Alfarama Journal of Basic & Applied Sciences

Faculty of Science Port Said University

https://ajbas.journals.ekb.eg
ajbas@sci.psu.edu.eg

http://sci.psu.edu.eg/en/

October 2023, Volume 4 , Issue IV

DOI:https://doi.org/10.21608/ajbas.2023.216710.1158

ISSN 2682-275X

Submitted: 24/12/2022
Accepted: 12/06/2023

Pages: 664 - 679

An Automatic system to classify MRI brain tumor using Convolutional Neural Network

Aml O. Zaghloul[1,2,*], Noha E. El-Attar[3], Ahmed A. El-Harby [4] and Wael A. Awad [5]

[1] Mathematics and Computer Science Department, Faculty of Science Port-Said University, Port-Said, Egypt.

[2] Institute of Specific Studies and Computer science, Ras Al-Bar.

[3]Faculty of computers and Artificial Intelligence ,Benha University ,Benha ,Egypt.

[4,5]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Damietta University, New Damietta, Egypt.

***Corresponding author: Amlomar782@gmail.com**

ABSTRACT

The brain tumor is regarded as a serious cancerous tumor that if not detected and accurately identified, may lead in the patient's death. Therefore, recent advances in the field of deep learning (DL) have assisted radiologists in diagnosing tumors with high accuracy and speed when compared to manual diagnosis, which requires the radiologist's effort and competence. Oncologists typically perform the initial evaluation of brain tumors using medical imaging techniques such as computerized tomography (CT) and magnetic resonance imaging (MRI). These two medical imaging techniques are commonly used to create highly detailed images of the brain's structure to monitor any changes. A surgical biopsy of the suspected tissue (tumor) is required for a detailed diagnosis by the specialist if the doctor suspects a brain tumor and needs more information about its type. These various techniques in brain tissue imaging have increased image contrast and resolution in recent years, allowing the radiologist to identify even small lesions and thus achieve higher diagnostic accuracy. This research introduced an automatic system using a Convolutional Neural Network (CNN) to classify MRI brain tumor images consisting of various layers, and then selected the best system that achieved an accuracy of ٩٩.٦% with different images sizes and learning rates.

Key Words:

Brain tumor, Convolutional neural network, Deep Learning

1. INTRODUCTION

The brain is the most sensitive and complex primary organ in the human body, controlling basic bodily functions and characteristics such as breathing, muscle movement and our senses. Each cell has its distinct power. Some cells grow normally, while others lose their power, resist and grow abnormally. A tumor is a collection of abnormal cells that make up tissues. [1]. A brain tumor is caused by abnormal and uncontrollable cell growth in the brain. It is divided into two types are malignant and benign [2, 3]. A malignant tumor is a type of cancer, it may spread quickly and invades other parts of the body. It is threatens life with death because it is more aggressive .On the contrary, A benign tumor may begin in the brain, grow slowly and does not spread throughout the body, usually remaining in one area. It is a non-cancerous tumor and can be removed at the appropriate time [4-5].

Manual identification of brain tumors is complex and prone to error by the radiologist; it depends on the expertise and sensitivity of the radiologist. So, it is necessary to rely on an automatic system. Automatic classification is very important to speed up the process of classification and treatment of the patient [6]. Brain tumors are typically evaluated using imaging techniques such as MRI and CT, which produce detailed images of the brain's structure, but doctors prefer the MRI method to classify the tumor. A good tumor diagnosis is very complex, it may require many steps such as physical examination, determination of the location, size and shape of the tumor, surgical excision and finally tissue analysis and a decision about the classification of the tumor. However, identifying the type of tumor by MRI is a difficult, error-prone and time-consuming task. Manual diagnosis is unreliable. Therefore, the field of artificial intelligence (AI) is used as one of an efficient automated diagnosis [7, 8].

This research introduced an automatic CNN system to classify brain tumor. The remainder of this article is arranged as follows: Section 2 provides background information, including related works and illustration of deep learning. The dataset is organized in Section 3. Section 4 introduces the proposed system. Section 5 contains the experimental results and discussion.Comparative study introduced in Section 6. Finally; the conclusion and future work are presented in Section 7.

2. BACKGROUND

2.1 Related Works

Many researchers have been used the same dataset that used in this research for brain tumor classification. Khan et al. [9] introduced linear contrast stretching; discrete cosine transform and used two pre-trained CNN models ,VGG16 and VGG19 together, they achieved an accuracy of 97.8%. The proposed VGG16 including the encoder and decoder networks with a classification layer and accuracy was 97.7% [10]. VGG16 and VGG19 architectures are applied with half filters in the first layer, new models called LBTS-Net16 and LBTS-Net19 respectively. These implementations had accuracy rates of 96.7 % and 95.9 % respectively [11].

The discrete wavelet transform fusion and a partial diffusion filter were employed. They then used a global thresholding algorithm to extract tumor features, which were then passed to the CNN model for

tumor classification, and achieved 96% accuracy [12]. The CNN with large and small kernels called (SK-TPCNN) and used the random forests classifier. Their model achieved classification accuracy 89% [13]. Three different 3D CNN models are 3D fully CNN based on the VGG (3D Net-1), a 3D version of the network that makes connections between corresponding layers in the contracting and expanding paths (3D Net-2) and a modification of the DeepMedic network that makes a combination between CNN layers (3D Net-3), The results obtained was 99.69%, 99.71% and 99.71% respectively [14]. A hybrid deep auto encoder was used in conjunction with a Bayesian fuzzy cluster-based segmentation approach. Finally, the softmax regression technique was used to classify the tumor type. Their classification model was 98.5 % accurate [15]. High and median filters were used to enhance MRI brain images and used a deep wavelet autoencoder model. The resulting accuracy was 99.3% [16]. Deep Residual Dilate Network with Middle Supervision was applied and achieved an accuracy of 86% [17]. Marker Controlled Watershed Mining was used to extract the diseased section from the selected brain Flair MRI images and obtained a satisfactory accuracy of 92 % [18].

