International Journal of Innovative Research in Engineering
Volume 3, Issue 4 (July-August 2022), PP: 24-29.
www.theijire.com
ISSN No: 2582-8746

Brain Tumor Detection Using Deep Learning

Akhil Johny[1], Ananthakrishnan K[2], Aswin Anilkumar[3], Divya K S[4]
[1,2,3,4]Adi shankara Institute of Engineering and Technology/ A P J Abdul kalam Technical University, Kerala, India

How to cite this paper:
Akhil Johny[1], Ananthakrishnan K[2], Aswin Anilkumar[3], Divya K S[4], "Brain Tumor Detection Using Deep Learning", IJIRE-V3I04-24-29.

Abstract: *Nowadays, every kind of data is digital data. Healthcare is so advanced that physiological data is stored as digital data once it is created. However, diagnosing diseases is still done by clinicians manually. But Machine Learning can be helpful in this situation. ML has a wide variety of applications which will be useful to detect any patterns of certain diseases within patient electronic healthcare records and it is not subjected to any past experiences like human's. While solving a problem with Machine Learning, we have to create different models and select the best among them. There are plenty of machine learning or deep learning classes we can try.As there's many algorithms and neural network architectures, we have to select the most promising and advanced model first. Before selecting the model , we have to select our data. Medical data has a lot of privacy concerns, so it is not widely available. Also the format and quality of data do arise other challenges like effort to clean and preprocessing. Our Aim is to select the best data available and build models for diagnosing . First , we are focusing on brain tumors and then on Arrhythmia as these abnormalities should be detected earlier and as human inspection on these data, specifically MRI images depends strongly on their experience. Also, operator-assisted classification methods are not practical for large amounts of data and also are not reproducible. Therefore, it is highly desirable to use Machine Learning or computer- aided technologies to address these problem*

Key Word*: Convolutional neural network ,Deep learning , computer aided design ,ConvNet*

I. INTRODUCTION

This A brain tumor is a collection, or mass, of abnormal cells in your brain. The skull, which encloses the brain, is very rigid. Any growth inside such a rigid space can cause problems. Brain tumors can be malignant (cancerous) or benign (noncancerous). When brain tumors grow, the growth might increase the pressure inside the skull. This can cause brain damage, and it could be life threatening. The human body is made up of many organs and brain is the most critical and vital organ of them all. One of the common reasons for dysfunction of brain is brain tumor. A tumor is nothing but excess cells growing in an uncontrolled manner. Brain tumor cells grow in a way that they eventually take up all the nutrients meant for the healthy cells and tissues, which results in brain failure. Currently, doctors locate the position and the area of brain tumor by looking at the MR Images of the brain of the patient manually. This results in inaccurate detection of the tumor and is considered very time consuming. A Brain Cancer is very critical disease which causes deaths of many individuals. The brain tumor detection and classification system is available so that it can be diagnosed at early stages. Cancer classification is the most challenging tasks in clinical diagnosis. This project deals with such a system, which uses computer, based procedures to detect tumor blocks and classify the type of tumor using Convolution Neural Network Algorithm for MRI images of different patients. Different types of image processing techniques like image segmentation, image enhancement and feature extraction are used for the brain tumor detection in the MRI images of the cancer-affected patients. Detecting Brain tumor using Image Processing techniques its involves the four stages is Image Pre-Processing, Image segmentation, Feature Extraction, and Classification. Image processing and neural network techniques are used for improve the performance of detecting and classifying brain tumor in MRI images.

The brain tumors are classified into mainly two types: Primary brain tumor (benign tumor) and secondary brain tumor (malignant tumor).The benign tumor is one type of cell grows slowly in the brain and type of brain tumor is gliomas. It originates from non neuronal brain cells called astrocytes. Basically primary tumors are less aggressive but these tumors have much pressure on the brain and because of that, brain stops working properly [6]. The secondary tumors are more aggressive and more quick to spread into other tissue. Secondary brain tumor originates through other part of the body. These type of tumor have a cancer cell in the body that is metastatic which spread into different areas of the body like brain, lungs etc. Secondary brain tumor is very malignant. The reason of secondary brain tumor cause is mainly due to lungs cancer, kidney cancer, bladder cancer etc

