

Brain Tumor Detection From MRI Image Using Deep Learning

Divya Pathak¹, Dadi Ganesh², Gosu Yogeswar³, Dodda Pavan Chandra⁴, P. Karunakar⁵

^{1,2,3,4} UG Student Dept. of EEE PVPSIT, Vijayawada, Andhra Pradesh, India- 520 007

⁵Assistant Professor, Dept. of EEE PVPSIT, Vijayawada, Andhra Pradesh, India- 520 007

Abstract— Identification of brain tumors is one of the most important and challenging tasks in medical imaging because manual human-assisted classification can lead to inaccurate predictions and diagnoses. Medicine requires fast, automated and reliable technology for tumor detection. Applying a deep learning approach in the context of improving health diagnosis provides an effective solution. This experimental work was performed with a dataset consisting of magnetic resonance imaging (MRI) images of tumors of various shapes, sizes, textures and locations. MRI scans reveal information about abnormal tissue growth in the brain. A self defined Convolution Neural Network (CNN) deep learning architecture is used for the classification of image as tumor or non-tumor. In our work, CNN model without transfer learning technique gained an accuracy of 81.42% and with transfer learning the accuracy reached to 98.8% which is very compelling.

Keywords—Convolution Neural Network, Magnetic Resonance Imaging, Transfer Learning.

I. INTRODUCTION

Medical imaging is a method and process for generating visual representations of the functions of specific organs or tissues as well as visual representations of the interior of the body for clinical analysis and medical intervention. Medical imaging aims to diagnose and treat disease by revealing the internal structures hidden in the skin and bones. Medical imaging also creates a database of normal anatomy and physiology that enables anomaly detection.

Image processing technology is the manipulation of digital images using computers. This method has many advantages such as elasticity, adaptability, data storage and communication. An increasing number of different image resizing methods allow you to store images efficiently. This method has a large set of rules for synchronously running the image. Multidimensional processing of 2D and 3D images is possible.

According to [1], cancer of the brain and other nervous system is the 10th leading cause of death, and the 5-year survival rate for brain cancer patients is 34% for men and 36% for women. Moreover, the World Health Organization (WHO) claims that around 400,000 people worldwide have a brain tumor, and 120,000 people have died in the past few years. In addition, it is expected that approximately 86,970 new cases of primary malignant

and benign brain tumors and other central nervous system (CNS) tumors will be diagnosed in the United States in 2019 [2].

Hence early detection of brain tumor is very important. In this paper, we proposed a Convolution Neural Network architecture with and without transfer learning and achieved an accuracy of 98.8%.

II. LITERATURE REVIEW

In medical diagnosis, the robustness and accuracy of prediction algorithms are important, as the results are crucial to the treatment of the patient. There are many popular classification and clustering algorithms used for prediction. The purpose of medical image clustering is to simplify the representation of an image into a meaningful one and make it easier to analyze. Several clustering and classification algorithms aim to improve the predictive accuracy of the diagnostic process in anomaly detection.

Devkota et al. [3] established the whole segmentation process based on the mathematical morphological operations and the spatial FCM algorithm to improve the computation time, but the proposed solution has not been tested until the evaluation stage. prices and results because it detects cancer with 92% and the classifier has an accuracy of 86.6%. Yantao et al. [4] is similar to the histogram-based segmentation technique. Treat the brain tumor segmentation task as a three-class classification problem (tumor including necrotic tumor and tumor, edema and normal tissue) involving two modalities FLAIR and T1. Abnormal regions were detected using a region-based active contour model on the FLAIR method. Edema and tumor tissues were differentiated into abnormal regions based on T1 contrast enhancement by kmeans method and obtained a Dice coefficient and sensitivity of 73.6% and 90.3%, respectively.

