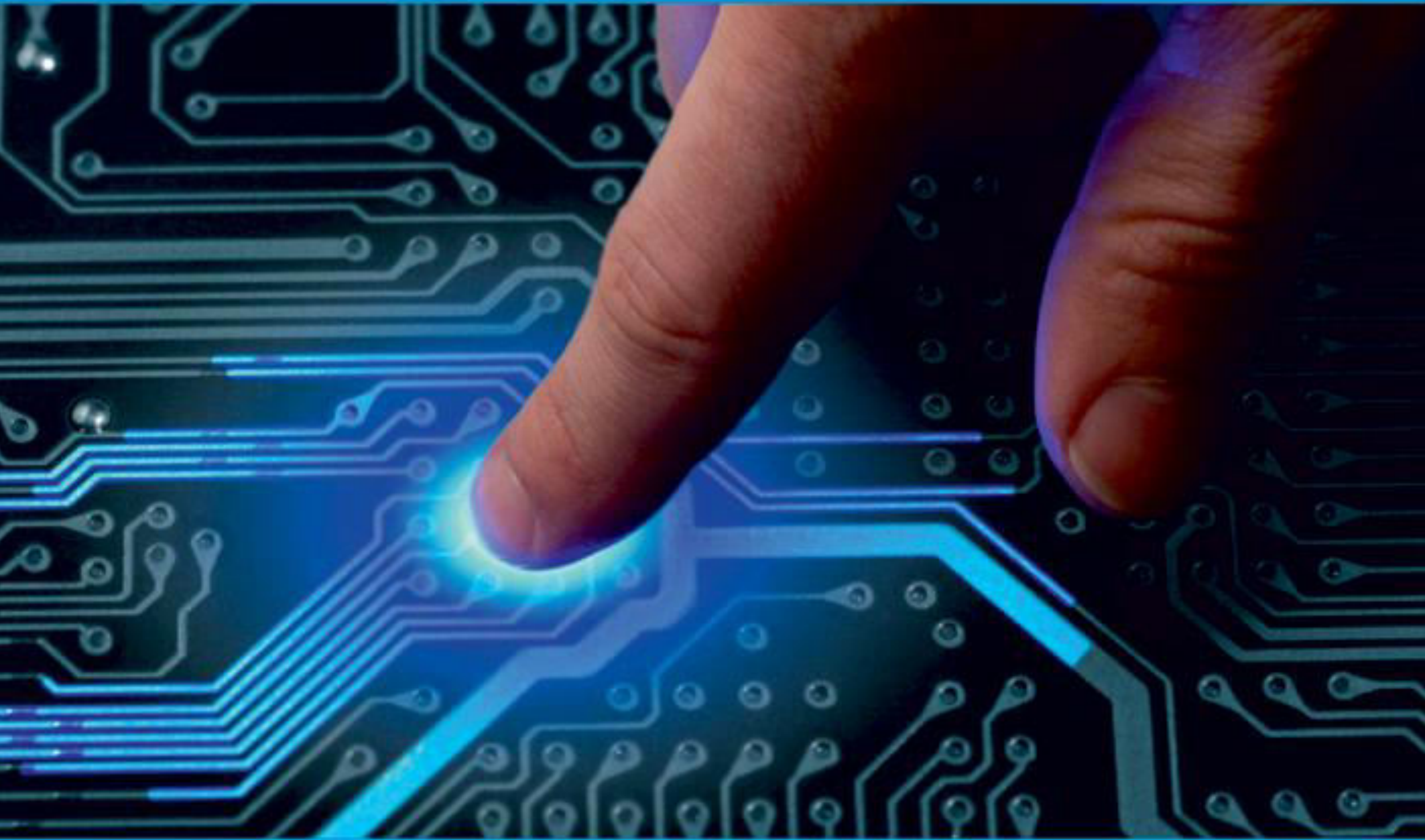
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 7, July 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

| DOI: 10.15680/IJIRCCE.2022.1007109 |

Detection and Classification of Tumors Using Deep Learning

Shagun Lahoti, Shilpa Mene, Shubham Pawar, Mayuri Nari

UG Student Department of Information Technology Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, India

Guide and Assistant Professor Department of Information Technology Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, India

UG Student Department of Information Technology Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, India