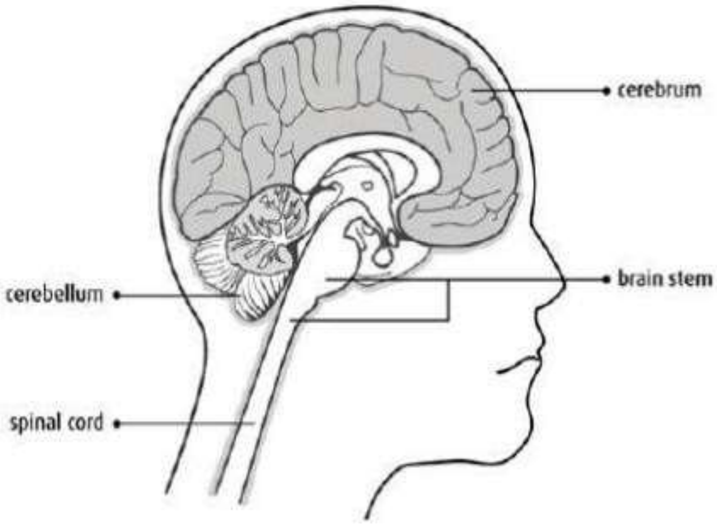

Fig.1: Basic Structure of human brain

II. METHODOLOGY

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image andbe able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, withenough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

1. STRATEGY

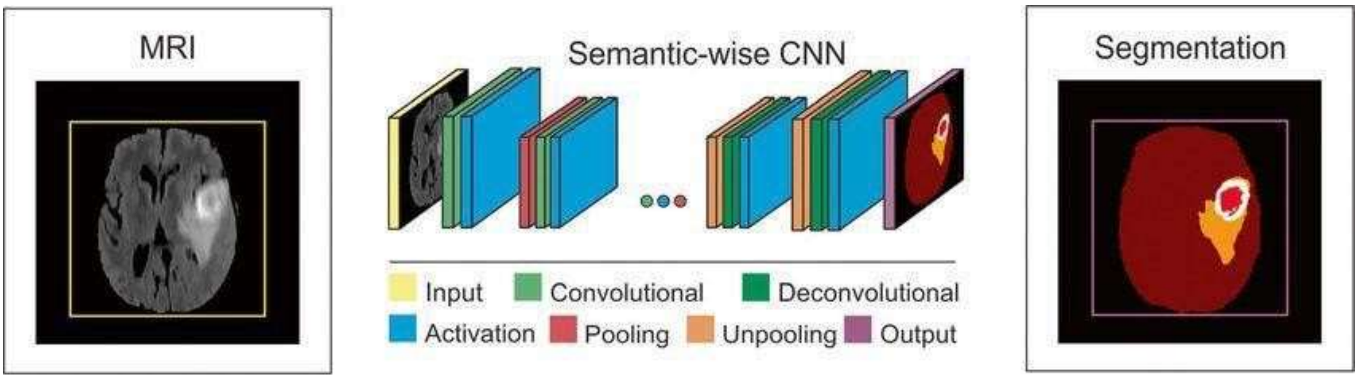

Fig.2:Working of CNN model for brain tumor detection

➢ **Layer of CNN model:**

o Convolution 2D

o MAX Poolig2D

o Dropout

o Flatten

o Dense

o Activation

- **Convolution 2D:** In the Convolution 2D extract the featured from inputimage. It given the output in matrix form.
- **MAX Poolig2D:** In the MAX polling 2D it take the largest element from rectifiedfeature map.
- **Dropout:** Dropout is randomly selected neurons are ignored during training.
- **Flatten:** Flatten feed output into fully connected layer. It gives data in list form.
- **Dense:** A Linear operation in which every input is connected to every output byweight. It followed by nonlinear activation function.
- **Activation:** It used Sigmoid function and predict the probability 0 and 1.
- In the compile model we used binary cross entropy because we have two layers 0 and 1.
- We used Adam optimizer in compile model.
- It used for non-convex optimization problem like straight forward to implement.
- Computationally efficient.
- Little memory requirement.

III.TECHNOLOGY USED

- Python: Python was the language of selection for this project. This was a straightforward call for manyreasons.