Yu et al. [5] states that image segmentation is used to extract important objects from an image. They propose to divide an image into three parts, including black, gray and white. The Z function and the s function are used for fuzzy division of the 2D histogram. Then, QGA is used to find the combination of 12 parameters belonging to, which has the largest value. This technique is used to improve the image segmentation and the implication of their work is that the

three-level image segmentation is used by following the maximum fuzzy partitioning of the 2D histogram. QGA was selected 11 for optimal matching of parameters with fuzzy partition entropy. The proposed method of 2D histogram fuzzy partition entropy gives better performance of than the 3-level one-way thresholding method. One way or another, a large number of possible combinations of 12 parameters in a multidimensional fuzzy partition are used, and it is practically impossible to compute every possible value; so that QGA can be used to find the optimal combination

[6] Nowadays, automatic detection of defects on MRI images is very important in many diagnostic and therapeutic applications. This paper presents a novel automated brain tumor detection method that uses T1-, T2- and PD-weighted MRI images to identify any abnormalities in brain tissue. Here we have tried to give a clear description of brain tissue using Gabor wavelets, energy, entropy, contrast and some other statistical features like mean, median, mean false, correlation, maximum and minimum intensity values. It is also used from an object selection method to reduce object space. This method uses a neural network to perform this classification. The goal of this project is to automatically classify brain tissue into normal and abnormal classes, saving radiologist time, increases accuracy and yield of diagnosis.

III. PROPOSED METHODOLOGY

In our proposed methodology, there are two distinct model for classification of Brain tumor. First model detects the tumor by VGG 6 layer convolution architecture without transfer learning and the second model detects the tumor by passing the input image to VGG16 layered architecture.

Convolutional neural networks are widely used in medical image processing. Over the years, many researchers have tried to build a model that can more effectively detect tumors. We attempted to propose a specimen capable of accurately classifying tumors from 2D brain MRI images. A fully connected neural network can detect tumors, but because of the sharing of parameters and sparse connectivity, we used CNN for our model.

[7] Transfer learning is a method for feature representation from a pre-trained model facilitating us that we don't need to train a new model from scratch. A pre-trained model is usually trained on a huge dataset such as ImageNet and the weights obtained from the trained model can be used for any other related application with your custom neural network. These newly built models can directly be used for predictions on relatively new tasks or can be used in training processes for related applications. This approach not only reduces the training time but also lowers the generalization error.

A. Brain MRI data Acquisition

The brain MRI image dataset was downloaded from the Kaggle website. The MRI dataset includes approximately 5318 MRI images including normal, benign, and malignant. These MRI images were taken as input for primary stage. MRI images are available in JPG and TIF formats. The training to validation ratio is taken as 80:20. Table 1 shows detailed information about the number of images.

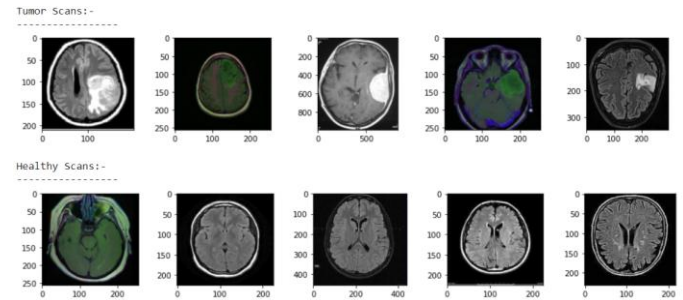


Fig. 1. MRI Image of Brain (Tumorous and Healthy)

Fig. 1 Represent the 5 sample MRI images of normal and tumor images respectively. For Performance Evaluation of our proposed model, we used the BRATS dataset, consisting two classes— class-0 and class-1 represents the Non-Tumor and Tumor MRI images. It consists of total 253 images

TABLE 1. Dataset Information

Type	Train Data	Validation Data	Test Data
Non-tumor	2126	502	98
Tumor	2130	560	155
Total	4256	1062	253

B. Data Preprocessing

These pre-processing techniques consist of filtering, image detection, and image enhancement mentioned in figure. To enhance and smooth the image while processing the Convolutional neural network pre-processing is used. Preprocessing is a necessary and initial step in improving the quality of MRI images of the brain. In image processing, image acquisition is done by retrieving an image from dataset for processing. It is the first step in the workflow sequence because, without an image no processing is possible. The image that is acquired is completely unprocessed. Here we process the image using the file path from the local device.

