

Brain Tumor Classification & Segmentation by Using Advanced DNN, CNN & ResNet-50 Neural Networks

Imran Javaid**

Hebei University of Technology
8 Dingzigu 1st Rd, Hongqiao Qu,
China, 300131.

Shuai Zhang*

Hebei University of Technology
8 Dingzigu 1st Rd, Hongqiao Qu,
China, 300131.

Abd El Kader Isselmou

Hebei University of
Technology
8 Dingzigu 1st Rd, Hongqiao
Qu, China, 3003131.

Souha Kamhi

Hebei University of Technology
8 Dingzigu 1st Rd, Hongqiao Qu,
China, 3003131.

Isah Salim Ahmad

Hebei University of Technology
8 Dingzigu 1st Rd, Hongqiao Qu,
China, 3003131.

Ummay Kulsum

Hebei University of
Technology
8 Dingzigu 1st Rd, Hongqiao
Qu, China, 3003131.

Received: September 27 2020. Revised: December 16, 2020. Accepted: December 21, 2020. Published: December 22, 2020.

Abstract— In the medical domain, brain image classification is an extremely challenging field. Medical images play a vital role in making the doctor's precise diagnosis and in the surgery process. Adopting intelligent algorithms makes it feasible to detect the lesions of medical images quickly, and it is especially necessary to extract features from medical images. Several studies have integrated multiple algorithms toward medical images domain. Concerning feature extraction from the medical image, a vast amount of data is analyzed to achieve processing results, helping physicians deliver more precise case diagnoses. Image processing mechanism becomes extensive usage in medical science to advance the early detection and treatment aspects. In this aspect, this paper takes tumor, and healthy images as the research object and primarily performs image processing and data augmentation process to feed the dataset to the neural networks. Deep neural networks (DNN), to date, have shown outstanding achievement in classification and segmentation tasks. Carrying this concept into consideration, in this study, we adopted a pre-trained model Resnet_50 for image analysis. The paper proposed three diverse neural networks, particularly DNN, CNN, and ResNet-50. Finally, the splitting dataset is individually assigned to each simplified neural network. Once the

image is classified as a tumor accurately, the OTSU segmentation is employed to extract the tumor alone. It can be examined from the experimental outcomes that the ResNet-50 algorithm shows high accuracy 0.996, precision 1.00 with best F1 score 1.0, and minimum test losses of 0.0269 in terms of Brain tumor classification. Extensive experiments prove our offered tumor detection segmentation efficiency and accuracy. To this end, our approach is comprehensive sufficient and only requires minimum pre-and post-processing, which allows its adoption in various medical image classification & segmentation tasks.

Keywords— DNN, CNN, ResNet-50, Image-Segmentation, Brain Tumor Classification.

1. INTRODUCTION

Medical imaging is the primary carrier and crucial vector of a current medical alternative terminology classification. It continually performs a frequently important part in the medical domain amidst objective description and instinctive expression [1]. Originally, from the 1970s to 1990s, a study of the medical image was performed by sequential utilization of low-level pixel processing (line detector filter, edge and growth region) [2]. Various mathematical modelling (ellipses, circles and fitting lines) to create compound rule-based systems that solved specific tasks. Those special systems were called GOFAI (good old-fashioned artificial intelligence) alike rule-based image processing [3]. The primary goal of the classification of brain tumors is to very precisely analyses of MRI images to identify with the type of tumor a patient is affected [4]. Determining whether a tumor confers to the image is benign or malignant is also a hot-topic not ended by the medical community [5]. Glioma is the most considerable mortality risk and incidence rate. Such neoplasms can be classified according to their violent and infiltrative features as low-grade gliomas (LGGs) and high-grade gliomas (HGGs). The brain tumor surgery depended on the size of the tumor; its types and its stage of development. However, in the healthcare sector, physicians use numerous medical imaging methods to diagnose the occupancy of a brain tumor [7]. Multiple imaging methods own to examine brain tumors, such as computed tomography (CT) [8], magnetic resonance imaging (MRI) [9], computed tomography with single-photon emission (SPECT) and positron emission tomography (PET) [10]. Medical imaging may be used as a tool to generate a medical image for clinical study, analysis. MRI Adopting a magnetic system to capture medical images. It can deliver reproducible quantitative and non-invasive determinations of tissue, including anatomical, architectural and functional information's [11]. Among them, MRI fitted the primary imaging method for tumor determination and practice; besides its dominance of proper soft-tissue contrast, non-invasive, multi-parameter, imaging arbitrary etc. Moreover, various modalities, e.g. T1-weighted with contrast improvement(T1c), T1-weighted (T1), Inversion Recovery (FLAIR), Fluid Attenuated and T2-weighted (T2) also obtained by MRI [12]. Several MRI modalities concentrate on various specified information of images and specify the features of brain tumors from diverse aspects. The precise Brain segmentation tumors are a prominent concern for scientific, surgical diagnosis and preparing for surgery. In particular, this is critical separating tumor tissues, for example, edema, necrosis, enhancement and non-enhancing nuclei of healthy brain tissues the white matter (WM), coronary spinal fluid (CSF) and the grey matter (GM)[13]. Despite to segment them correctly, it is a notably challenging task mainly because of the following causes. Primary, gliomas can differ significantly in form, position, appearance, and size from case to case. Secondary, gliomas

typically penetrate surrounding tissues, thereby blurring the boundary [14]. Third, the deformity of the image and the noise-induced by various factors such as imaging devices or acquisition protocols additional enhances the effect [15]. The image segmentation is a division of an image into segments described groups or sub-sets according to an individual or more features or characteristics, and intensifying areas of concern by separating them from the background and other regions [16]. To overcome this problem, the earliest successful practice of image features extraction implied in 1972 when Sutton and Hall identified healthy and abnormal lungs [17]. About the completion of the 1990s, supervised methods were frequently growing popular in medical image analysis [18]. The algorithm is based on training data which is used to design a system. Examples are included, atlas approaches (where the atlases that are suited to current data from the training data), active shape models (for segmentation), and the theory of extracting features and utilizing statistical classifiers (for computer-aided detection and analysis) [19]-[20]. In the past and recent years, by the advancement of computer execution and the continuous improvement of machine learning algorithms, many automatic and semi-automatic techniques for tumor segmentation have been introduced [21]. Towards the medical image processing, both conventional machine learning [22]-[23]-[24]-[25] and deep learning techniques [26]-[27]-[28]-[29] have been broadly studied and practiced by researchers. However, in most circumstances, deep learning techniques produce abler in computer vision tasks, but still, deep learning algorithms own substantial success in recognition of images [30]. Currently, various conventional models are fully trained end-to-end in a supervised manner, effects simplifying the training method. Neural networks (NN) classify by learning from data and never apply rule sets [31]. The neural network can generalize utilizing previous data and discover from past experience. They hold benefits such as learning through themselves, fault tolerance and the search for the optimum [32]. They work adequately on non-linear, complicated multivariate and noisy domains, such as segmentation of brain tissue. As a long time, deep neural networks (DNN) have long been considered challenging to train efficiently [33]. They barely grew reputation in 2006 while it did show that training layer by layer DNNs(pre-training) in an unsupervised manner, supported by supervised fine-tuning of the network stacked, could lead to high-grade results [34]. Convolutional neural networks (CNNs) are the most popular kind of models for image analysis. CNN hold multiple layers which transform their input by utilizing convolution filters of a small size [35]. Since the late seventies, research on CNNs has been performed and already practiced to medical images study in 1995 [36]. Despite particular primary successes, the performance of CNN's did not attain momentum until modern new systems were developed to train deep network work efficiently, and advancement was created in core computing systems [37]. In addition to deeper networks, further complex building blocks

were offered to advance the performance of the training method and reduce the number of parameters once again [38]. Szegedy et al (2014) established a 22-layer Google Net network, also known as Inception, which employed such named inception blocks [39]. The architecture of ResNet won the contest of ImageNet in 2015 and comprised so-called ResNet blocks [40]. Instead of reading a function, the residual block barely learns the residual and is consequently pre-conditioned in each layer to learn mappings that are approaching to the identity function.

It is repeatedly challenging to assess whether this is due to a superior or simple architecture because the model is a default selection option in modern software packages [41]. In deep learning or machine learning the process of classification is known as a supervised learning challenge that indicates a link among class labels and features (characteristics of the dataset) [42]. These derived features are the basis for the classification concerning high-dimensional data [43]. However, in the medical image analysis and segmentation, the domain had the models with low accuracy efficiency of diagnosis, tumor types and area. Therefore, decision-making is it of great impact on a patient's life; thus, the specialist began utilizing data mining methods to recognize the appearance of these tumors. So, specialists consider the brain tumor detection system as a secondary option since it corresponded to their own judgment in deciding the right determination and therapy of brain tumors. Through adopting this method, much deeper models could be trained efficiently. Subsequently, ResNet, Alex-Net or other simplistic models such as VGG remain still famous for medical data [44]-[45]. Though current studies all apply a Google Net version called Inception v3 [46]. The core objective of this paper is to develop an intelligent model that can analysis and segments the MR brain images further efficiently than existing models.

II. PROPOSED METHODOLOGY

The experiment was conducted carried out on Spyder, in an environment of Keras 2.2.4 with Tensor flow 2.0 and Theano 1.0 backend. The System in which the experiment was carried out runs on Windows 10 and had a RAM of 8GB and no Graphics Processing Unit's (GPU's) were utilized in this experiment Fig 1 shows the flow of the algorithm.