Adaptive Momentum, Adaptive Gradient, Adaptive Delta, Stochastic Gradient Descent, Nesterov accelerated gradient, Cyclic Learning Rate, Adaptive Max Pooling, Root Mean Square Propagation, Nesterov Adaptive Momentum and Momentum for CNN were studied in comparison. The best accuracy was achieved by the Adaptive Momentum optimizer, which was 99.2 % [19, 20]. A gray-level co-occurrence matrix and histogram were proposed to extract features to predict the grade of gliomas. They achieved an accuracy of 89.81%. Hybrid statistical and wavelet features were applied on the low and high gliomas grades. They obtained 96.72% for high-grade glioma and 96.04% for low-grade glioma [21]. Non-Sub sampled contourlet transform was used to enhance the brain image, then texture features were extracted, and the brain image was classified as normal or glioma using an Adaptive Neuro-Fuzzy Inference System approach. Using morphological functions, the tumor regions in the glioma brain image are then segmented, achieved an accuracy of 99.30% [22].

2.2 Deep Learning

Generally, deep neural networks are based on the idea of the human brain. It is an artificial neural network with multiple layers and it is a subset of ML but the important advantage of DL over ML is that it can automatically extract appropriate features rom large amounts of data. Following an effective training process, it can classify these features and make an appropriate decision. A weighted average of the total input is computed during the training process. After that, the output is computed by applying a nonlinear activation function and it is a process in which weights are learned for all layers using the reverse propagation algorithm [23-25]. CNN is the most famous type of deep neural networks, it is chosen to be used in this research.

2.2.1 Convolutional Neural Network

It can be used to achieve a great success in the classification field, as it is characterized by the existence of hidden layers and extracting features from images automatically. In the training and test stages, a set of layers are applied [26]. These layers are:

Layer1 (a): Convolution Layer

It is a set of filters that plays an important role in DL networks.The filter moves from top right to left with a specified value stride and from top to bottom until it distributes the full width. Move, and repeat this process until the entire image has been traversed, and extract the image's features map [27,28]. The output equation after applying stride S, given the image of $N \times N$ dimension and the filter size of $F \times F$. The output size O is shown in the following equation.

$$O = 1 + \frac{N-F}{S} \quad (1)$$

Layer 1(b): Batch Normalization Layer

During the training stage, the distribution of the hidden layer's inputs would change constantly. It is referred to as an internal covariate transformation. Generally, during back propagation, the new distribution increasingly approaches the top and lower limits of the interval of the activation function, which might result in the gradient of shallow hidden layers being reduced and disappearing. The distribution of the inputs can be converted to a standard normal distribution using batch normalization with a variance equal to 1, a mean equal to 0 and standardize the inputs to a layer for each mini-batch. This stabilizes the learning process and significantly reduces the number of training epochs needed to create deep networks [29, 30]. Consider an input I_m and an output O_m in a mini batch$m = \{1,2, \ldots, M\}$. This process can be defined in the following steps

-Calculate the mini-batch mean μ with equation (2)

$$\mu = \frac{1}{M}\sum_{m=1}^{M} I_m \quad (2)$$

-Calculate the mini-batch variance σ^2 with equation (3)

$$\sigma^2 = \frac{1}{M}\sum_{m=1}^{M}(I_m - \mu)^2 \quad (3)$$

-Calculate the normalized value of the input x_m with equation (4)

$$x_m = \frac{I_m - \mu}{\sqrt{\sigma^2 - \varepsilon}} \quad (4)$$

- Calculate the output O_m with equation (5)

$$O_m = \alpha x_m + \beta \quad (5)$$

Where ε is a small positive number for avoiding the divisor is 0, α and β are two parameters that can be learned via back propagation.

Layer 1(c): Rectified Linear Unit (Relu)

It is the activation function used in DL, his function will give zero if the input value is less than or equal to zero. Otherwise, it will give a value equal to the input value. The ReLu function has great advantages; it provides a much faster computing rate. Since it is unsaturated, there is no gradient diffusion problem, unlike the sigmoid and tan functions. Although these advantages, it have a big drawback because the derivative of the ReLu function is always zero when the input value is negative [31]. The following equation is used to calculate it.

$$f(x) = max(0, x) \tag{6}$$

Layer 1 (d): Padding

The loss of information that may be present at the image boundaries is a drawback of the convolution layer. Since they are only captured when the filter slips, they will never have a chance to see them. Using zero padding is a very easy and effective technique to tackle the problem. The management of output volume is another advantage of zero padding. This padding technique allows us to keep the network output size from reducing as the depth of the network grows [32]. where the output equation is

$$O = 1 + \frac{N+2P-F}{S} \tag{7}$$

Layer 2: Pooling Layer

It is similar to the convolutional layer. It aims to gradually reduce the dimensions of the representation by reducing the number of parameters and the computational complexity of the model. One of the most popular pooling methods is max-pooling. It divides the image into sub-area rectangles and only returns the maximum value of each sub- area. One of the most common filter sizes used is *2x2*. As can show in Figure1, when the assembly is performed in the upper left *2x2* blocks (blue area), it moves with stride 2 and focuses on the upper right. This means that step 2 is used for pooling [32,33].

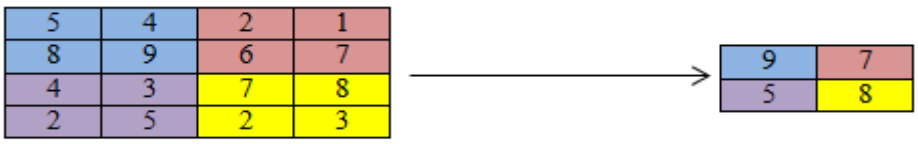


Figure 1 :Max Pooling using *2×2* filter and stride 2.

Layer 3: Drop Out Layer

Overfitting is caused by a number of parameters, particularly in the fully connected layer. A dropout layer is used in many CNN systems to prevent overfitting by removing a part of units with probability p at each training phase. In other words, after adopting dropout, the networks are distinct from one another and perform better than a regular neural network, which improves the model's resistance to overfitting and speeds up training [29].

Layer 4(a): Fully Connected Layer (FC)

It's similar to how neurons in a traditional neural network are organized. In other words every node in a fully-connected layer is directly connected to every node in the previous and subsequent layers. as shown in Figure 2. The main disadvantage of a fully-connected layer is that it has a large number of parameters that need complicated calculation in training phase. Therefore, strive to reduce the number of nodes and connections. The dropout technique can be used to restore the lost nodes and connections. The number of outputs is equal to the number of target classes in this layer [12, 29].