UG Student Department of Information Technology Engineering K. K. Wagh Institute of Engineering Education and Research Nashik, India

ABSTRACT: Brain tumors are uncontrolled mass growth of abnormal tissues in the brain which may affect the working of the body in certain manner. Thus, the detection of brain tumors in an early stage becomes a crucial task. There are many imaging techniques used to detect brain tumors. Magnetic Resonance Image is commonly used due to its superior image quality and the fact of relying on no ionizing radiation. Deep learning (DL) is a subfield of machine learning and has recently shown a remarkable performance, especially in classification and segmentation problems. Here, a Deep Learning Model based on convolutional neural networks using a 13-layer CNN architecture is proposed to classify the tumours into cancerous and non-cancerous and based on this further in- meningioma, glioma and pituitary tumours. The dataset used here will contain a total of 3265 images with some T1-contrast weighted images in order to detect the brain tumours in an efficient manner and further a total of 2764 images in order to classify the tumors in an efficient manner. The proposed network structure is expected to achieve a significant performance. The model proposed here will efficiently predict the brain tumours type and location from the available Magnetic Resonance Images.

KEYWORDS: Brain tumor, Magnetic Resonance Image, Deep Learning, Convolutional Neural Networks.

I. INTRODUCTION

The aggregation of abnormal cells in some tissues of the brain is referred to as the formation of a brain tumor [1]. Since the skull has a rigid boundary, the volume inside it remains unchanged and the existence of tumor in brain may cause damage to normal brain cells by causing inflammation, exerting pressure on some parts of the brain leading to increasing the pressure inside the skull [2]. The tumors developed can either be Cancerous (also known as Malignant) or Non-Cancerous (also called as Benign). As compared to the Benign tumors, Malignant tumors grow at a much rapid rate and spread over the areas of the brain and spine which makes them more life-threatening [1]. These tumors that are developed in the brain are divided into two categories based on their place of origin. The categories are: Primary tumors and Secondary or Metastatic tumors [3]. The primary tumors originate in the brain itself whereas the metastatic tumors originate in other parts of the body and then travel to the brain [1].

The most common type of tumors that arise in adults are Gliomas and originate due to the presence of glial cells in the brain. 30% of all brain tumors and 80% of all malignant brain tumors are Gliomas. Gliomas can be classified into four sub grades from type I to type IV. Grade I tumors have a much similar texture of glial cells and are benign. Grade II tumors have a slightly different texture. Grade III tumors have appearance of abnormal tissues and are malignant whereas Grade IV tumors are considered to be the most severe stage of gliomas and the abnormality of these tissues can even be seen with naked eye [4].

Brain cancer accounts for less than 2% of human cancer according to the World Cancer Report published by the World Health Organisation (WHO). However, it produces severe morbidity and complications [11].

Early detection and classification of brain tumors turn into a vital task in case assessment and accordingly help in selecting the most convenient treatment method to save patients life [4]. The brain tumors are generally detected with the help of MRI scans followed by a biopsy to predict whether the tumor is benign or malignant. This traditional method of detecting the tumor may require a lot of time, it may take up to a month to give some concrete results.

In the recent times, some new Computer-Aided techniques are being used to perform the detection of brain tumors using these methods helps overcome the drawbacks of the traditional biopsy method. Unlike biopsy, detection of tumors via the computer aided techniques using the MR images is a non-invasive, automatic and an efficient system for brain tumor detection and its further classification which help the doctors to interpret the MR images and furthermore support the predictions made by the specialists in the early stages.

In this study, a Computer-Aided Diagnosis system is proposed for the non-invasive detection and classification of brain tumors without any human interventions. Machine Learning (ML) is the study of algorithms and statistical models that can be used to perform a specific task without using outright instructions, relying on patterns instead of that [5]. ML algorithms have been widely emerged in the medical imaging field as a part of artificial intelligence [6]. The types of Machine Learning algorithms can broadly be categorized into Supervised Learning, Unsupervised Learning, Semi-supervised Learning and Reinforcement. The Supervised Learning algorithm is consistent of an output variable which is predicted from a given set of predictors. Using these set of predictors, a function is generated to map the input to the output which continues until the trained model achieves the required accuracy on the training and validation set. Some algorithms that come under supervised learning are: Nearest Neighbour, Naïve Bayes, Decision Tress, Linear Regression, Support Vector Machines and Neural Networks. On the other hand, Unsupervised Learning algorithm uses different techniques on the input data to derive rules, detect patterns and to cluster the data in different groups, which is widely used for segmentation of data into different groups with some specific intervention. Some algorithms that come under Unsupervised Learning are: Apriori Algorithm, K-means Clustering.