➢ Python as a language has a vast community behind it. Any problems which may be faced is simply resolved with a visit to Stack Overflow. Python is among the foremost standard language on the positioning that makes it very likely there will be straight answer to any question.

➢ Python has an abundance of powerful tools prepared for scientific computing Packages like NumPy, Pandas and SciPy area unit freely available and well documented. Packages like these will dramatically scale back, and change the code required to write a given program. This makes iteration fast.

➢ Python as a language is forgiving and permits for program that appear as if pseudo code. This can be helpful once pseudo code given in tutorial papers must be enforced and tested. Using python this step is sometimes fairly trivial. However, Python is not without its errors. The language is dynamically written and packages are area unit infamous for Duck writing. This may be frustrating once a package techniquereturns one thing that, for instance, looks like an array instead of being an actual array. Plus, the actual fact that standard Python documentation does not clearly state the return type of a method, this can lead to a lot of trials and error testing that will not otherwise happen in a powerfully written language. This isa problem that produces learning to use a replacement Python package or library more difficult than it otherwise may be.

➢ <u>Noise Removal and Sharpening:</u> Unwanted data of element are remove using filter and image Can besharpen and black and white gray scale image is used as a input.

➢ <u>Erosion and Dilation:</u> It is applied to binary image, but there are many versions so that can be work on grayscale images. The basic effect of the operator on a binary image is eroding away to the boundaries of regions for ground pixels.

➢ <u>Negation:</u> A negative is an image, usually it used on a strip or sheet of transparent plastic film, in negation the lightest areas of the photographed subject appear darkest and the darkest areas appear lightest.
➢ Subtraction: Image subtraction process is the digital numeric value of one pixel or whole image is subtracted from another image. The white part of tumor can be subtracted from another remaining part that is the black portion of the images.
➢ <u>Threshold:</u> Thresholding is a process of image segmentation. It converts the gray scale image into binary image.

➢ <u>Boundary Detection:</u> Total area or boundary can be form properly using boundary detection method. White part of tumor tissues can be highlighted and there proper boundary can be detected. It is useful method to calculate the size and shape occupy by tumor tissues.

IV.RESULT

- **Give the label of the image**

```
<built-in function dir>
/content/drive/My Drive/brain_dataset/yes
X_data shape: (235, 224, 224, 3)
y_data shape: (235,)
<built-in function dir>
/content/drive/My Drive/brain_dataset/no
X_data shape: (413, 224, 224, 3)
y_data shape: (413,)
```

Fig.11.Label of the image

These outputs in images are resized and give label name to all images.

- Split the Data

```
X_data shape: (330, 224, 224, 3)
X_data shape: (83, 224, 224, 3)
Y_data shape: (330,)
Y_data shape: (83,)
```

Fig.12.Split the image data

Fig 12. Contain Total 413 dataset images are divided into two parts 330 are in training part and 83 is the testingpart.

- Train Data

```
Epoch 245/250
15/15 [==============================] - 48s 3s/step - loss: 0.4972 - accuracy: 0.7542 - val_loss: 0.5175 - val_accuracy: 0.8485
Epoch 246/250
15/15 [==============================] - 48s 3s/step - loss: 0.5008 - accuracy: 0.7374 - val_loss: 0.5198 - val_accuracy: 0.7879
Epoch 247/250
15/15 [==============================] - 52s 3s/step - loss: 0.4825 - accuracy: 0.7879 - val_loss: 0.5250 - val_accuracy: 0.7879
Epoch 248/250
15/15 [==============================] - 48s 3s/step - loss: 0.4977 - accuracy: 0.7407 - val_loss: 0.5130 - val_accuracy: 0.8182
Epoch 249/250
15/15 [==============================] - 48s 3s/step - loss: 0.4789 - accuracy: 0.7542 - val_loss: 0.5168 - val_accuracy: 0.8788
Epoch 250/250
15/15 [==============================] - 48s 3s/step - loss: 0.4849 - accuracy: 0.7441 - val_loss: 0.5262 - val_accuracy: 0.7879
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_ops.py:1817: calling BaseResourceVari
Instructions for updating:
If using Keras pass * constraint arguments to layers.
```