$$\text{Steps_per_epoch} = \text{Total Image} / \text{batch size}$$

TABLE 2. HYPERPARAMETER VALUE

Hyper-parameter	Value
Batch_size	32
Epoch	30
Target_size	(224,224)
Steps_per_epoch (training data)	133
Steps_per_epoch (validation data)	34

i. Image Preprocessing Without Transfer Learning

The original MR image of train dataset is pre-processed by the following parameters before passing to the convolution neural network :

TABLE 3. PREPROCESSING PARAMETER VALUE

Parameter	Value
zoom_range	0.2
Shear_range	0.2
rescaling	1/255
Horizontal_flip	True

Whereas the validation and test dataset is normalized by rescaling by 1/255.

Fig. 2. Shows the original image of tumor and Fig. 3. Shows the image after preprocessing.

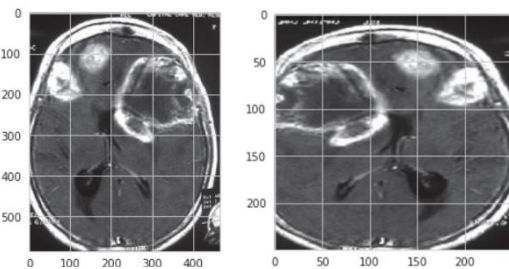


Fig. 2.Original Image Fig. 3.Preprocessed Image

ii. With Transfer Learning

The original MR image of train dataset is pre-processed by the following parameters:

TABLE 4. PREPROCESSING PARAMETER VALUE

Parameter	Value
zoom_range	0.2
Width_shift_range	0.1
Height_shift_range	0.1
Shear_range	0.2
preprocessing_function	preprocess_input
Horizontal_flip	True

Preprocessing input is pre-trained model used in VGG16 algorithm. Whereas the validation and test dataset preprocessing_function: preprocess_input

Fig. 4. Shows the original image of tumor and Fig. 5. Shows the image after preprocessing.

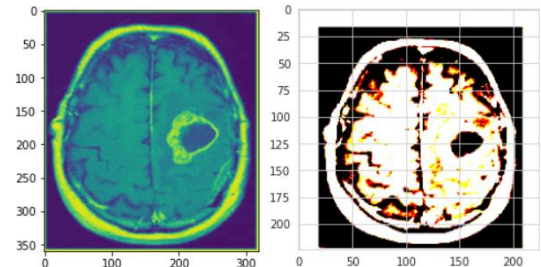


Fig. 4.Original Image Fig. 5.Preprocessed Image

C. Convolutional Neural Network Architecture

Convolutional neural networks are widely used in medical image processing. Over the years, many researchers have tried to build a model that can more effectively detect tumors. We attempted to propose a specimen capable of accurately classifying tumors from 2D brain MRI images. A fully connected neural network can detect tumors, but because of the sharing of parameters and sparse connectivity, we used CNN for our model.

To initialize the neural network, we create an object of the Sequential class. The layers of CNN are as follows:

A. CONVOLUTION:

This method creates a compound class. The first parameter is the number of filters and the second is the size of the filter. For example, in the first convolution layer, we create 32 filters of size 3x3. We use relu nonlinearity as a trigger. We also enable padding. In Keras, there are two choices for padding: same or valid. Even means we fill with numbers on the edge and valid means no fill. Stride is 1 for convolutional layers by default, so we won't change that. This class can be further customized with additional settings.

B. POOLING:

The Pooling layer is responsible for reducing the spatial size of the convolved feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling.

Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other

hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Generally, we use max pooling. In this step we reduce the size of the feature map.

Generally, we create a pool size of 2x2 for max pooling. This enables us to reduce the size of the feature map while not losing important image information

C. FLATTEN:

In this step, all grouped feature maps are taken and placed into a single vector for import into the next layer. The Flatten function flattens all feature maps into a single long column.

D. FULLY CONNECTED:

The next step is to use the vector we got above as input to the neural network using the Dense function in Keras. The first parameter is the output which is the number of nodes in the hidden layer. You can determine the most appropriate number by experiment. The higher the number of dimensions, the more computing resources you will need to fit the model. It is a common practice to choose the number of nodes by powers of two.