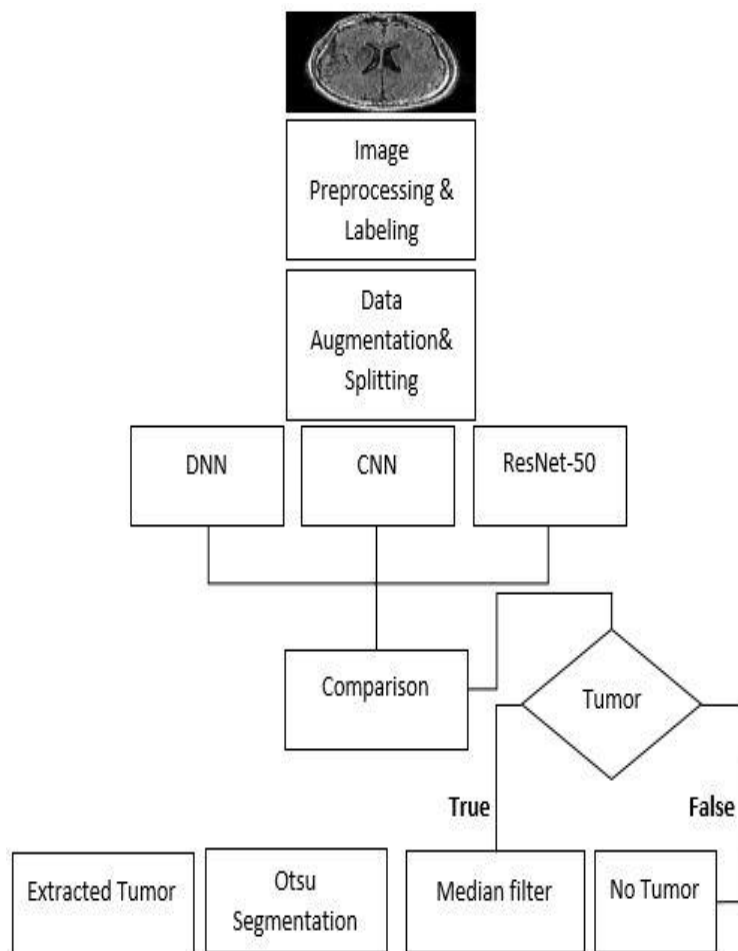


Fig. 1 Flow chart of Brain Tumor classification and Segmentation

III. DATA SET

In our study, we utilized MR brain images of benign 1211, malignant 1212 tumors case as well as the healthy brains of 1739 patients. There are total 4162 T1 weighted, T2 weighted and Flair MR images inside a particular dataset. The entire images are 256 x 256-pixel 8-bit RGB (color images) JPEG format. This open-access dataset is provided by one of the most patronized organization TCIA (The Cancer Imaging Archive) [47]-[48] and some of the other data obtained from General Electric MR machine 1.5 T. In the proposed study, three classes of brain images are picked for classification. The dataset holds a diverse number of image samples, each belongs to a particular class, this leads to emphasize the imbalanced of data set, in each class is conducted in this paper. This dataset is randomly split into two sets as training and validation data. Mixed classes of Brain images trained the proposed models for detection as shown in Fig 2. We further trained the models such that they can learn about characteristics as well as to be able to identify its status.

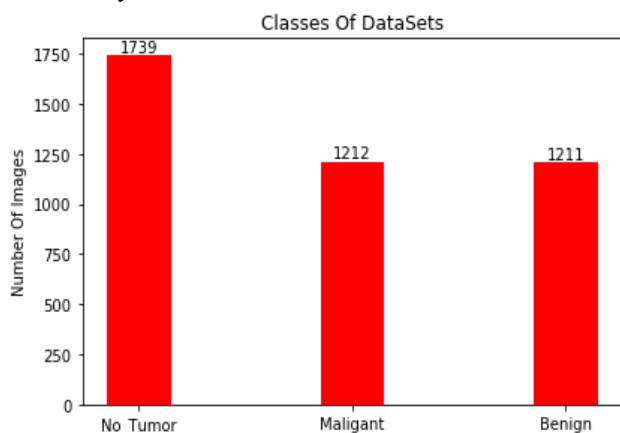


Fig. 2 Shows the Number of images and Category

IV. IMAGE PREPROCESSING & LABELING

This stage involves linear smoothing named Gaussian filtering to intensify the MR brain images for qualify them toward the next stage which is Image preprocessing it helps enhances the necessary image data for image labeling tasks. In preprocessing methods, we choose to use geometric transformations such as image rotation, image scaling, and image translation. While in our preprocessing, we have decreased the resolution concerning all the images into 128*128 pixels. It wants to ensure by that the whole images are supposed to have identical resolution, and after the normalization, we transform RGB to the grayscale images. After the normalized, then we labeling the data in a binary arrangement.

$$S(x) = \ell^{-\frac{x^2}{2\rho^2}} \quad (1)$$

Where the Gaussian range parameter (ρ defines the width of the Gaussian. Toward image processing, the two-dimensional zero-mean discrete Gaussian function is utilized as a smoothing filter

$$S[i, j] = \ell^{-\frac{(i^2+j^2)}{2\rho^2}} \quad (2)$$

The Fourier transform of a Gaussian is calculated by

$$D\{S(x)\} = \int_{-\infty}^{\infty} S(x) \ell^{-j\omega x} dx \quad (3)$$

$$D\{S(x)\} = \int_{-\infty}^{\infty} \ell^{-\frac{x^2}{2\rho^2}} \ell^{-j\omega x} dx \quad (4)$$

$$D\{S(x)\} = \int_{-\infty}^{\infty} \ell^{-\frac{x^2}{2\rho^2}} (\cos \omega x + \sin \omega x) dx \quad (5)$$

$$D\{S(x)\} = \int_{-\infty}^{\infty} \ell^{-\frac{x^2}{2\rho^2}} \cos \omega x dx + j \int_{-\infty}^{\infty} \ell^{-\frac{x^2}{2\rho^2}} \sin \omega x dx \quad (6)$$

The Gaussian is a symmetric function. Furthermore, the sine function is antisymmetric; hence the integrand in the second integral signifies antisymmetric. Since the integral must be zero, moreover the Fourier transform simplifies to:

$$D\{S(x)\} = \int_{-\infty}^{\infty} \ell^{-\frac{x^2}{2\rho^2}} \cos \omega x dx \quad (7)$$

$$D\{S(x)\} = \sqrt{2\pi\rho} \ell^{-\frac{\omega^2}{2\kappa^2}} \quad (8)$$

$$\kappa^2 = \frac{1}{\rho^2} \quad (9)$$

This spatial frequency parameter is w . The range of the Gaussian in the particular frequency domain is controlled through κ , which denotes the reciprocal of the spread parameter ρ in the spatial domain. These features designate that the Gaussian smoothing filters are efficient low-pass filters from the prospect of both the spatial and frequency domains, remain efficient to execute, and can be practiced effectively in image smoothing. These characteristics associates with the noise suppressing capability of a Gaussian filter.

V. DATA AUGMENTATION

Augmentation of image data is a method that can be adopted to artificially expand the capacity of a training dataset by generating updated image in the data set. This method is implemented to advance the number of images data in the Learning dataset. It assists in avoiding overfitting difficulties while training the model [49]. Training deep learning models of neural networks on more exceeding data can produce efficient skillful models, and the augmentation methods can generate image variations that can boost the strength of the fit models to induce anything they have learned into new images [50]. Image data augmentation is conceivably the standard, well-known kind of data augmentation, which includes creating transformed images versions in the training dataset that relate to the corresponding class as the primary image. Transforms hold a spectrum of image processing operations such as flips, zooms, shifts and many more. In Fig 3. we describe the data is splitted in to training as well as validation.

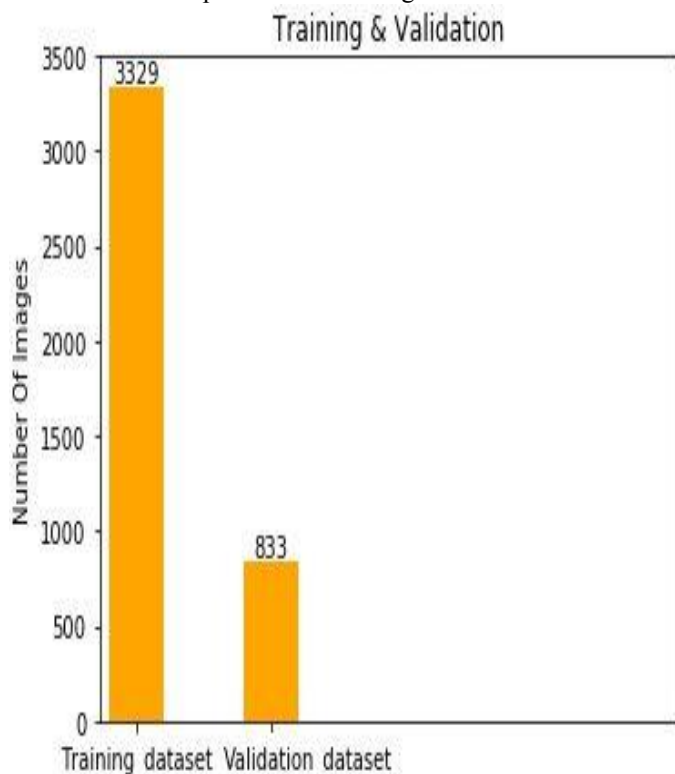


Fig. 3 Shows the splitting of data in to Training and validation dataset.

However, once the data augmentation is completed, the dataset is splitting into two sets, training and validation data sets.