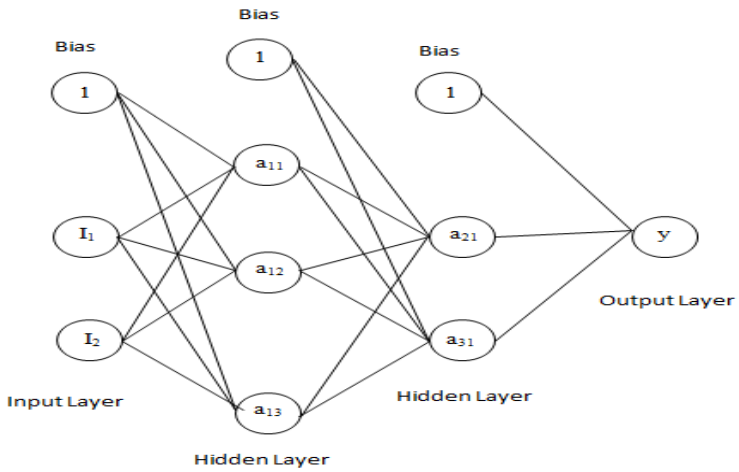


Figure2: The structure of the fully connected layer.

Layer 4 (b): Softmax Activation Function

It is a mathematical function that turns a vector of numbers into a vector of probabilities and is used in the output layer of classification networks. These probabilities are used for a classification process. This softmax function $f(x)$ is defined as follows

$$Softmax :\{ I \epsilon R^N \} \qquad \{P \epsilon R^N ; P_i > 0 , \sum_{j=1}^{N} P_j = 1\} \tag{8}$$

$$P_j = \frac{exp(I_j)}{\sum_{n=1}^{N} exp(I_n)} \qquad for\ j = 1,2, \dots, N \tag{9}$$

Where I_j is the input neuron to the softmax layer, and I is the input vector. P_j is the output that is cognate of I_j, and P is the output vector and N is the total number of classes [33, 34].

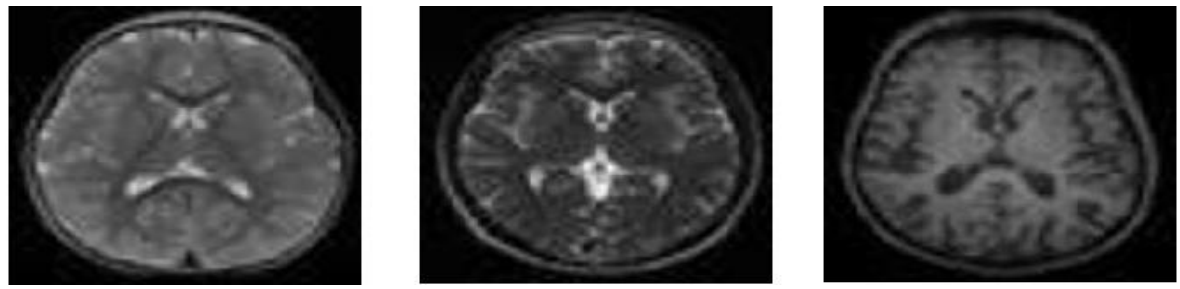
Layer 4 (c) Classification Layer

It is a classifier with the softmax function that is the last layer of the CNN system. This layer calculates the class probability for the input image. During training the network tries to optimize a loss function, the cross-entropy function works with the softmax function and it is considered the better option after applying the softmax function. It is used to measure the difference between a probability distribution $f(x)$ and the desired distribution $d(x)$. The cross-entropy is expressed as the following equation; where N is the total number of classes [35].

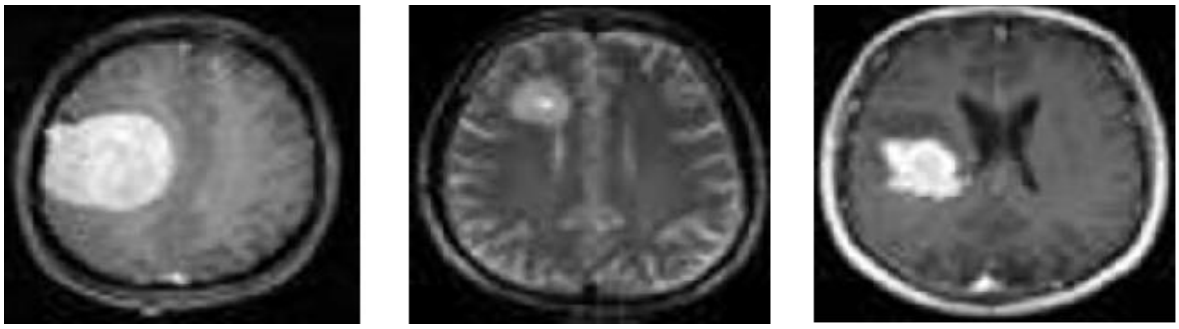
$$C_{\text{i}} = \sum_{n=1}^{N} [d(x^i)]_n \log[f(x^i)]_n \tag{10}$$

3. DATA SET

This research is applied on a famous dataset, called BRATS2015 [36]. The dataset contains 580 MRI patients images. Classify them into two categories, 80 are normal and 500 are abnormal and store each category in a specific folder; as shown in Figure 3. They have different sizes and are resized to 64×64, 128×128, 224×224 and 255×255 then divided these dataset images into 70% for training and 30% for test.



a)



b)

Figure 3: Some examples of the used dataset

a) Normal images b) Abnormal images

4. PROPOSED SYSTEM

This research presents an automatic system based on the CNN to classify MRI brain tumor using number of layers starting from 6 layers to 10 layers by an increment 2 with different sizes of images 64,128,224 and 255 . These layers include convolutional layers, max pooling, fully connected layer , softmax and Relu activation functions as illustrate in table 1. As well as the number of epochs is specified 50 epochs, Learning rates are chosen to be 0.01, 0.1 and 0.5. The system is applied using a linear combination among above three parameters. The whole process of this system is shown in Figure 4. The system is trained and tested on the specified training and test datasets. All experiments were performed on a PC with an NVIDIA GeForce GTX 1070 Ti GPU.