E. DROPOUT:

Dropout is by far the most popular regularization technique for deep neural networks. Even the state-of-the-art models which have 95% accuracy get a 2% accuracy boost just by adding dropout, which is a fairly substantial gain at that level. Dropout is used to prevent overfitting and the idea is very simple. During training time, at each iteration, a neuron is temporarily “dropped” or disabled with probability *p*.

i. *Without Transfer Learning*

This VGG (Visual Geometry Group) consist of six convolution layer, three max-pooling 2D layer and one flatten layer and two dropout layer to avoid overfitting. The summary of this layer is shown below.

TABLE 5. Model Summary

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 16)	448
conv2d_1 (Conv2D)	(None, 220, 220, 36)	5220
max_pooling2d	(None, 110, 110, 36)	0
conv2d_2 (Conv2D)	(None, 108, 108, 64)	20800

max_pooling2d_1	(None, 54, 54, 64)	0
conv2d_3 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_2	(None, 26, 26, 128)	0
dropout (Dropout)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 64)	5537856
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

=====
 =====
 Total params: 5,638,245
 Trainable params: 5,638,245
 Non-trainable params: 0

ii. *With Transfer Learning*

The VGG16 (Visual Geometry Group) consist of sixteen convolution layer, five max-pooling 2D layer and one flatten layer. The summary of this layer is shown below.

TABLE 6. Model Summary

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080

block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 128)	3211392
dense_1 (Dense)	(None, 1)	129

=====

TABLE 7. Summary of trained model

Epoch	Loss	Validation Loss	Accuracy	Validation Accuracy
1	0.8535	0.6880	0.4727	0.5527
2	0.6750	0.6653	0.5742	0.5605
3	0.6880	0.6440	0.5664	0.5898
4	0.6636	0.6435	0.5547	0.5938
5	0.6437	0.6514	0.6328	0.6016
6	0.6289	0.6253	0.6428	0.6152
7	0.5860	0.5923	0.6484	0.6465
8	0.6218	0.5914	0.6875	0.6914
9	0.5876	0.5587	0.7031	0.7227
10	0.5612	0.5480	0.7227	0.7090
11	0.5474	0.5687	0.7305	0.7324
12	0.5533	0.5078	0.7344	0.7598
13	0.5663	0.5276	0.7461	0.7441
14	0.5811	0.5313	0.7070	0.7500
15	0.5074	0.5139	0.7812	0.7188
16	0.5387	0.5049	0.7266	0.7578
17	0.4985	0.5407	0.7812	0.7227
18	0.5645	0.4833	0.7422	0.7734
19	0.5172	0.5031	0.7500	0.7637
20	0.4936	0.5278	0.7930	0.7539
21	0.5234	0.5127	0.7539	0.7500
22	0.4703	0.4799	0.7930	0.7715
23	0.4894	0.4784	0.7773	0.7500
24	0.5113	0.4568	0.7500	0.7988
25	0.4794	0.4372	0.7852	0.7891
26	0.4286	0.5754	0.8164	0.7461
27	0.5382	0.4658	0.7383	0.7832
28	0.4920	0.4742	0.7930	0.8027
29	0.5490	0.4531	0.7109	0.7891
30	0.5015	0.4238	0.7734	0.8242

Total params: 17,926,209
 Trainable params: 3,211,521
 Non-trainable params: 14,714,688

IV. RESULTS

The result consist of both train and validation dataset loss and accuracy information at specific epoch. Validation dataset is used in order to ensure that the model is stable and not overfitted. Once the model is trained then test dataset is evaluated.

A. Without Transfer Learning Model Result

The 6 convolution layer model is used for the training. Loss and accuracy for both training and validation dataset is tabulated below. Epoch 26 has the best accuracy of 81% and loss of 0.4286.

Fig. 6 Represents the performance curve of the model. The performance curve are epoch vs binary cross-entropy loss and epoch vs accuracy.

