
Open
Cancel
umor in Brain
Learning
Upload
Detect

Oh No! Brain Tumor Detected

x

Detect

Future Scope

In terms of near future

As medical image segmentation plays a very important role in field of Image Guided Surgeries.So will to update this project to find the exact location of the tumor in MRI image.

We will generate automated patient medical report.

Conclusion

What is the conclusion of your experiment? Did the results support your hypothesis or predicted outcome? How will your findings help the area of science you've researched?

What should we do next?

(any suggestions)