VI. CONVOLUTION NEURAL NETWORK

CNN is an enduring robust method in the field of image classification. The primary architecture of CNN covers input layer, hidden layer, and output layer. The utilization of convolution layers [51] consists of convolving an image or signal by the kernel to get feature maps. So, a unit in a feature map is associated with the previous layer with the weights of the kernels. The kernel weights are adjusted during the backpropagation training process, to intensify the input characteristics. In CNN, the quality of output can be achieved from the forward propagation, and through backpropagation, the bias and weight can be balanced. Since the kernels are shared among all layers of the same feature maps, convolutional layers own some weights to train than dense FC layers, addressing CNN more accessible to train, including less prone to overfitting. Furthermore, since the similar kernel is convolved across all the image, the likewise feature is identified individually of the location translation invariance. By employing kernels, neighborhood information is taken into account, which contributes with valuable contextual information. Normally, non-linear activation function is employed to a particular output of the individual neural unit. The subsequent concepts are essential in the context of CNN.

VII. CONVOLUTIONAL LAYER

The central task of the convolution layer is to identify local conjunctions of features from the former layer and map their presence to a feature map. The image is divided into perceptron's as a consequence of convolution in neuronal networks, producing local receptive fields and eventually reducing the perceptron's in feature maps of size $m_2 \times m_3$. However, this stores the details where these features appear in the image furthermore how expertly corresponds over the filter. Inside the individual layer, they own a bank of m_1 -filters. The estimate of how several filters are applied in the single-stage is equal to the volume depth concerning output feature maps. The individual filter recognizes a distinct feature at each location on the input. Hence the output y_p^l of layer l depend on m_1^l feature maps of size $m_2^l \times m_3^l$. The i^{th} feature map, denoted y_p^l , is calculated as follow.

$$y_p^l = C_p^l + \sum_{n=1}^{m_1^{(l-1)}} K_{p,n}^l * y_n^{(l-1)} \quad (10)$$

Where C_p^l is a bias matrix and $K_{p,n}^l$ is the size of filter $2g_1^l + 1 \times 2g_2^l + 1$ joining the j^{th} map feature in layer $(l-1)$ with i^{th} map feature in layer. The outcome of organizing these convolution layers in combination with the subsequent layers is that the image information is organized similarly in vision.

VIII. ACTIVATION FUNCTION ReLU

The rectified linear units (ReLU) hold a particular implementation that consolidates rectification layers and non-linearity toward convolutional neural networks. A graphical representation of ReLU is defined in fig.4. A ReLU (i.e. thresholding at zero) implies a piecewise linear function represented as:

$$Z_i^l = \max(0, Z_i^{(l-1)}) \quad (11)$$

However, ReLU arises with three vital advantages in convolutional neural networks over to the conventional logistic or hyperbolic tangent activation functions. In Fig4 we represent the graphical structure of ReLU function.

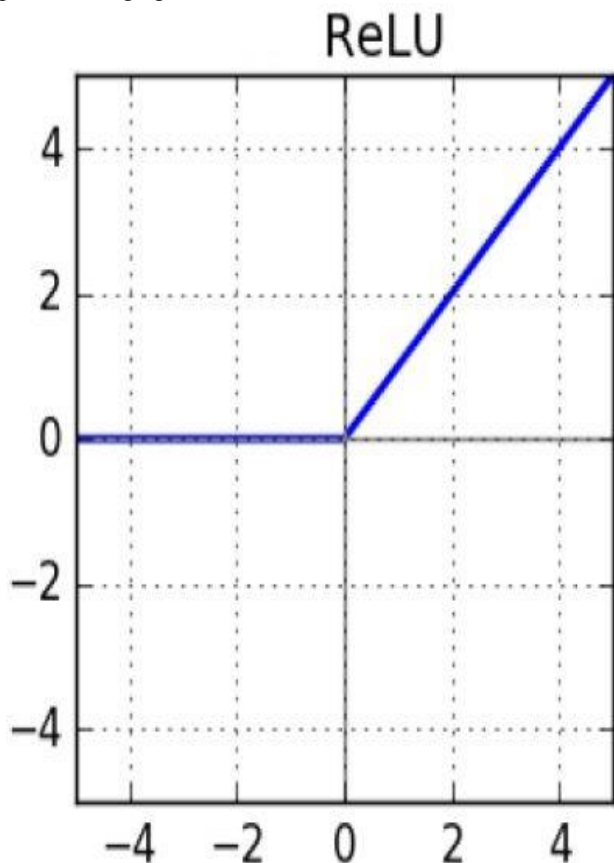


Fig. 4 Shows the graph of activation function (ReLU).

- Rectified linear units efficiently propagate the gradient and thus reduce the probability of a disappearing gradient problem typical in deep neural architectures.
- ReLU threshold negative values to zero, and thus resolve the problem of cancellation, resulting in a significantly sparse activation volume at its outcome. The sparsity is valuable for various reasons but primarily gives robustness to minor input changes such as noise.
- ReLU consist only of basic computational operations (mainly comparisons) and are much effective to implement in convolution neural networks.

IX. POOLING LAYER

It is the pooling or down sampling layer capable of decreasing the special size of the activation maps. In common, they are applied next multiple stages of different layers (i.e., convolution and non-linearity layers) to progressively reduce computational requirements across the network, as well as minimize the probability of overfitting. The pooling layer l holds two hyper parameters, the stride S^l and the spatial range of the filter F^l that gets an input volume of size $M_1^{(l-1)} \times M_2^{(l-1)} \times M_3^{(l-1)}$ and offers an output volume of a size $M_1^l \times M_2^l \times M_3^l$ where.

$$M_1^l = M_1^{(l-1)} \quad (12)$$

$$M_2^l = (M_2^{(l-1)} - F^l) / S^l + 1 \quad (13)$$

$$M_3^l = (M_3^{(l-1)} - F^l) / S^l + 1 \quad (14)$$

The pooling layer's key concept is to provide translational invariance since, especially in image identification tasks, the feature detection is more vital associated to the feature's specific location.

X. FULLY CONNECTED LAYER

In a convolutional network, the fully connected layers are efficiently a multilayer perceptron (usually a two- or three-layer MLP) that target to map the $M_1^{(l-1)} \times M_2^{(l-1)} \times M_3^{(l-1)}$ activation volume from the consolidation of several former layers into a class probability distribution. Hence, the multilayer perceptron's output layer would have $M_1^{(l-i)}$ outputs i.e. output neurons where i indicates the number of layers in the multilayer perceptron. The main difference from a typical multilayer perceptron is the input layer, where instead of a vector, an activation volume is taken as the input. As a result, the fully connected layer is described as:

If a fully connected layer is $l-1$:

$$y_i^l = f(Q_i^l) \text{ with } Q_i^l = \sum_{j=1}^{M_1^{l-1}} R_{i,j}^l y_i^{l-1} \quad (15)$$

Otherwise;

$$y_i^l = f(Q_i^l) \text{ with } Q_i^l = \sum_{j=1}^{M_1^{l-1}} \sum_{\alpha=1}^{M_2^{l-1}} \sum_{s=1}^{M_3^{l-1}} R_{i,j,\alpha,s}^l (Y_i^{l-1})_{\alpha,s} \quad (16)$$

The objective of the entire fully connected formation is to adjust the weight parameters $R_{i,j}^l$ or $R_{i,j,\alpha,s}^l$ to build a stochastic possibility representation of particular class based on the activation maps created by the concatenation of convolutional, rectification, nonlinear and pooling layers. However, individual fully connected layers work identically with multilayer perceptron including the unique exception holding the input layer. It is exceptional that the function f once again represents the non-linearity; despite, in a fully connected construction, the non-linearity is formed inside the neurons and is not a separate layer.

XI. ARCHITECTURE

In our proposed CNN model, we utilize three convolutional layers (Conv Layer1, Conv Layer2, Conv Layer3) three Maxpooling layers (Maxpooling1, Maxpooling2, Maxpooling3), and two fully connected layers (Dense Layer), which implies one hidden layer including one output layer fig2 shows the proposed CNN model. The range of RGB we normalize it from 0-255 to 0-1. Besides this, we assign the number of filters for every convolutional layer (32,32,64); we assigned the value of zero to the padding. The Fig5 represent the basis structure of CNN model.

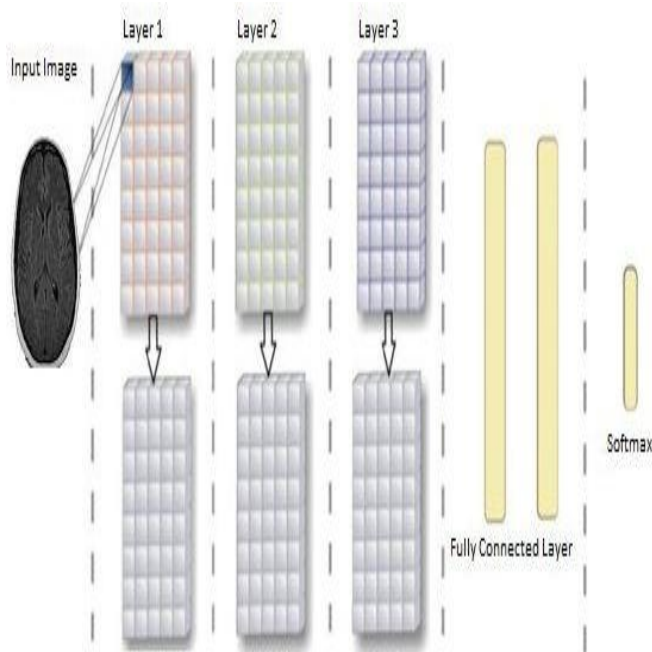


Fig. 5 Show the basic structure of CNN Model

As the CNN model's computational complexity, the testing and training time depended on the size and number of layers and filters. We empirically fixed the optimum number of filters by assessing work of [52]-[53]-[54]-[55]-[56]. To build the 1st, 2nd and 3rd convolutional layers, we utilized filter sizes of 3x3,3x3,3x3 and stride sizes are identical to all 2 ×2, respectively. However, the detailed summary of the model is defined in Fig6.