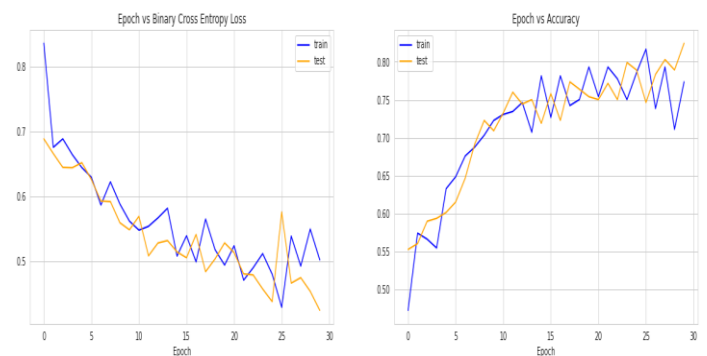


Fig. 6. Model Performance Curve

The summary of model that is evaluated on the test dataset is as follows:

Loss: 0.576 and Accuracy: 81.42%

B. With Transfer Learning Model Result

The VGG16 transfer learning model is used and the loss and accuracy of both training and validation dataset for corresponding epoch is tabulated below. Epoch 28 has the best accuracy of 99% and loss of 0.0289.

TABLE 8. Summary of trained model

Epoch	Loss	Validation Loss	Accuracy	Validation Accuracy
1	2.2774	1.1614	0.8584	0.9097
2	0.6008	0.1479	0.9169	0.9522
3	0.1972	0.0926	0.9410	0.9756
4	0.1356	0.0673	0.9549	0.9745
5	0.1139	0.0991	0.9609	0.9681
6	0.0807	0.0513	0.9721	0.9819
7	0.0856	0.0644	0.9697	0.9787
8	0.1099	0.0621	0.9653	0.9830
9	0.1064	0.0456	0.9653	0.9809
10	0.0638	0.1050	0.9765	0.9756
11	0.0515	0.0456	0.9828	0.9841
12	0.0455	0.0400	0.9852	0.9883
13	0.0456	0.0506	0.9855	0.9894
14	0.0366	0.0663	0.9872	0.9830
15	0.0402	0.0514	0.9841	0.9872
16	0.0369	0.0683	0.9885	0.9830
17	0.0568	0.0425	0.9861	0.9904
18	0.0427	0.0427	0.9855	0.9883
19	0.0387	0.0382	0.9858	0.9894
20	0.0328	0.0489	0.9874	0.9862
21	0.0437	0.0443	0.9855	0.9862
22	0.0404	0.0534	0.9855	0.9862
23	0.0311	0.0506	0.9910	0.9841
24	0.0374	0.0466	0.9847	0.9862
25	0.0438	0.0678	0.9872	0.9851
26	0.0333	0.0465	0.9885	0.9851
27	0.0343	0.0664	0.9802	0.9841
28	0.0289	0.0531	0.9907	0.9872
29	0.0369	0.0815	0.9869	0.9851
30	0.0459	0.0574	0.9882	0.9862

Fig. 7 Represents the performance curve of the model. The performance curve are epoch vs binary cross-entropy loss and epoch vs accuracy.

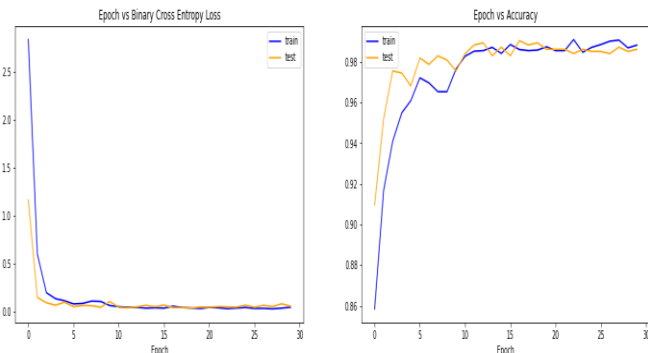


Fig. 7. Model Performance Curve

The summary of model that is evaluated on the test dataset is as Follows:
Loss: 0.1576 and Accuracy: 98.81%

C. Comparative Analysis

In this study, we have compared our model with the previous best model which is VGG with 6 convolution layer with VGG16 convolution architecture. Our model has shown reliable results with improved accuracy and reduced loss. This is a better model for classification of brain MR image compared with the previous model. The results that we have obtained are as follows.