Layer no	Layer Name
1	Input
2	Conv_1
3	Maxpooling1
4	FC
5	Softmax
6	Classification

Layer no	Layer Name
1	Input
2	Conv_1
3	Maxpooling1
4	Conv_2
5	Maxpooling2
6	FC
7	Softmax
8	Classification

Layer no	Layer Name
1	Input
2	Conv_1
3	Maxpooling1
4	Conv_2
5	Maxpooling2
6	Conv_3
7	Maxpooling3
8	FC
9	Softmax
10	Classification

Table 1 : CNN with various Layers

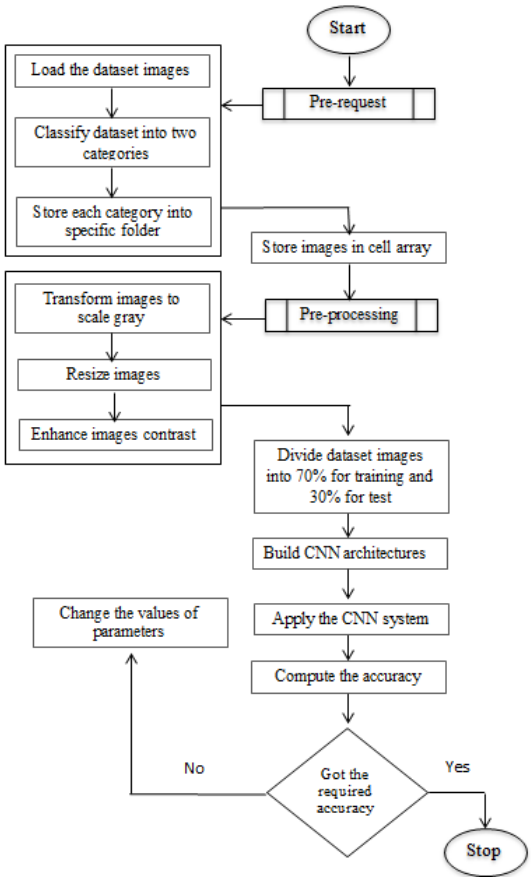


Figure4: Proposed Classification System

4.1 The Best System Steps

-Start

-Load dataset images and label them

-Resize images

- Divide dataset images into 70% for training and 30% for test

-Build CNN architecture

1 -Input Layer with different images size

2- Convolutional Layer (Cov١) having 8 filters with size 3×3, padding=1 and stride 1×1

- Batch Normalization Layer (bn1)
- Relu applied (Relu1)

3 -Max pooling Layer (max pooling١) with size 3×3, padding=1 and stride 2×2

4 -Convolutional Layer (Cov٢) having 16 filters with size 3×3, padding=1 and stride 1×1

- Batch Normalization Layer (bn2)
- Relu applied (Relu2)

5 -Max pooling Layer (max pooling2) with size 3×3, padding=1 and stride 2×2

6- Convolutional Layer (Cov3) that have 32 filters with size 3×3, padding=1and stride 1×1

- Batch Normalization Layer (bn3)
- Relu applied (Relu3)

7- Max pooling Layer (max pooling3) with size 3×3, padding=1 and stride 2×2

8 -Fully connected layer with 2 neurons

9- Softmax Layer to convert a vector of numbers into a vector of probabilities

10- Classification Layer to classify input image according to the probabilities from the previous layer

- Train CNN system using options images sizes 64,128,224 and 255, learning rate 0.01, 0.1and 0.5, minibatchs=128 and number of epochs id 50 by an increment 5

-Test the system and compute performance accuracy

-Stop

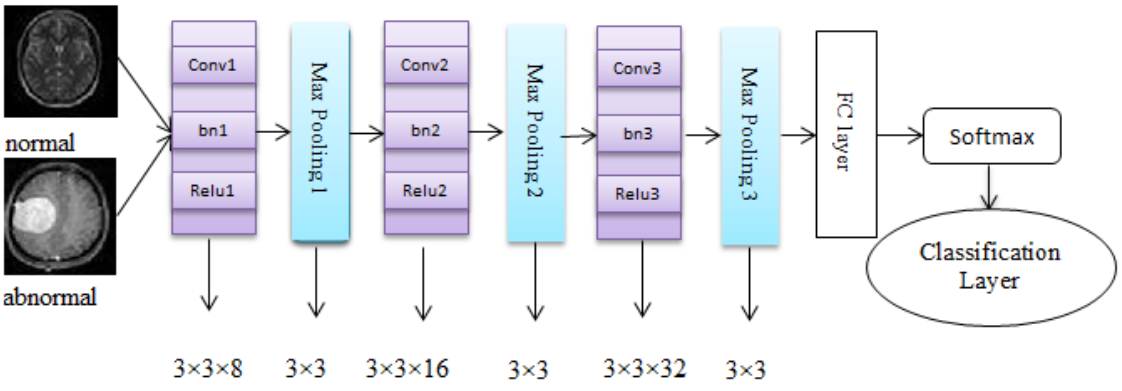


Figure 5: The architecture of the best system having 10 layers

5. EXPERIMENTS RESULTS AND DISCUSSIONS

Various experiments were performed according to three parameters; these parameters are a number of layers, size of images, and learning rates and computed the training, testing accuracies, and time complexity as shown in tables 2, 3, and 4.observed that the time complexity decreases when the number of layers and the size of images decreases and the learning rate increases.

In Table 2, twelve experiments show training, testing accuracies, and time complexity with 6 layers and epochs =50, different learning rates (0.01,0.1,0.5) and different sizes of images 64,128,224 and 255 that achieved average training accuracies from 86.2% to 99.1% ,average testing accuracies from 86.2% to 99.4% and maximum time complexity is 9 min 40 sec as shown in figure 6 .In Table3, twelve experiments show training, testing accuracies, and time complexity with 8 layers and epochs =50, different learning rates (0..01,0.1,0.5) and different sizes of images 64,128,224 and 255 that achieved average training accuracies from 86.2% to 99.1% ,average testing accuracies from 86.2% to 99.4% and maximum time complexity is 14 min 52 sec as shown in figure 7. In Table 4, twelve experiments show training, testing accuracies, and time complexity with 10 layers and epochs =50, different learning rates (0.01,0.1,0.5) and different sizes of images 64,128,224 and 255 that achieved average training accuracies from 86.2% to 99.1% ,average testing accuracies from 86.2% to 99.6% and maximum time complexity is 17 min 46 sec as shown in figure 8.