Neural Networks are an integral part of programming which were originally inspired by the neurons and the brain's processing in the human body. The neural networks in general have a number of neurons, each having a number of inputs but only one output [9]. These are generally used for image visualization and to design the minimal pre-processing requirements [10].

Deep Learning (DL) is a subdivision of ML that is based on learning data representations and hierarchical feature learning [4]. DL is derived from the conventional neural network but considerably outperforms its predecessors [8]. Deep neural networks contain multiple non-linear hidden layers and this makes them very expressive models that can learn very complicated relationships between their inputs and outputs [7].

In this paper, a CNN architecture is proposed for the detection and classification of brain tumors using MR images. The paper is organized in the following manner: Section 2 contains work previously done related to the detection and classification of tumors using different algorithms. Section 3 contains a detailed view of the methodology used. Section 4 is dedicated to all the experiments performed and the obtained results of the respective experiments. Section 5 consists of a conclusion.

II. RELATED WORK

The proposed CNN architecture in [1] is a combination of fully connected, pooling and convolutional layers. A method based on CNN is proposed to classify three grades of Gliomas with MR images. Selecting an appropriate deep neural network architecture for a specific purpose, consists of a challenging procedure which is usually done by trial and error or employing a common architecture.

In [6] the authors have carried a survey on deep learning for image classification, object detection, segmentation, registration, and other tasks. The default CNN architecture can easily accommodate multiple sources of information or representations of the input, in the form of channels presented to the input layer. It is clear that applying deep learning algorithms to medical image analysis presents several unique challenges. The lack of large training data sets is often mentioned as an obstacle.

The concept of dropout is discussed in [9] by the authors. Dropout is a technique used to prevent overfitting and provide a way of approximately combining exponentially many different neural network architectures efficiently. In this paper, they have trained dropout neural networks for classification problems on data sets in different domains and they found that dropout improved generalization performance on all data sets Dropout was applied to all the layers of the network (going from input to convolutional layers to fully connected layers). Max-norm regularization was used for weights in both convolutional and fully connected layers. The idea of dropout is not limited to feed-forward neural nets. It can be more generally applied to graphical models such as Boltzmann Machines.

III. PROPOSED ALGORITHM

FIGURE 1 shows the block diagram of the proposed method, wherein the images from the input stage further are sent for data preparation where the images and their labels are loaded in the system. These images are further passed to the pre-processing step where some operations like image grayscaling, image thresholding, image dilation-thresholding and image contouring are performed. Just before the images are passed to multiple architectures for pre-processing, they split into training and testing sets. After setting the hyper-parameters and optimizers, the best performing architecture is selected. Finally, after the performance evaluation, a report is generated containing the required specifications.