TABLE 9. Summary of proposed model

Comparative Analysis				
Model Epoch	With Transfer Learning		Without Transfer Learning	
	Loss	Accuracy	Loss	Accuracy
1	2.2774	0.8584	0.8535	0.4727
2	0.6008	0.9169	0.6750	0.5742
3	0.1972	0.9410	0.6880	0.5664
4	0.1356	0.9549	0.6636	0.5547
5	0.1139	0.9609	0.6437	0.6328
6	0.0807	0.9721	0.6289	0.6428
7	0.0856	0.9697	0.5860	0.6484
8	0.1099	0.9653	0.6218	0.6875
9	0.1064	0.9653	0.5876	0.7031
10	0.0638	0.9765	0.5612	0.7227
11	0.0515	0.9828	0.5474	0.7305
12	0.0455	0.9852	0.5533	0.7344
13	0.0456	0.9855	0.5663	0.7461
14	0.0366	0.9872	0.5811	0.7070
15	0.0402	0.9841	0.5074	0.7812
16	0.0369	0.9885	0.5387	0.7266
17	0.0568	0.9861	0.4985	0.7812
18	0.0427	0.9855	0.5645	0.7422
19	0.0387	0.9858	0.5172	0.7500
20	0.0328	0.9874	0.4936	0.7930
21	0.0437	0.9855	0.5234	0.7539
22	0.0404	0.9855	0.4703	0.7930
23	0.0311	0.9910	0.4894	0.7773
24	0.0374	0.9847	0.5113	0.7500
25	0.0438	0.9872	0.4794	0.7852
26	0.0333	0.9885	0.4286	0.8164
27	0.0343	0.9802	0.5382	0.7383
28	0.0289	0.9907	0.4920	0.7930
29	0.0369	0.9869	0.5490	0.7109
30	0.0459	0.9882	0.5015	0.7734

V. CONCLUSION

We have proposed a computer-based method to identify brain tumors using an Convolution neural network. The input MR image is read from the local device using the file path, then the input image is preprocessed by zooming, cropping, scaling and flipping. These images are passed through VGG (Visual Geometry Group) with 6 layers of

convolution applied, which helps to classify images as tumor or non-tumor. This proposed model has an accuracy of 81.42%.

Also, for better results, we have implemented VGG16 architecture with transfer learning technique which gives 98.8% accuracy and gives promising results without any error and much less computer downtime.

TABLE 10. PERFORMANCE COMPARISON

Methodology	Accuracy (%)
CNN Model without Transfer Learning	81.42
CNN Model with Transfer Learning	98.8

REFERENCES

- [1] Brain Tumor: Statistics, Cancer.Net Editorial Board, 11/2017
- [2] General Information About Adult Brain Tumors". *NCI*. 14 April 2014. Archived from the original on 5 July 2014. Retrieved 8 June 2014.
- [3] B. Devkota, Abeer Alsadoon, P.W.C. Prasad, A. K. Singh, A. Elchouemi, "Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction," 6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India
- [4] Song, Yantao & Ji, Zexuan & Sun, Quansen & Yuhui, Zheng. (2016). "A Novel Brain Tumor Segmentation from Multi-Modality MRI via A Level-Set-Based Model". *Journal of Signal Processing Systems*. 87. 10.1007/s11265-016-1188-4.
- [5] P. Naga Srinivasu, T. Srinivasa Rao, Valentina Emilia Balas. (2020). A systematic approach for identification of tumor regions in the human brain through HARIS algorithm, *Deep Learning Techniques for Biomedical and Health Informatics*, Academic Press. Pages 97-118. <https://doi.org/10.1016/B978-0-12-819061-6.00004-5>.
- [6] A Neural Network-based Method for Brain Abnormality Detection in MR Images Using Gabor Wavelets.
- [7] Vijaysinh, "A comparison of 4 Propular Transfer Learning Model," Sep' 2021 Developer Corners.