In [9]: CNN_model.summary()

Layer (type)	Output Shape	Param #	Connected to
convolution2d_1 (Convolution2D)	(None, 32, 128, 128)	320	convolution2d_input_1[0][0]
activation_1 (Activation)	(None, 32, 128, 128)	0	convolution2d_1[0][0]
convolution2d_2 (Convolution2D)	(None, 32, 126, 126)	9248	activation_1[0][0]
activation_2 (Activation)	(None, 32, 126, 126)	0	convolution2d_2[0][0]
maxpooling2d_1 (MaxPooling2D)	(None, 32, 63, 63)	0	activation_2[0][0]
dropout_1 (Dropout)	(None, 32, 63, 63)	0	maxpooling2d_1[0][0]
convolution2d_3 (Convolution2D)	(None, 64, 61, 61)	18496	dropout_1[0][0]
activation_3 (Activation)	(None, 64, 61, 61)	0	convolution2d_3[0][0]
maxpooling2d_2 (MaxPooling2D)	(None, 64, 30, 30)	0	activation_3[0][0]
dropout_2 (Dropout)	(None, 64, 30, 30)	0	maxpooling2d_2[0][0]
flatten_1 (Flatten)	(None, 57600)	0	dropout_2[0][0]
dense_1 (Dense)	(None, 64)	3686464	flatten_1[0][0]
activation_4 (Activation)	(None, 64)	0	dense_1[0][0]
dropout_3 (Dropout)	(None, 64)	0	activation_4[0][0]
dense_2 (Dense)	(None, 3)	195	dropout_3[0][0]

Total params: 3,714,723
 Trainable params: 3,714,723
 Non-trainable params: 0

Fig. 6 Represent the CNN model Summary

Inside the hidden layer, we utilize 128 neurons, furthermore in the output layer, we employ three neurons. The ReLu activation function is implemented to whole convolutional layers, including the fully connected layer. The coefficient of the ReLu activation function is set as 0.1 according to [57]. It is determined experimentally. Some relevant study can be found in [58]. In our work, we employ the softmax activation function inside the output layer. We practice a dropout layer on MaxPooling and the hidden layer, at the rate of 0.5. The dropout rate is picked experimentally. While training, an arbitrary number of neurons are excluded corresponding to the dropout rate. Throughout testing, the rate is multiplied with all the weights regarding the corresponding neurons of those layers. The below equation describes the convolutional process for each convolutional layer.

$$G_{\varphi}^t(y) = \left(\sum_{f=1}^c \sum_{k=1, x=n}^{k=a, x=n+a} (\omega_{\varphi}^t(k) \times G_f^{t-1}(x)) \right) + \beta_{\varphi} \quad (17)$$

Where $G_{\varphi}^t(y)$ indicates the pixel of t^{th} convolutional layer toward location y concerning the 3^{rd} filter $G_f^{t-1}(x)$ expresses the pixel of the past layer of t^{th} convolutional layer toward location x for specific f^{th} channel. In Equation (17), $\omega_{\varphi}^t(k)$ holds the value of the 3^{rd} filter regarding the t^{th} , convolutional layer through location k , a is the whole elements of the filter $\omega_{\varphi}^t(k)$, β_{φ} and holds the bias term of the 3^{rd} filter to our model. Besides this, in our study, we fixed the number of filters ϕ for every convolutional layer as 32. Equation (18) describes the method of ReLu activation function.

$$\mu_f^t(y) = \max(\mu_f^t(y), \mu_f^t(y) \times 0.1) \quad (18)$$

Where $\mu_f^t(y)$ belongs to the pixel after implementing the process of the ReLu activation function toward location y for the f^{th} channel of the t^{th} convolutional layer. The Maxpooling layer is represented in Equation (19).

$$Z_f^t(y) = \frac{1}{t_h t_w} \sum_{x=1}^{x=t_h t_w} \mu_f^t(x) \quad (19)$$

Where $Z_f^t(y)$ shows the pixel on location y after employing the method of Maxpooling in the t^{th} convolutional layer for the f^{th} channel, where t_h and t_w indicate the image patch width and height, sequentially. From Equation (20) - (23), we explain the process of a fully connected layer and an output layer.

$$\delta_c = \sum_k^E (\varphi_k \times \omega_{k,l}) + \beta_l \quad (20)$$

$$A_c = \max(\delta_c, \delta_c \times 0.1) \quad (21)$$

$$\delta_d = \sum_{c=1}^{\tau} (A_c \times \omega_{c,d}) + \beta_d \quad (22)$$

$$V_d = \frac{\exp(\delta_d)}{\sum_{d=1}^3 \exp(\delta_d)} \quad (23)$$

Next the linking of the Maxpooling layers took place, the features are transformed into a matrix form. We address them a single-vector form, moreover, they are the inputs over the fully connected layer. In our proposed model, we utilize one (hidden layer) fully connected layer. While in Equation (20), φ_k holds the input feature vector $\omega_{k,l}$ is the weight of the input feature through the l^{th} hidden layer neuron, including specific β_l bias term of l^{th} hidden layer neuron. Meanwhile, δ_c and E designate the input toward the l^{th} hidden layer neuron and the whole number of input features, sequentially. In Equation (21), A_c the hidden layer denotes the output of the c^{th} neuron after a specific process of the ReLu activation function. In Equation (22), δ_d denotes the input toward d^{th} neuron in the output layer. Besides, we have to predict three kinds of brain images, hence in the output layer, the cumulative number of output neurons is three. to this instance, $\omega_{c,d}$, β_d , and τ express the weight of the c^{th} neuron (hidden layer) to the d^{th} neuron (output layer), bias term of each d^{th} neuron inside the output layer, and this whole number of neurons within the hidden layer, sequentially. We employed 128 hidden layer neurons (τ). In Equation (23), V_d holds the softmax output of the d^{th} neuron in the output layer.

XII. DEEP NEURAL NETWORK

A (DNN) deep neural network is a multilayered artificial neural network (ANN) among particular input and output layers. The DNN finds the accurate numerical manipulation to transform the input into the output, either in a non-linear or linear relationship. The network flows through the layers, assessing the probability of specific output. In our study, we employ a DNN model that is trained to recognized tumor category as well as the non-existing tumor. Furthermore, calculate the probability outcomes belonging to every category respectively. Every mathematical manipulation as such is named a layer, and complex DNNs hold several layers. Usually, DNNs are feedforward networks where data flows from the input layer to a particular output layer without looping backwards. The Fig7 shows the basic structure of DNN network.

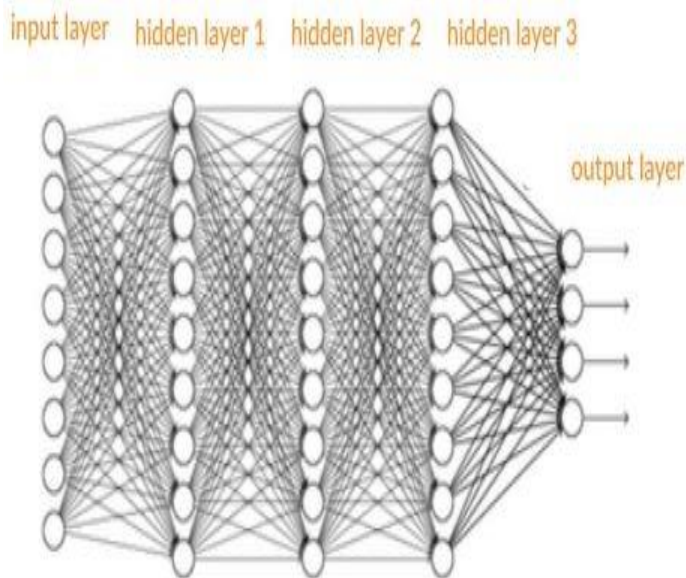


Fig. 7 Show the basic structure of DNN.

The DNN initially generates a map of virtual neurons and selects arbitrary numerical values, or "weights," to links among them. The input and weights are calculated and return an output amid 1 and 0. If the network did not correctly recognize any appropriate sequence, an algorithm would change the weights. Such way, the algorithm can select specific parameters more influential until it discovers the exact mathematical manipulation to process the data fully. In our study, we employed multiple layers in our DNN model. The input layer, hidden layers, and the last one is the output layer. The first hidden layers hold 128 number of perceptron's, the second one own 64, and the third one has 32. The last output layer occupies three perceptron's for giving the predicted output. The detail summary of the model is given in Fig8.

In [10]: DNN_model.summary()

Layer (type)	Output Shape	Param #	Connected to
flatten_2 (Flatten)	(None, 16384)	0	flatten_input_2[0][0]
dense_5 (Dense)	(None, 128)	2097280	flatten_2[0][0]
dropout_4 (Dropout)	(None, 128)	0	dense_5[0][0]
dense_6 (Dense)	(None, 64)	8256	dropout_4[0][0]
dropout_5 (Dropout)	(None, 64)	0	dense_6[0][0]
dense_7 (Dense)	(None, 32)	2080	dropout_5[0][0]
dropout_6 (Dropout)	(None, 32)	0	dense_7[0][0]
dense_8 (Dense)	(None, 3)	99	dropout_6[0][0]

Total params: 2,107,715
 Trainable params: 2,107,715
 Non-trainable params: 0

Fig.8 Represent the DNN model summary

Furthermore, to reduce training time, we resized the input images to 128x128. Hereafter, the input shape is 128x128x1 number of rows and columns along with a single channel. The first layer flattens the input to one-dimension tensor, which is then fed into dense layers. Optimizer function RMSprop are utilized in our proposed method further details of the optimizer are given below.