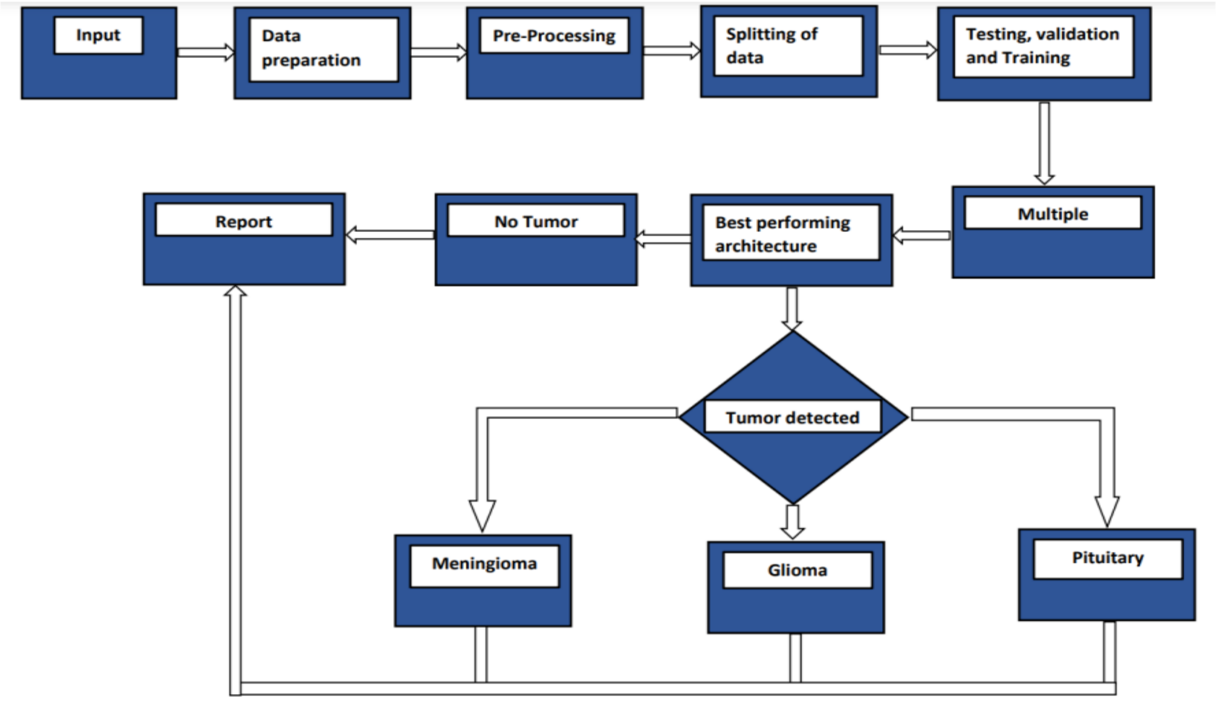
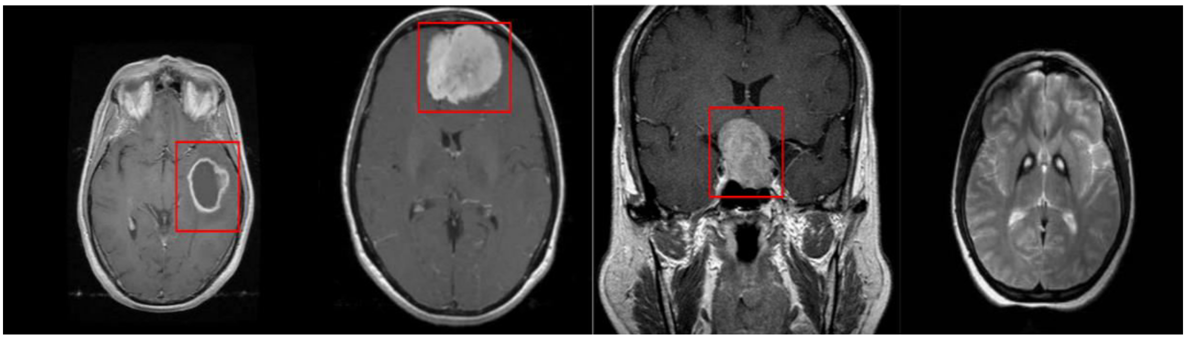


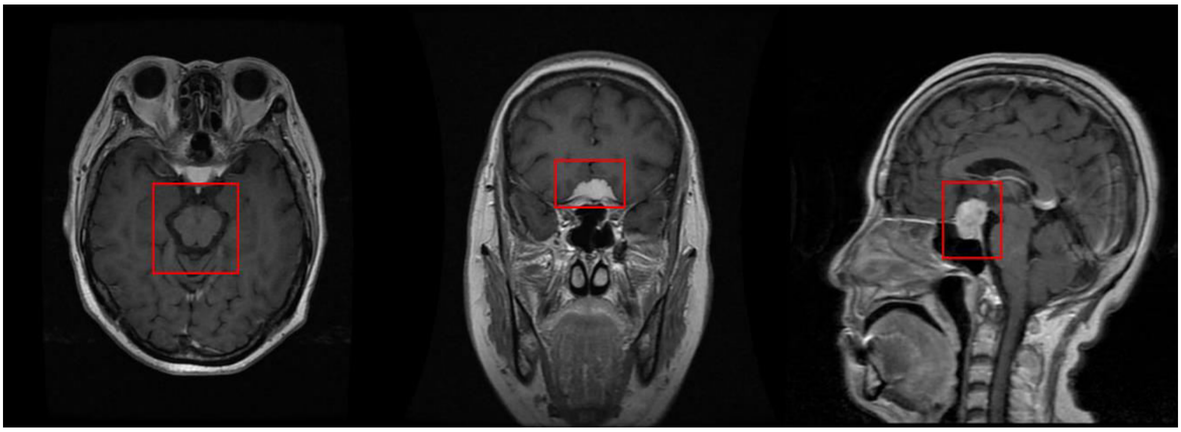
FIGURE 1 Block Diagram of the proposed Architecture