Fig.13.Train CNN image data

Fig . 13. Consist output of train the convolutional neural network. Train 330 samples and validate on 150 samples

- Test Data

```
scores=model.evaluate(xTest, yTest)
print("%s: %2f%%" %(model.metrics_names[1], scores[1]*100))
```

```
3/3 [==============================] - 2s 808ms/step - loss: 0.5499 - accuracy: 0.8072
accuracy: 80.722892%
```

Fig.14.Test CNN image Data

Fig 14. Consist output of convolutional neural network testing accuracy score 80.72%

- Train Data

```
Epoch 65/70
20/20 [==============================] - 338s 17s/step - loss: 0.5633 - accuracy: 0.7473
Epoch 66/70
20/20 [==============================] - 329s 16s/step - loss: 0.5276 - accuracy: 0.7677
Epoch 67/70
20/20 [==============================] - 333s 17s/step - loss: 0.5474 - accuracy: 0.7382
Epoch 68/70
20/20 [==============================] - 329s 16s/step - loss: 0.5726 - accuracy: 0.7524
Epoch 69/70
20/20 [==============================] - 340s 17s/step - loss: 0.5436 - accuracy: 0.7598
Epoch 70/70
20/20 [==============================] - 326s 16s/step - loss: 0.5467 - accuracy: 0.7587
```

Fig.15.Test VGG16 image Data

Fig . 15. Consist output of train the VGG 16 Transfer Learning model. Train 330 samples and validate on150 samples

- Test Data

```
model.save('braintransfer-VGG70.model')
scores=model.evaluate(xTest, yTest)
print("%s: %2f%%" %(model.metrics_names[1], scores[1]*100))
```

```
83/83 [==============================] - 42s 500ms/step
accuracy: 85.542166%
```

Fig.16.Test VGG16 image Data

Fig 16. Consist output of VGG 16 testing accuracy score 85.54%

- Implementation: CNN model summary

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_4 (MaxPooling2	(None, 127, 127, 32)	0
dropout_5 (Dropout)	(None, 127, 127, 32)	0
conv2d_5 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_5 (MaxPooling2	(None, 62, 62, 64)	0
dropout_6 (Dropout)	(None, 62, 62, 64)	0
conv2d_6 (Conv2D)	(None, 60, 60, 128)	73856
activation_3 (Activation)	(None, 60, 60, 128)	0
max_pooling2d_6 (MaxPooling2	(None, 30, 30, 128)	0
dropout_7 (Dropout)	(None, 30, 30, 128)	0
conv2d_7 (Conv2D)	(None, 28, 28, 512)	590336
activation_4 (Activation)	(None, 28, 28, 512)	0
max_pooling2d_7 (MaxPooling2	(None, 14, 14, 512)	0
dropout_8 (Dropout)	(None, 14, 14, 512)	0
flatten_1 (Flatten)	(None, 100352)	0
dense_2 (Dense)	(None, 64)	6422592
dropout_9 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65
activation_5 (Activation)	(None, 1)	0

Total params: 7,106,241
Trainable params: 7,106,241
Non-trainable params: 0

None

Table.1. CNN model summary table

VI. FUTURE WORK

It should very effective if this system can be implemented in hospital usages . The need for systems like this is increasing day by day because we all are living in a busy world . This should help the peoples to check weather brain tumor is there or not from the safeness of their own home .We can classify the brain tumor into mainly 3 types . Glioma, Meningioma , pituitary tumor.This can make the users to take preventive measures from the beginning of the disease.

References

1. *Image Segmentation Algorithms Overview -Song Yuheng , Yan Ha*
2. *U-Net: Convolutional Networks for Biomedical Image Segmentation -Olaf Ronneberger, Philipp Fischer, and Thomas Brox*
3. *Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition -Jun Cheng , Wei Huang , Shuangliang Cao , Ru Yang , Wei Yang , Zhaoqiang Yun , Zhijian Wang , Qianjin Feng*
4. *Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network Milica M. Badža and Marko C. Barjaktarovi´c ˇ*
5. *An Empirical Study of Oversampling and Undersampling for Instance Selection Methods on Imbalance Datasets -Julio Hernandez, Jes´us Ariel Carrasco-Ochoa, and Jos´e Francisco Mart´ınez-Trinidad*