XIII. RMSPROP OPTIMIZER

The RMSprop optimizer acts similar to the gradient descent algorithm. The RMSprop optimizer limits the oscillations toward a specific vertical direction. RMSprop adopts an adaptive learning rate rather than using the learning rate as a hyper parameter, indicating that the learning rate varying over time. The distinction between RMSprop and gradient descent is approaching how the gradients are calculated. RmsProp is an optimizer that employs a particular magnitude of new gradients to normalize the gradients. We continuously keep a moving average across the root mean squared gradients, through which we divide a specific current gradient. Let $D'(\theta_t)$ will be the derivative from the loss with respect to the parameters toward time step t. In the primary form, assign a step rate η including a decay term λ we achieve the following updates.

$$r_t = (1 - \lambda)D'(\theta_t)^2 + \lambda r_{t-1} \quad (24)$$

$$h_{t+1} = \frac{\eta}{\sqrt{r_t}} D'(\theta_t) \quad (25)$$

$$\theta_{t+1} = \theta_t - h_{t+1} \quad (26)$$

While some cases, computing a momentum term β is useful.

$$\theta_{t+\frac{1}{2}} = \theta_t - \beta h_t \quad (27)$$

$$r_t = (1 - \lambda)D'(\theta_{t+\frac{1}{2}})^2 + \lambda r_{t-1} \quad (28)$$

$$h_{t+1} = \beta h_t + \frac{\eta}{\sqrt{r_t}} \quad (29)$$

$$\theta_{t+1} = \theta_t - h_{t+1} \quad (30)$$

Additionally, that implementation holds adaptable step rates since the components of the step, including the momentum point in the corresponding direction (consequently have the identical sign) the step rate toward that parameter is multiplied with one plus step-adapt. Otherwise, it is multiplied with one minus step-adapt. However, the maximum and minimum step rates step rate-max and step rate-min are respected, and exceeding values truncated to it. RmsProp it is a quite robust optimizer which associates pseudo curvature. Additionally, it can also deal very adequately with stochastic objectives, addressing this applicable to mini-batch learning.

XIV. RESNET-50 NEURAL NETWORK

In CNN's, after the convolution process, those images from diverse datasets distribute similar low-level features. Performing training from scratch is not economical, particularly on a small range dataset. A new dataset's training strategy employs parameters that are transferred from pre-trained models, fine-tuned based on the initial dataset. Therefore, we utilized pre-trained models to produce our task. ResNet-50 implies the abbreviated form for Residual Network. Throughout the year's deep CNN hold produced a series of breakthroughs in the area of an image classification and recognition. It becomes a trend to go deeper to resolve further complicated problems and advance classification or accuracy in recognition. Furthermore, training deeper neural networks has been challenging due to the degradation problem and vanishing gradient problem. For neural networks, each layer learns low or high-level features while continuing to train for the task. Meanwhile, residual learning, rather than trying to determine features, the model serves to learn some residual. As see in Fig9, specific input 'x' is added as residue to the weight layers' output, and the activation is carried out.

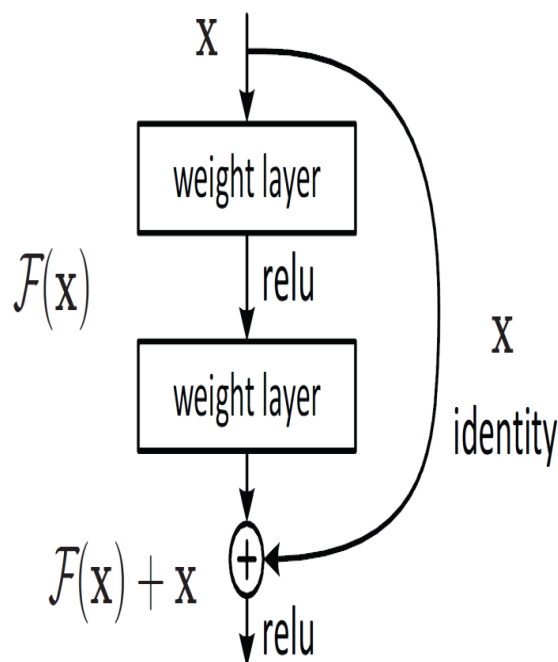


Fig. 9 Represent the basic structure of Residual layer.

Residual Networks (ResNet) hold deep convolutional networks where the primary approach is to skip blocks of convolutional layers by utilizing alternative connections. The main blocks, called "bottlenecks," follow two fundamental principles of design. During the identical map size of the output feature, the layers own the equal number of filters. Whereas if the feature map size is half, the quantity of filters is twice. The down-sampling is delivered direct by convolutional layers that hold a stride of two, furthermore batch normalization is executed right after every convolution plus before ReLu activation.

However, if the input and output are of the equivalent dimensions, then identity shortcut is utilized. When the dimensions were increasing, the projection bypass is used to meet dimensions through 1×1 convolutions. Meanwhile, both cases, if shortcuts move crossed the feature maps of two sizes, then they are delivered with a stride of 2. The ResNet-50 network concludes with a 1,000 fully-connected (fc) layer including softmax activation. However, the total quantity of weighted layers is 50, including 23,534,592 trainable parameters. The architecture of the primary ResNet-50 is demonstrated in Fig10.

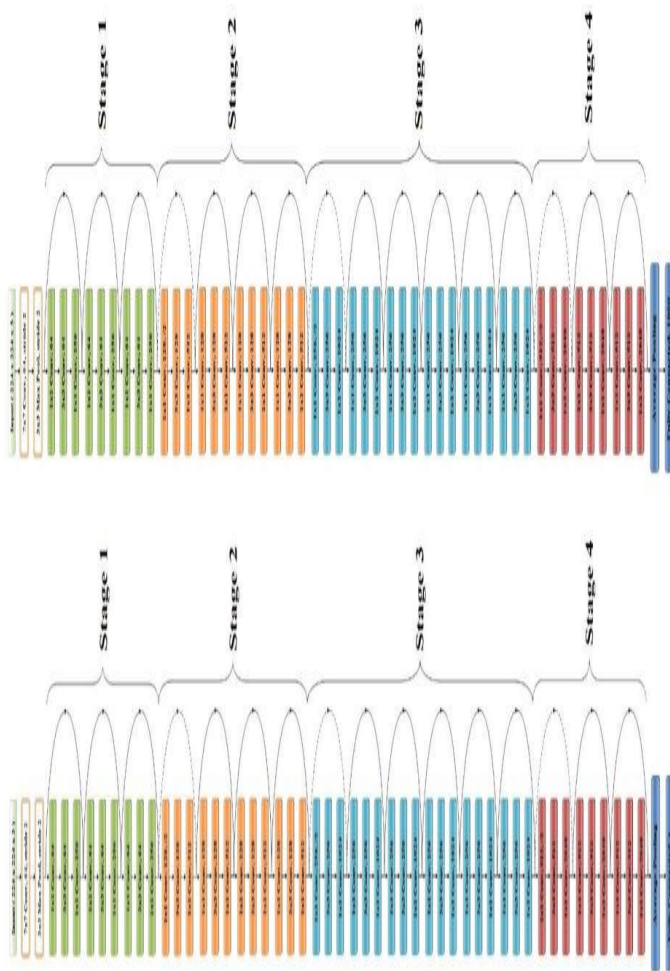


Fig. 10 Demonstrate the architecture of primary and advanced ResNet-50

In our study Pre-Trained weights are to be imported for ResNet-50 model. Furthermore, by utilizing the pre-trained weights, the input data is trained, and the only layer, which implies learning with backpropagation holds the dense layer as shown in Fig10. In our study, From Fig10 second image, the last fully connected layer, we declare fc3 because we have three different category data sets to classify. The base of our model architecture is the same as the primary ResNet-50 in Fig10 except for the last fully connected layer, the model summary, as shown in Fig11.

activation_46 (Activation)	(None, 7, 7, 2048)	0	add_15[0][0]
res5c_branch2a (Conv2D)	(None, 7, 7, 512)	1049088	activation_46[0][0]
bn5c_branch2a (BatchNormalizatio	(None, 7, 7, 512)	2048	res5c_branch2a[0][0]
activation_47 (Activation)	(None, 7, 7, 512)	0	bn5c_branch2a[0][0]
res5c_branch2b (Conv2D)	(None, 7, 7, 512)	2359808	activation_47[0][0]
bn5c_branch2b (BatchNormalizatio	(None, 7, 7, 512)	2048	res5c_branch2b[0][0]
activation_48 (Activation)	(None, 7, 7, 512)	0	bn5c_branch2b[0][0]
res5c_branch2c (Conv2D)	(None, 7, 7, 2048)	1050624	activation_48[0][0]
bn5c_branch2c (BatchNormalizatio	(None, 7, 7, 2048)	8192	res5c_branch2c[0][0]
add_16 (Add)	(None, 7, 7, 2048)	0	bn5c_branch2c[0][0] activation_46[0][0]
activation_49 (Activation)	(None, 7, 7, 2048)	0	add_16[0][0]
avg_pool (AveragePooling2D)	(None, 1, 1, 2048)	0	activation_49[0][0]
flatten (Flatten)	(None, 2048)	0	avg_pool[0][0]
output (Dense)	(None, 3)	6147	flatten[0][0]
Total params: 23,593,859			
Trainable params: 6,147			
Non-trainable params: 23,587,712			

Fig. 11 Represent the summary of the last (fc layer) of our ResNet-50 model.