Learning Rate	Training accuracy				Time Complexity				Testing accuracy			
	64×64	128×128	224×224	255×255	64×64	128×128	224×224	225×225	64×64	128×128	224×224	225×225
0.01	99.1%	99.1%	98.3%	94.8%	1min14sec	3min1sec	8min15sec	9min40sec	99.4%	99.4%	99.1%	99.1%
0.1	98.3%	86.2%	86.2%	86.2%	1min12sec	2min54sec	7min28sec	9min38sec	99.1%	86.2%	86.2%	86.2%
0.5	86.2%	86.2%	86.2%	86.2%	1min11sec	2min53sec	7min23sec	9min18sec	86.2%	86.2%	86.2%	86.2%

Table2: Training , Testing Accuracies and time complexity with layers=6 & different learning rates, images size and epochs=50

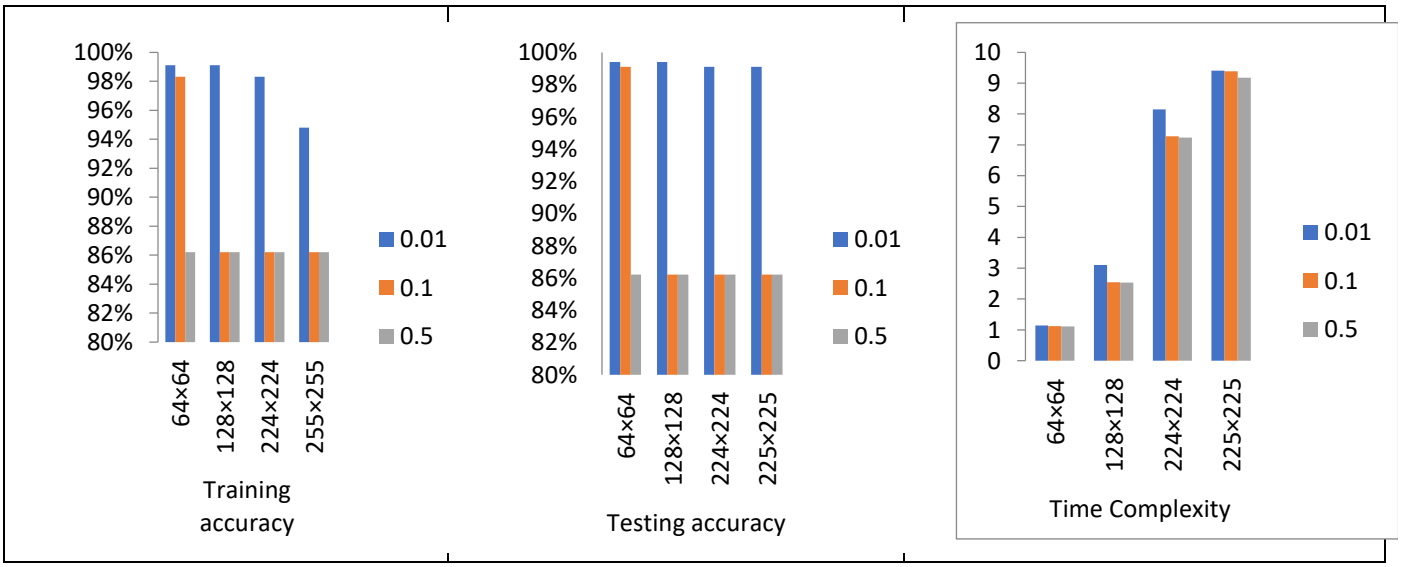


Figure 6:The Training , Testing accuracies and time complexity ratios according to 6 Layers

Learning Rate	Training accuracy				Time Complexity				Testing accuracy			
	64×64	128×128	224×224	255×255	64×64	128×128	224×224	225×225	64×64	128×128	224×224	225×225
0.01	98.3%	97.4%	99.1%	98.3%	1min38sec	4min16sec	11min20sec	14min52sec	99.4%	99.4%	99.4%	99.1%
0.1	99.1%	89.7%	97.4%	92.2%	1min37sec	4min12sec	11min18sec	14min44sec	99.4%	90.5%	99.4%	98.1%
0.5	86.2%	86.2%	86.2%	86.2%	1min35sec	4min10sec	11min7sec	14min40sec	86.2%	86.2%	86.2%	86.2%

Table3: Training , Testing Accuracies and time complexity with layers=8& different learning rates, images size and epochs=50