A. *DATASET*

In this work we have used one freely available dataset. The dataset was acquired from an open-source platform where it was published 3 years ago by - Navoneel Chakrabarty and Swati Kanchan. The dataset consists a total of 3264 MR images acquired from different patients. The images are further subdivided into four types which are Glioma Tumor, Meningioma Tumor, Pituitary Tumor and No Tumor. These tumors in the images differ in their size, shape and location depending on their tumor type. FIGURE 3(a) shows images of all these categories of tumors and an image containing no tumor. The images in the database have different views which are – Axial view, Coronal view, Sagittal view which is shown in FIGURE 3(b).



(a)



(b)

FIGURE 2. **(a)**Different three axial brain tumor types as follows: Glioma Tumor, Meningioma Tumor, Pituitary Tumor and No tumor.
(b) Pituitary tumor is demonstrated in three different views – Axial, Coronal and Sagittal views.
The tumors are localized in the red rectangles

TABLE 1 – The table below shows the number of lesion samples of the images either containing a tumor or no tumor present in dataset 1

Tumor Category	Number of Samples
No Tumor	502
Tumor	2765
Total	3267

TABLE 2 – The table below shows the number of lesion samples of each tumor type (glioma, meningioma and pituitary tumor) present in dataset 2

Tumor Category	Number of Samples
Glioma	926
Meningioma	937
Pituitary	901
Total	2764