The ResNet-50 layer input image should be of the size 224 * 224. Therefore, all the images must be re-sized to the spot size of 224*224. We adopted stochastic gradient descent (SGD) optimizer. SGD optimizer has confirmed to be functioning more reliable than several other optimizers. In our study we utilize the sigmoid activation function. The intention to adopt sigmoid activation other than softmax is that the task which is to deliver is a binary classification of images and sigmoid acts considerably with binary classification. The mathematical description of sigmoid activation is shown in Equation (31) below where x denotes the dot product of individual neuron value with the weights.

$$\rho(x) = \frac{1}{1 + e^{-x}} \quad (31)$$

XV. SEGMENTATION

The next stage is implementing the segmentation process after the image classification from neural networks here; we embedded the condition if the dataset has the tumor either (Benign or Malignante), the segmentation process is executed; otherwise, the algorithm is terminated. For implementation, the segmentation and extract, the tumor following method are utilized.

XVI. MEDIAN FILTER

In our study, a non-linear Median filter technique is adopted, which exceptionally enhances the quality of the image. The filter is serving by moving through the image pixel by pixel replacing the individual value by a specific median value of neighboring pixels. The arrangement of neighbor's is termed the window that drifts pixel by pixel over the entire image. Moreover, the filter is working by shifting in the image pixel by pixel swapping the value with a specific median value of neighboring pixels. The sequence of neighbor's is named as the window, which moves pixel by pixel over the complete image. The Window in numerical order modified the existing pixel with taking in considering the (median) middle pixel value. It is more robust to edges of the images. Since the median is a pixel value that can be traced from the pixel neighborhood itself that accommodates to reduce the damage of image detail as well as edge blurring. The replacing manner of the pixel's value is show in Figure 12.

Sorted in ascending order (001112244).

Input Image pixels' value	Output Image pixels' value
1 4 0 1 3 1	1 4 0 1 3 1
2 2 4 2 2 3	2 1 1 1 1 3
1 0 1 0 1 0	1 1 1 1 2 0
1 2 1 0 2 2	1 1 1 1 1 2
2 5 3 1 2 5	2 2 2 2 2 5
1 1 4 2 3 0	1 1 4 2 3 0

Fig. 12 Replacing method of the pixel's value is shown.

XVII. OTSU THRESHOLDING

Digital image segmentation remains one of the necessary steps toward computer vision and image processing. It serves in splitting the pixels toward diverse regions according to their intensity level. During segmentation, many methods have been suggested, and some of them practice complex computational processes. However, the most favored method that can be employed satisfactorily is the Otsu thresholding method, amongst others. This paper proposes a distinct heuristic approach to image segmentation that resolves multilevel thresholds automatically by analyzing a digital image histogram. Our method's importance is to determine a valley as an optimal threshold value. Meanwhile, in term of computing time and outcome, the Otsu thresholding technique is considerably more robust than different segmentation technique. Furthermore, the Otsu thresholding technique is applied to conclude the image threshold and apply the labels, respectively. Here, the thresholding method assumes that the histogram intensity should be bi-modal if non, then modify it into one. However, minimizes the alteration intra-classes and maximizer the inter-classes alteration effects in a distinct foreground and background.

The mathematical implementation of Otsu is as follow.

$$\sigma_{\alpha}^2(t) = \alpha_0(t)\sigma_0^2(t) + \alpha_1(t)\sigma_1^2(t) \quad (32)$$

$$\alpha_0(t) = \sum_{a=0}^{t-b} h(a) \quad (33)$$

$$\alpha_1(t) = \sum_{a=t}^{b-1} h(a) \quad (34)$$

Here, h(a) is the probability of single pixels' intensity. Furthermore, Otsu's approach symbolizes that minimizing the intra-class variance effect maximizes the inter-class variance, as presented below.

$$\sigma_d^2(t) = \sigma^2 - \sigma_{\alpha}^2(t) \quad (35)$$

XVIII. SKULL STRIPPING OF BRAIN TUMOR SEGMENTATION.

In the domain of image processing, Skull Stripping holds one of the primary tools in which the tissues formed of white and gray matter are isolated from the skull. While in MRI images, quantitative morphometric examinations of MR brain images usually need preliminary processing to isolate the brain from extra-cranial or non-brain tissues, generally referred to as skull stripping. The current method is deeply dependent on specific geometric hypotheses.

$$\sigma_d^2(t) = \alpha_0(G_0 - G_r)^2 + \alpha_1(G_1 - G_r)^2 \quad (36)$$

$$\sigma_d^2(t) = \alpha_0(t)\alpha_1[G_0(t) - G_1(t)]^2 \quad (37)$$

XIX. COMPARISON

The DNN, CNN and ResNet-50 performance of the algorithm can be considered in terms of Accuracy, Recall, Precision, F1 score, Training accuracy vs Validation accuracy and Training losses vs Validation losses.

XX. RECALL

A test can diagnose all those accurately having a specific disease. Mathematically it can be determined as in Equation (29).

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (38)$$

XXI. F1 SCORE

F1 score merges precision and recalls related to a particular positive class. The F1 score can be represented as a weighted average of recall and precision, where an F1 score approaches its greatest value at one and worst at 0.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (39)$$

XXII. ACCURACY

The Accuracy holds one metric for assessing classification models. Informally, Accuracy is a specific fraction of predictions our model got right. However, Accuracy has the following description.

$$Accuracy = \frac{Number\ of\ Correct\ Prediction}{Total\ Number\ of\ Prediction}$$

As a binary classification, accuracy can also be determined in terms of negative and positive as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (40)$$

XXIII. PRECISION

Precision is described as the fraction of relevant instances between all retrieved instances.

$$precision = \frac{TP}{TP + FP} \quad (41)$$

However, where $TP =$ True Positives, $TN =$ True Negatives, $FP =$ False Positives, and $FN =$ False Negatives.

XXIV. RESULT AND DISCUSSION

In our study, the performance analysis of our models (DNN, CNN and ResNet-50) is done by experimenting individually preferably of on using the MRI dataset. The distribution of a dataset is shown in Fig 2. We evaluate our models on the three group order dataset (No tumor, Benign tumor and Malignant tumor) that holds about 4162 images. The 80% of data were used for training while the remaining 20% were used for testing the model as shown above in Fig 3.

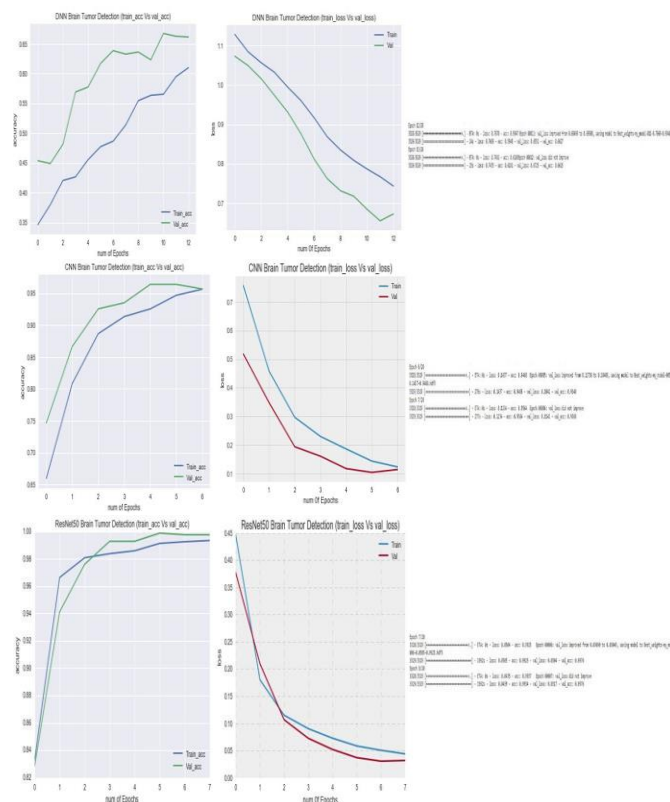


Fig. 13 Shows the Parameters of DNN, CNN and ResNet-50

In our study, we adopt the callback function to the overall models and monitoring the validation losses, as shown in Fig 13. However, from Fig 13, the three models are compared, and we assessed that the higher accuracy and minimum validation losses achieved by ResNet-50 model