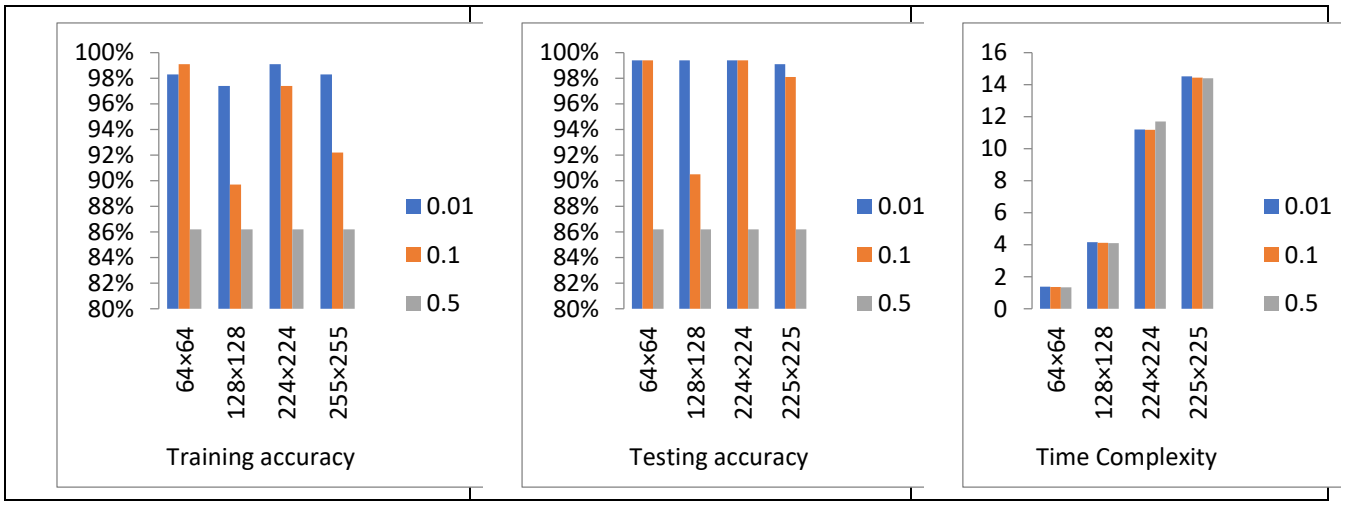


Figure 7:The Training , Testing accuracies and time complexity ratios according to ٨ Layers

Learning Rate	Training accuracy				Time Complexity				Testing accuracy			
	64×64	128×128	224×224	255×255	64×64	128×128	224×224	225×225	64×64	128×128	224×224	225×225
0.01	96.6%	99.1%	99.3%	97.4%	1min48sec	5min13sec	14min20sec	17min46sec	99.6%	99.6%	99.4%	99.4%
0.1	98.3%	91.4%	99.1%	92.2%	1min21sec	5min10sec	13min44sec	16min24sec	99.4%	98.2%	99.4%	94%
0.5	86.2%	86.2%	86.2%	86.2%	1min20sec	5min10sec	13min35sec	16min20sec	86.2%	86.2%	86.2%	86.2%

Table 4: Training , Testing Accuracies and time complexity with layers=10& different learning rates, images size and epochs=50

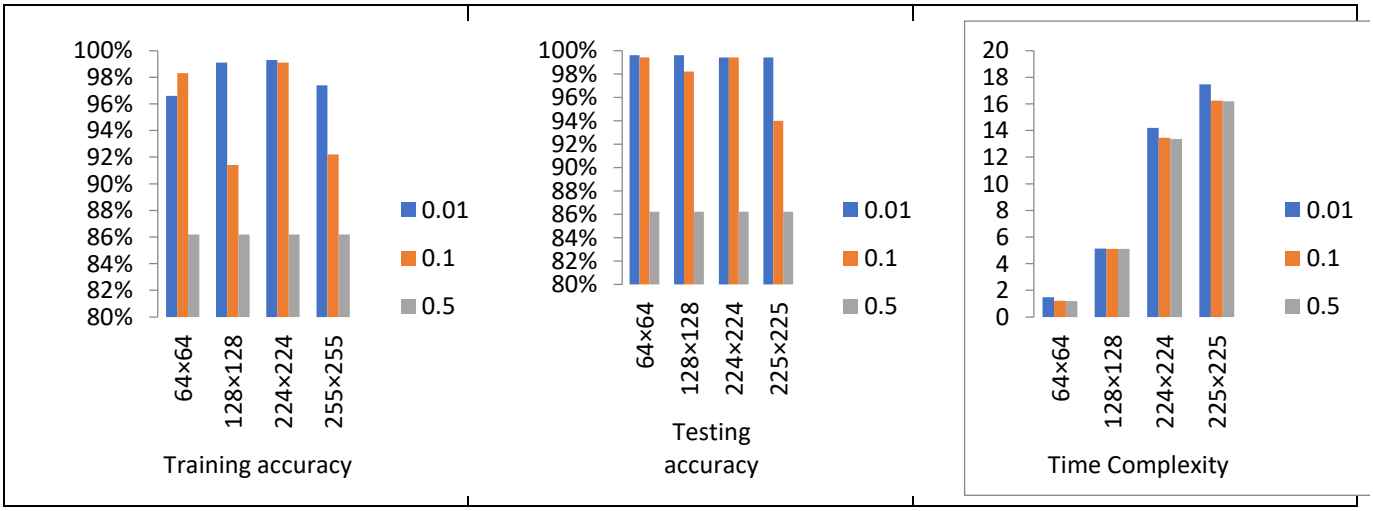


Figure 8:The Training , Testing accuracies and time complexity ratios according to 10 Layers

5-1 Parameters measures:

To evaluate the performance of the proposed model architecture, the well-known performance measures for the evaluation are used, in terms of, the training accuracy, testing accuracy and time complexity.

$$\text{Training accuracy } \frac{TP+TN}{TP+FP+TN+FN} \quad (11)$$

Where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. And Accuracy determines the ability to correctly differentiate between types of brain tumors. To estimate test accuracy, we calculate the ratio of true positivity and true negativity in all evaluated cases[23]. Time complexity is the length of time an algorithm takes to run as a function of the input length. It calculates the amount of time needed to run each algorithmic statement of code. It won't look at an algorithm's total time spent running. Instead, it will provide data on how an algorithm's execution time varies (increases or decreases) depending on the number of operations it contains. The length of the input alone determines how long it takes.CNN time complexity is Linear time because the running time increases linearly with the length of the input. When the function involves checking all the values in input data, with this order O(n)[37].

6. COMPARATIVE STUDIES

In this research make comparative studies with the same dataset used. Khan et al.[9] used linear contrast stretching, discrete cosine transform and VGG16 , VGG19 together and achieved accuracy 97.8%. Alkassar et al. [10] used used VGG16 including the encoder and decoder networks that achieved accuracy 97.7%. Abdullah et al. [11] used two models LBTS-Net16 and LBTS-Net19 that achieved accuracy 96.7% and 95.9% respectively. Amin et al .[12] used discrete wavelet transform fusion and a partial diffusion filter passed to CNN model and achieved 96% accurate. Yang et al .[13] used The CNN with large and small kernels called (SK-TPCNN) achieved accuracy 89%. Rani et al.[15] used hybrid deep auto encoder with a Bayesian fuzzy cluster and achieved 98.5% accuracy. Abd El Kader et al.[16] used a deep wavelet auto encoder model that achieved accuracy 99.3%. Ding et al.[17] used Deep Residual Dilate Network with Middle Supervision (RDM-Net) and achieved 86% accuracy. Hariharan et al. [18] used marker controlled watershed mining achieved 92% accurate. Where proposed system used Convolutional Neural Network with multi-layers and parameters achieved height accuracy 99.6 % at 10 layers .

Author	model	Accuracy
Khan et al. [9]	linear contrast stretching , discrete cosine transform and VGG16 , VGG19 together	97.8%
Alkassar et al. [10]	VGG16 including the encoder and decoder networks	97.7%
Abdullah et al .[11]	LBTS-Net16 and LBTS-Net19	96.7 %
Amin et al .[12]	discrete wavelet transform fusion and a partial diffusion filter passed to CNN model	96%
Yang et al .[13]	The CNN with large and small kernels called (SK-TPCNN)	89%
Raja et al.[15]	hybrid deep auto encoder with a Bayesian fuzzy cluster	98.5%
Abd El Kader et al.[16]	a deep wavelet autoencoder model	99.3%
Ding el al. [17]	deep residual dilate network	86%
Hariharan et al. [18]	marker controlled watershed mining	92%
Proposed system	Convolutional Neural Network	99.6%

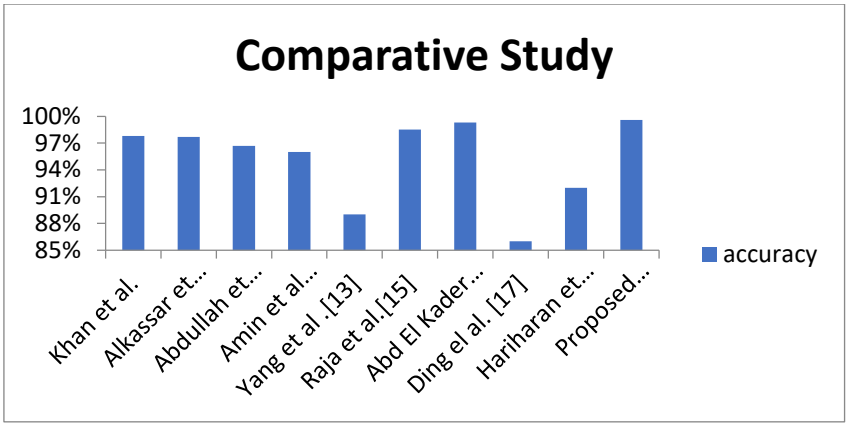


Figure 9: Comparative Study

7. CONCLUSIONS AND FUTURE WORK

In this research Deep Learning techniques are very necessary and essential in brain tumor classification processes. The menu script presents an effective automatic CNN system with different parameters. These parameters are number of layers, learning rates and size of images using the BRATS 2015 dataset. The proposed system was trained automatically and applied several times with various layers from 6 to 10 by increment 2, epochs = 50 and learning rates values 0.01, 0.1 and 0.5 with different size of images 64,128,224 and 255. Observed that the time complexity decreases when the number of layers and the size of images decrease and the learning rate increases, and the best system at 10 layers which achieve higher accuracy 99.6%.

The proposed system will be improved to classify several medical datasets and update this system to determine the brain tumor and specify type and degree.

8. REFERENCES

[1] A. H. Khan *et al.*, "Intelligent Model for Brain Tumor Identification Using Deep Learning," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–10, Jan. 2022, doi: 10.1155/2022/8104054.

[2] Venkatesh and M. J. Leo, "MRI Brain Image Segmentation and Detection Using K-NN Classification," *J. Phys.: Conf. Ser.*, vol. 1362, no. 1, p. 012073, Nov. 2019, doi: 10.1088/1742-6596/1362/1/012073.

[3] C. Arizmendi, A. Vellido, and E. Romero, "Classification of human brain tumours from MRS data using Discrete Wavelet Transform and Bayesian Neural Networks," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5223–5232, Apr. 2012, doi: 10.1016/j.eswa.2011.11.017.

[4] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, "Brain Tumor Classification Using Convolutional Neural Network," in *World Congress on Medical Physics and Biomedical Engineering 2018*, vol. 68/1, L. Lhotska, L. Sukupova, I. Lacković, and G. S. Ibbott,

Eds. Singapore: Springer Nature Singapore, 2019, pp. 183–189. doi: 10.1007/978-981-10-9035-6_33.

[5] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," *Journal of Computational Science*, vol. 30, pp. 174–182, Jan. 2019, doi: 10.1016/j.jocs.2018.12.003.

[6] S. Ahuja, B. K. Panigrahi, and T. K. Gandhi, "Enhanced performance of Dark-Nets for brain tumor classification and segmentation using colormap-based superpixel techniques," *Machine Learning with Applications*, vol. 7, p. 100212, Mar. 2022, doi: 10.1016/j.mlwa.2021.100212.

[7] N. Zulpe and V. Pawa , "GLCM textural features for brain tumor classification," in International Journal of Computer Science Issues (IJCSI), vol 9, no.3 ,pp. 354,2012.

[8] B. Ahmad, J. Sun, Q. You, V. Palade, and Z. Mao, "Brain Tumor Classification Using a Combination of Variational Autoencoders and Generative Adversarial Networks," *Biomedicines*, vol. 10, no. 2, p. 223, Jan. 2022, doi: 10.3390/biomedicines10020223.

[9] M. A. Khan *et al.*, "Multimodal Brain Tumor Classification Using Deep Learning and Robust Feature Selection: A Machine Learning Application for Radiologists," *Diagnostics*, vol. 10, no. 8, p. 565, Aug. 2020, doi: 10.3390/diagnostics10080565.

[10] S. Alkassar, M. A. M. Abdullah, and B. A. Jebur, "Automatic Brain Tumour Segmentation using fully Convolution Network and Transfer Learning," in *2019 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering (ICECCPCE)*, Mosul, Iraq, Feb. 2019, pp. 188–192. doi: 10.1109/ICECCPCE46549.2019.203771.

[11] M. A. M. Abdullah, S. Alkassar, B. Jebur, and J. Chambers, "LBTS-Net: A fast and accurate CNN model for brain tumour segmentation," *Healthc. technol. lett.*, vol. 8, no. 2, pp. 31–36, Apr. 2021, doi: 10.1049/htl2.12005.