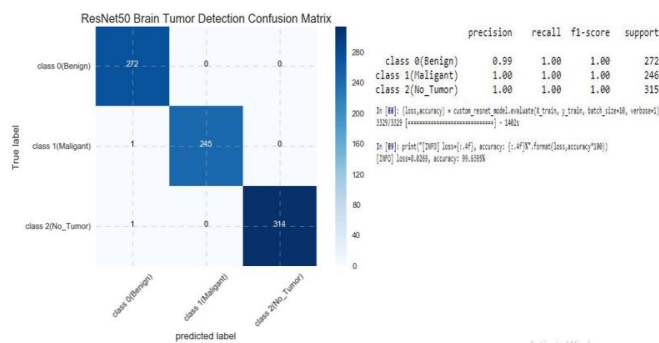
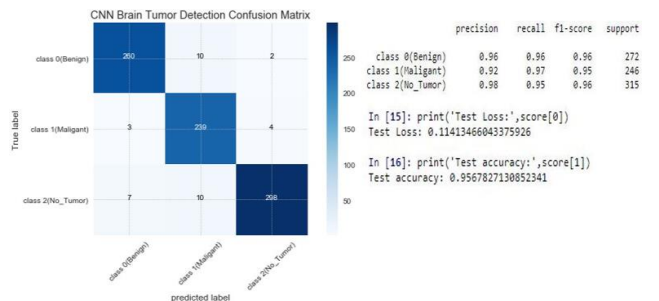
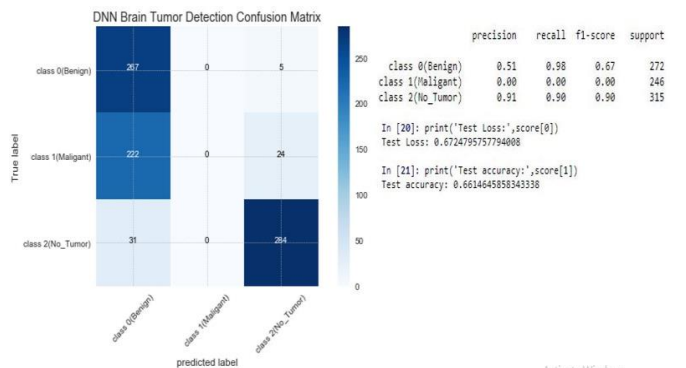


Fig.14 Represent the Confusion Matrix of DNN,CNN & ResNet-50

Furthermore, from Fig 14, we declared a confusion matrix of our models, which describe the efficacy of models classification. Moreover, the parameters of our models are shown in Fig 14. After the classification of a dataset next, we performed the segmentation method as shown in Fig14. However, if the model classifies the MRI image that having a

tumor either (benign or malignant), the segmentation process executed; otherwise, the algorithm is terminated. In our proposed method, we adopted the Otsu method for the segmentation and have extracted tumor.

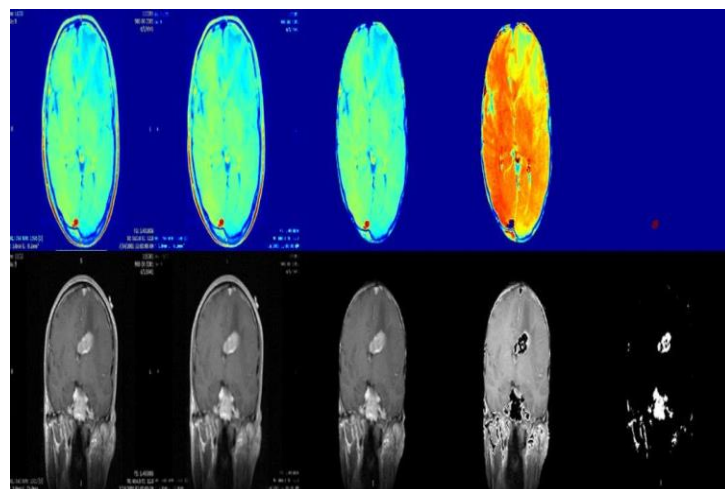


Fig. 15 Shows the segmentation process of MRI dataset

From Fig 15 firstly, the median filter is applied to enhance the input image. The next phase to implemented the skull strip method to eliminate the unwanted area after performing the skull stripping. Next, we employed OSTU algorithm, and lastly, we got the extracted tumor on the output. In our study, we evaluated from Fig 13 that the DNN model improved the validation loss from 1.07274 to 0.6627, and CNN update from 0.51781 to 0.1041, while ResNet-50 succeed to reduce the validation loss from 0.37695 to 0.0304. Despite, from Tabell, we approximate the result; there are three class of dataset class 0, class 1, and class 2. Additionally, the main objective of the results is given below.

Tabel.1 Represent the overall parameters of our proposed models

Comparison of Overall Results						
	Precision	Recall	F1 Score	Test Accuracy	Test Loss	Support
Class 0 (Benign)						
DNN	0.51	0.98	0.67	0.6614	0.6724	272
CNN	0.96	0.96	0.96	0.9567	0.1141	
ResNet-50	0.99	1.00	1.00	0.9963	0.0269	
Class 1 (Malignant)						
DNN	0.0	0.0	0.0	0.6614	0.6724	246
CNN	0.92	0.97	0.95	0.9567	0.1141	
ResNet-50	1.00	1.00	1.00	0.9963	0.0269	
Class 2 (No Tumor)						
DNN	0.91	0.90	0.90	0.6614	0.6724	315
CNN	0.98	0.95	0.96	0.9567	0.1141	
ResNet-50	1.00	1.00	1.00	0.9963	0.0269	

- The DNN model failed to detect the overall class 1. It needs to be further improved by adding some more layers.
- The CNN result for overall classes is considered satisfactory, but it requires advanced in terms of F1 score and test losses.
- However, it is explicit that the ResNet-50 delivered a higher precision, recall, accuracy rate with minimum test loss, along with holding the best F1 score for all classes.

XXV. CONCLUSION

In past studies, both objective features and traditional subjective features are practiced. Objective features hold features extracted, while Traditional subjective features involve morphology, density, and texture. Those features exist flawed to some extent. In this study, we consolidate subjective and objective features, considering the doctor's expertise and the required attributes of the Brain Tumor account at the same time. Next, extracting the features, each model is applied to classify the Tumor mass's benign or malignant. The main contribution of this paper is to produce an automated system that can segment and classify three classes, namely no tumor,

benign tumor and malignant tumor. In our study, we segmented brain MR images and extracted the tumor from the MRI dataset. Also, we adopted 1212 patients suffering from the Malignant tumor and 1211 from the Benign tumor while 1739 are healthy patients. We registered three major types of MRI images T1-weighted, T2-weighted usually mentioned as T1(tissue type is Bright & Fat), T2 (tissue types are bright, Fat & Water) and Flair images(Fluid Aattenuated Inversion Rrecovery). After filtering with a median filter we design an algorithm that merges threshold and morphological methods for stripping the skull that is not our area of interest. furthermore, we adopt the Otsu method for image segmentation. The paper introduces three models of neural networks DNN, CNN and ResNet-50 for the same data samples, the most robust and advanced we discovered is ResNet-50 model that owns a maximum test accuracy 0.996, precision 1.00 with best F1 score 1.0 including minimum validation losses 0.0269 for medical images, which is presented for the follow-up development specialist diagnosis method and self-checking technique, this provides strong technical support. The advanced ResNet-50 model outcomes prove that its accuracy and efficacy are competing the offered two models in the paper. In our future work it would be considerably further interesting to investigate the possibility of merging the ResNet-50 with several other auto-encoder modifications to see the outcome or performance in the same Mri brain dataset also focus on improving reducing the time taken by the algorithm through adopting advanced computer and GPU Additionally.

- Extending the number of classes, which could accommodate more knowledge on the classes of tumors types.
- Examine the classification capacity of the pre-trained ResNet-50 on some further datasets including CT and Pet-scan.

ACKNOWLEDGMENT

The authors would like to thanks members of the Diagnostic image analysis group for discussions and suggestions. The research was funded by grants. This work is supported by the **National Natural Science Foundation of Hebei province under grant No: E2015202292 and No: E2015202050**, high-level talent support project in **Hebei province under grant No: C2015005012**, **Key research and development program under grant No: 15272002 and No: 15275704**.

XXVI. REFERENCES

- [1] S. Stolte and R. Fang, "A survey on medical image analysis in diabetic retinopathy," *Med. Image Anal.*, vol. 64, p. 101742, Aug. 2020.
- [2] K. D. Toennies, *Guide to Medical Image Analysis*, vol. 25. London: Springer London, 2017.
- [3] G. J. Aulisio, "Common Sense, the Turing Test, and the Quest for Real AI," *Eur. Leg.*, vol. 25, no. 1, pp. 105–107, Jan. 2020.
- [4] K. Kaplan, Y. Kaya, M. Kuncan, and H. M. Ertunç, "Brain tumor classification using modified local binary patterns (LBP) feature extraction methods," *Med. Hypotheses*, vol. 139, p. 109696, Jun. 2020.
- [5] A. B. Câmara-de-Souza *et al.*, "Insulinoma: A retrospective study analyzing the differences between benign and malignant tumors," *Pancreatology*, vol. 18, no. 3, pp. 298–303, Apr. 2018.
- [6] S. Türkoğlu and M. Kayan, "Differentiation between benign and malignant ovarian masses using multiparametric MRI," *Diagn. Interv. Imaging*, vol. 101, no. 3, pp. 147–155, Mar. 2020.
- [7] M. Lather and P. Singh, "Investigating Brain Tumor Segmentation and Detection Techniques," *Procedia Comput. Sci.*, vol. 167, pp. 121–130, 2020.
- [8] S. John *et al.*, "Brain Imaging Using Mobile CT: Current Status and Future Prospects," *J. Neuroimaging*, vol. 26, no. 1, pp. 5–15, Jan. 2016.
- [9] L. Cordero-Grande, E. J. Hughes, J. Hutter, A. N. Price, and J. V. Hajnal, "Three-dimensional motion corrected sensitivity encoding reconstruction for multi-shot multi-slice MRI: Application to neonatal brain imaging," *Magn. Reson. Med.*, vol. 79, no. 3, pp. 1365–1376, Mar. 2018.
- [10] A. Sasikumar *et al.*, "Diagnostic Value of ⁶⁸Ga PSMA-11 PET/CT Imaging of Brain Tumors—Preliminary Analysis," *Clin. Nucl. Med.*, vol. 42, no. 1, pp. e41–e48, Jan. 2017.
- [11] A. E. Campbell-Washburn *et al.*, "Opportunities in Interventional and Diagnostic Imaging by Using High-Performance Low-Field-Strength MRI," *Radiology*, vol. 293, no. 2, pp. 384–393, Nov. 2019.
- [12] A. Işin, C. Direkoğlu, and M. Şah, "Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods," *Procedia Comput. Sci.*, vol. 102, pp. 317–324, 2016.
- [13] D. S. Meier *et al.*, "Dual-Sensitivity Multiple Sclerosis Lesion and CSF Segmentation for Multichannel 3T Brain MRI," *J. Neuroimaging*, vol. 28, no. 1, pp. 36–47, Jan. 2018.
- [14] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson, "Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions," *Journal of Digital Imaging*. 2017.
- [15] S. Aja-Fernández and G. Vegas-Sánchez-Ferrero, *Statistical Analysis of Noise in MRI*. Cham: Springer International Publishing, 2016.
- [16] G. Mohan and M. M. Subashini, "MRI based medical image analysis: Survey on brain tumor grade classification," *Biomed. Signal Process. Control*, vol. 39, pp. 139–161, Jan. 2018.
- [17] M. Roomi, T. Kalinovsky, M. Rath, and A. Niedzwiecki, "A Specific Mixture of Nutrients Suppresses Ovarian Cancer A-2780