[12] J. Amin, M. Sharif, N. Gul, M. Yasmin, and S. A. Shad, "Brain tumor classification based on DWT fusion of MRI sequences using convolutional neural network," *Pattern Recognition Letters*, vol. 129, pp. 115–122, Jan. 2020, doi: 10.1016/j.patrec.2019.11.016.

[13] T. Yang, J. Song, and L. Li, "A deep learning model integrating SK-TPCNN and random forests for brain tumor segmentation in MRI," *Biocybernetics and Biomedical Engineering*, vol. 39, no. 3, pp. 613–623, Jul. 2019, doi: 10.1016/j.bbe.2019.06.003.

[14] A. Casamitjana,S. Puch,A. Aduriz,E. Sayrol and V. Vilaplana , "3D Convolutional Networks for Brain Tumor Segmentation, " in Proceedings of the MICCAI challenge on multimodal brain tumor image segmentation (BRATS) pp. 65-68 ,2016.

[15] P. M. Siva Raja and A. V. rani, "Brain tumor classification using a hybrid deep autoencoder with Bayesian fuzzy clustering-based segmentation approach," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 440–453, Jan. 2020, doi: 10.1016/j.bbe.2020.01.006.

[16] I. Abd El Kader *et al.*, "Brain Tumor Detection and Classification on MR Images by a Deep Wavelet Auto-Encoder Model," *Diagnostics*, vol. 11, no. 9, p. 1589, Aug. 2021, doi: 10.3390/diagnostics11091589.

[17] Y. Ding, C. Li, Q. Yang, Z. Qin, and Z. Qin, "How to Improve the Deep Residual Network to Segment Multi-Modal Brain Tumor Images," *IEEE Access*, vol. 7, pp. 152821–152831, 2019, doi: 10.1109/ACCESS.2019.2948120.

[18] D. Hariharan, S. Hemachandar, N. Sri Madhava Raja, H. Lin, and K. Sundaravadivu, "Inspection of 2D Brain MRI Slice Using Watershed Algorithm," in *Intelligent Data Engineering and Analytics*, vol. 1177, S. C. Satapathy, Y.-D. Zhang, V. Bhateja, and R. Majhi, Eds. Singapore: Springer Singapore, 2021, pp. 721–730. doi: 10.1007/978-981-15-5679-1_70.

[19] M. Yaqub *et al.*, "State-of-the-Art CNN Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images," *Brain Sciences*, vol. 10, no. 7, p. 427, Jul. 2020, doi: 10.3390/brainsci10070427.

[20] H. Cho and H. Park, "Classification of low-grade and high-grade glioma using multi-modal image radiomics features," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Seogwipo, Jul. 2017, pp. 3081–3084. doi: 10.1109/EMBC.2017.8037508.

[21] G. Latif, D. N. F. A. Iskandar, J. M. Alghazo, and N. Mohammad, "Enhanced MR Image Classification Using Hybrid Statistical and Wavelets Features," *IEEE Access*, vol. 7, pp. 9634–9644, 2019, doi: 10.1109/ACCESS.2018.2888488.

[22] A. Selvapandian and K. Manivannan, "Fusion based Glioma brain tumor detection and segmentation using ANFIS classification," *Computer Methods and Programs in Biomedicine*, vol. 166, pp. 33–38, Nov. 2018, doi: 10.1016/j.cmpb.2018.09.006.

[23] O. M. Elzeki, M. Shams, S. Sarhan, M. Abd Elfattah, and A. E. Hassanien, "COVID-19: a new deep learning computer-aided model for classification," *PeerJ Computer Science*, vol. 7, p. e358, Feb. 2021, doi: 10.7717/peerj-cs.358.

[24] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[25] W. Widhiarso, Y. Yohannes, and C. Prakarsah, "Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network," *Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, p. 179, Oct. 2018, doi: 10.22146/ijeis.34713.

[26] M. Pawar,K. Desai,S. Gupta,T. Chavan and K. Mhamunkar , "Object Detection and Currency Recognition Using CNN," in Cikitusi Journal For Multidisciplinary, Vol 6,no.4, pp 357-362, April 2019.

[27] J. Seetha and S. S. Raja, "Brain Tumor Classification Using Convolutional Neural Networks," in *Biomed. Pharmacol. J.*, vol. 11, no. 3, pp. 1457–1461, Sep. 2018, doi: 10.13005/bpj/1511.

[28] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Aug. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[29] S.-H. Wang, K. Muhammad, J. Hong, A. K. Sangaiah, and Y.-D. Zhang, "Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization," *Neural Comput & Applic*, vol. 32, no. 3, pp. 665–680, Feb. 2020, doi: 10.1007/s00521-018-3924-0.

[30] N. Bjorck ,C. P. Gomes ,B. Selman and K. Q. Weinberger , "Understanding batch normalization," in neural information processing systems, vol.31,2018..

[31] B. Hanin, "Universal Function Approximation by Deep Neural Nets with Bounded Width and ReLU Activations," *Mathematics*, vol. 7, no. 10, p. 992, Oct. 2019, doi: 10.3390/math7100992.

[32] W. Widhiarso, Y. Yohannes, and C. Prakarsah, "Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network," *Indonesian J. Electron. Instrum. Syst.*, vol. 8, no. 2, p. 179, Oct. 2018, doi: 10.22146/ijeis.34713.

[33] M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, "A High-Speed and Low-Complexity Architecture for Softmax Function in Deep Learning," in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Chengdu, Oct. 2018, pp. 223–226. doi: 10.1109/APCCAS.2018.8605654.

[34] A. Pashaei, H. Sajedi, and N. Jazayeri, "Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Oct. 2018, pp. 314–319. doi: 10.1109/ICCKE.2018.8566571.

[35] A.de la Calle,A. Aller,J. Tovar and E. J. Almazan "Geometric interpretation of a CNN's last layer ," in CVPR Workshops, 2019,pp-79-82.

[36]https://www.kaggle.com/xxc025/brats2015.

[37] https://www.mygreatlearning.com/blog/why-is-time-complexity-essential/