- Tumor Incidence, Growth, and Metastasis to Lungs,” *Nutrients*, vol. 9, no. 3, p. 303, Mar. 2017.
- [18] A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, “A Supervised Learning Algorithm for Learning Precise Timing of Multiple Spikes in Multilayer Spiking Neural Networks,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 11, pp. 5394–5407, Nov. 2018.
- [19] C. Tor-Díez, N. Passat, I. Bloch, S. Faisan, N. Bednarek, and F. Rousseau, “An iterative multi-atlas patch-based approach for cortex segmentation from neonatal MRI,” *Comput. Med. Imaging Graph.*, vol. 70, pp. 73–82, Dec. 2018.
- [20] H. Ravishankar, R. Venkataramani, S. Thiruvenkadam, P. Sudhakar, and V. Vaidya, “Learning and Incorporating Shape Models for Semantic Segmentation,” 2017, pp. 203–211.
- [21] A. L. Beam and I. S. Kohane, “Big Data and Machine Learning in Health Care,” *JAMA*, vol. 319, no. 13, p. 1317, Apr. 2018.
- [22] I. E. Kaya, A. Ç. Pehlivanlı, E. G. Sekizkardeş, and T. Ibrikci, “PCA based clustering for brain tumor segmentation of T1w MRI images,” *Comput. Methods Programs Biomed.*, vol. 140, pp. 19–28, Mar. 2017.
- [23] M. Gesho, W. Chaisontornyotin, O. Elkhatib, and L. Goual, “Auto-Segmentation Technique for SEM Images Using Machine Learning: Asphaltene Deposition Case Study,” *Ultramicroscopy*, p. 113074, Jul. 2020.
- [24] R. F. Shah, A. M. Martinez, V. Pedroia, S. Majumdar, T. P. Vail, and S. A. Bini, “Variation in the Thickness of Knee Cartilage. The Use of a Novel Machine Learning Algorithm for Cartilage Segmentation of Magnetic Resonance Images,” *J. Arthroplasty*, vol. 34, no. 10, pp. 2210–2215, Oct. 2019.
- [25] D. Komura and S. Ishikawa, “Machine Learning Methods for Histopathological Image Analysis,” *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 34–42, 2018.
- [26] Y. Ding *et al.*, “Multi-modal brain tumor image segmentation based on SDAE,” *Int. J. Imaging Syst. Technol.*, vol. 28, no. 1, pp. 38–47, 2018.
- [27] Yi Ding, Cong Zhang, Tian Lan, Zhiguang Qin, Xinjie Zhang, and Wei Wang, “Classification of Alzheimer’s disease based on the combination of morphometric feature and texture feature,” in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015, pp. 409–412.
- [28] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, “Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images,” *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1240–1251, May 2016.
- [29] K. Kamnitsas *et al.*, “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,” *Med. Image Anal.*, vol. 36, pp. 61–78, Feb. 2017.
- [30] “Applications of deep learning to MRI images: A survey,” *Big Data Min. Anal.*, vol. 1, no. 1, pp. 1–18, Mar. 2018.
- [31] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” *Heliyon*, vol. 4, no. 11, p. e00938, Nov. 2018.
- [32] X. Bai, C. Wang, and Z. Tian, “Self-adaptive Superpixels based on Neural Network Models,” *IEEE Access*, pp. 1–1, 2020.
- [33] H. Mohsen, E.-S. A. El-Dahshan, E.-S. M. El-Horbaty, and A.-B. M. Salem, “Classification using deep learning neural networks for brain tumors,” *Futur. Comput. Informatics J.*, vol. 3, no. 1, pp. 68–71, Jun. 2018.
- [34] J. J. Titano *et al.*, “Automated deep-neural-network surveillance of cranial images for acute neurologic events,” *Nat. Med.*, vol. 24, no. 9, pp. 1337–1341, Sep. 2018.
- [35] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, “Brain Tumor Classification Using Convolutional Neural Network,” 2019, pp. 183–189.
- [36] Y. Mao and Z. Yin, “A Hierarchical Convolutional Neural Network for Mitosis Detection in Phase-Contrast Microscopy

- Images,” 2016, pp. 685–692.
- [37] A. Rana, G. Yaune, A. Lowe, and P. Shah, “Computational Histological Staining and Destaining of Prostate Core Biopsy RGB Images with Generative Adversarial Neural Networks,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 828–834.
- [38] H.-C. Shin, Le Lu, L. Kim, A. Seff, J. Yao, and R. M. Summers, “Interleaved text/image Deep Mining on a large-scale radiology database,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1090–1099.
- [39] P. Tang, H. Wang, and S. Kwong, “G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition,” *Neurocomputing*, vol. 225, pp. 188–197, Feb. 2017.
- [40] Y. Jiang, L. Chen, H. Zhang, and X. Xiao, “Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module,” *PLoS One*, vol. 14, no. 3, p. e0214587, Mar. 2019.
- [41] D. Shaikhislamov, A. Sozykin, and V. Voevodin, “Survey on software tools that implement deep learning algorithms on intel/x86 and IBM/Power8/Power9 platforms,” *Supercomput. Front. Innov.*, vol. 6, no. 4, pp. 57–83, Dec. 2019.
- [42] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, “Deep Label Distribution Learning With Label Ambiguity,” *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2825–2838, Jun. 2017.
- [43] J. C. Caicedo *et al.*, “Data-analysis strategies for image-based cell profiling,” *Nat. Methods*, vol. 14, no. 9, pp. 849–863, Sep. 2017.
- [44] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, “Going Deeper in Spiking Neural Networks: VGG and Residual Architectures,” *Front. Neurosci.*, vol. 13, Mar. 2019.
- [45] T. Shanthi and R. S. Sabeenian, “Modified Alexnet architecture for classification of diabetic retinopathy images,” *Comput. Electr. Eng.*, vol. 76, pp. 56–64, Jun. 2019.
- [46] M. A. Gülsün, G. Funke-Lea, P. Sharma, S. Rapaka, and Y. Zheng, “Coronary Centerline Extraction via Optimal Flow Paths and CNN Path Pruning,” 2016, pp. 317–325.
- [47] The Cancer Imaging Archive (TCIA), “{TCGA-LUSC} - The Cancer Imaging Archive ({TCIA}) Public Access - Cancer Imaging Archive Wiki,” 2015. [Online]. Available: <https://wiki.cancerimagingarchive.net/display/Public/REMBRANDT>. [Accessed: 27-Jul-2020].
- [48] K. Clark *et al.*, “The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository,” *J. Digit. Imaging*, vol. 26, no. 6, pp. 1045–1057, Dec. 2013.
- [49] 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,” *J. Comput. Sci.*, vol. 30, pp. 174–182, Jan. 2019.
- [50] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019.
- [51] V. A. Sindagi and V. M. Patel, “A survey of recent advances in CNN-based single image crowd counting and density estimation,” *Pattern Recognit. Lett.*, vol. 107, pp. 3–16, May 2018.
- [52] X. Zheng, Y. Wang, G. Wang, and J. Liu, “Fast and robust segmentation of white blood cell images by self-supervised learning,” *Micron*, vol. 107, pp. 55–71, Apr. 2018.
- [53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [54] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 2261–2269.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp.

- 770–778, Jun. 2016.
- [56] G. Liang, H. Hong, W. Xie, and L. Zheng, “Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification,” *IEEE Access*, vol. 6, pp. 36188–36197, 2018.
- [57] Fannjiang, C. “Nonlinear activations for convolutional Neural Network acoustic models.” (2016).
- [58] A. Tashk, J. Herp, and E. Nadimi, “Automatic segmentation of colorectal polyps based on a novel and innovative convolutional neural network approach,” *WSEAS Trans. Syst. Control*, vol. 14, pp. 384–391, 2019.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US

Author Contributions:

Imran Javaid: Writing - original draft, Visualization, Investigation, Software, Validation.

Shuai Zhang: Writing - review & editing.

Abd El Kader Isselmou: Conceptualization, Methodology, Supervision.

Souha Kamhi: Writing -review & editing.

Isah Salim Ahmad: Methodology, Supervision.

Ummay Kulsum: Formatting & review.

Sources of funding for research presented in a scientific article or scientific article itself.

The research was funded by grants. This work is supported by the National Natural Science Foundation of Hebei province under grant No: E2015202292 and No: E2015202050, high-level talent support project in Hebei province under grant No: C2015005012, Key research and development program under grant No: 15272002 and No: 15275704.