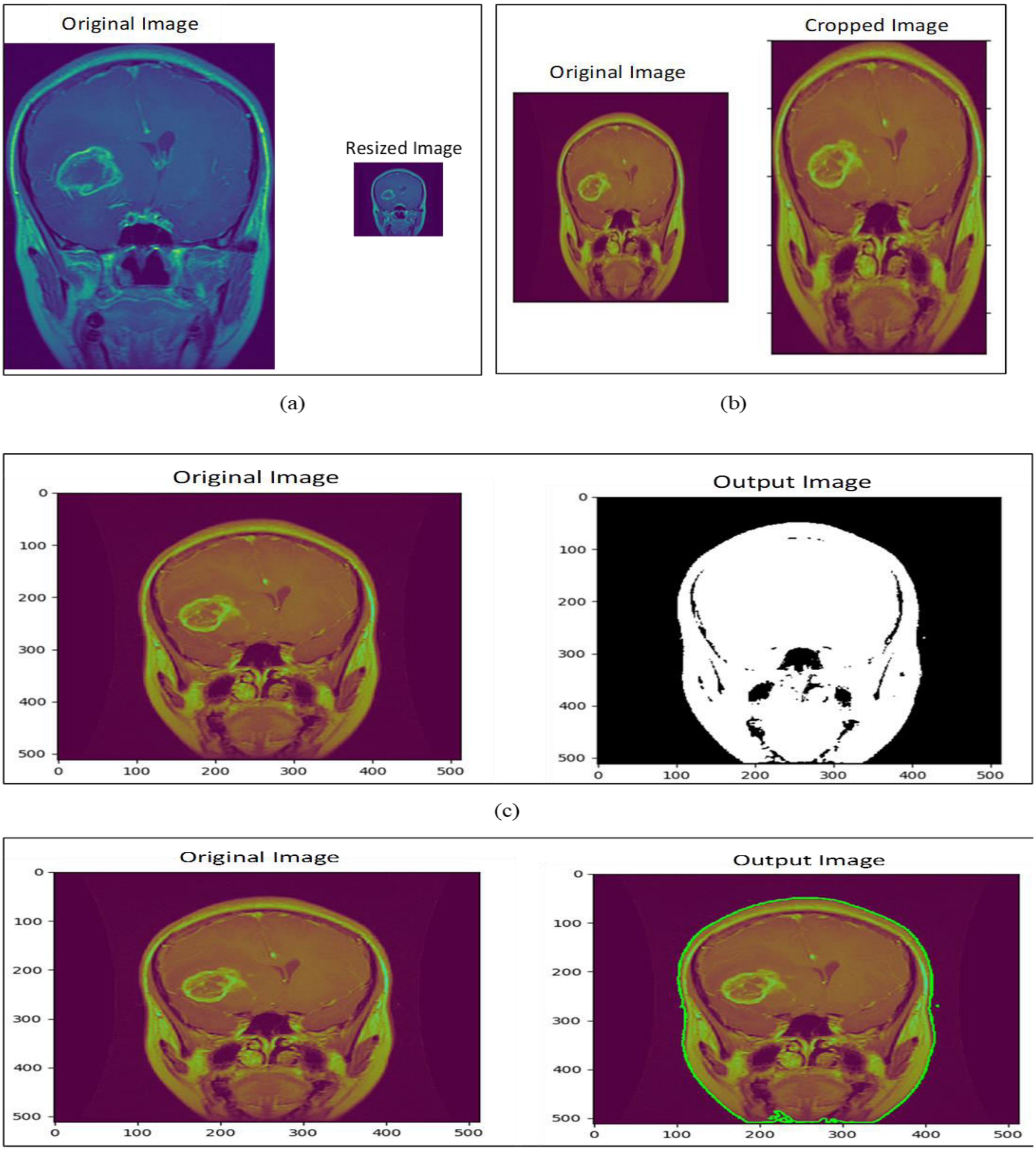
B. *PRE-PROCESSING STAGE*

Before providing the images of the dataset directly as the input to the system, some pre-processing steps are performed over these images. The images are first converted from 512 × 512 × 1 pixels to 150 × 150 × 3 pixels. This increases the computation speed as images with higher dimensionality take a longer time for processing. The images then undergo cropping, grayscaling and thresholding, resizing and image contouring as shown in FIGURE 3. The dataset then is split into three different sets – Training set and Test set. After performing all the aforementioned operations over the images of the dataset, these images are sent as the input to the system.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165

|| Volume 10, Issue 7, July 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1007109 |



(a)

(b)

(c)

(d)

FIGURE 3. (a) Image resizing, **(b)** Image cropping, **(c)** Image grayscaling and thresholding, **(d)** Image contouring

C. *PROPOSED ARCHITECTURE*

FIGURE 4 shows the proposed CNN architecture. It is a 13-layer architecture with the first layer as the input layer to which the images are passed from the preprocessing stage. Both, detection and classification stages use this architecture to detect and classify the brain tumor via the provided input images. The images are sent to the input layer and then they travel through the convolution layers and their respective activation functions, a normalization layer, max-pooling layers, dropout layers are used to prevent overfitting of the model followed by a fully connected layer, softmax layer and a classification layer to get the classified output i.e. the classified tumor type.

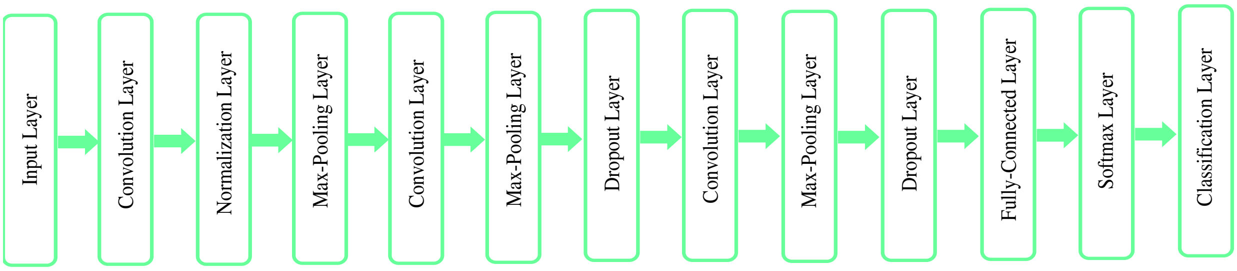
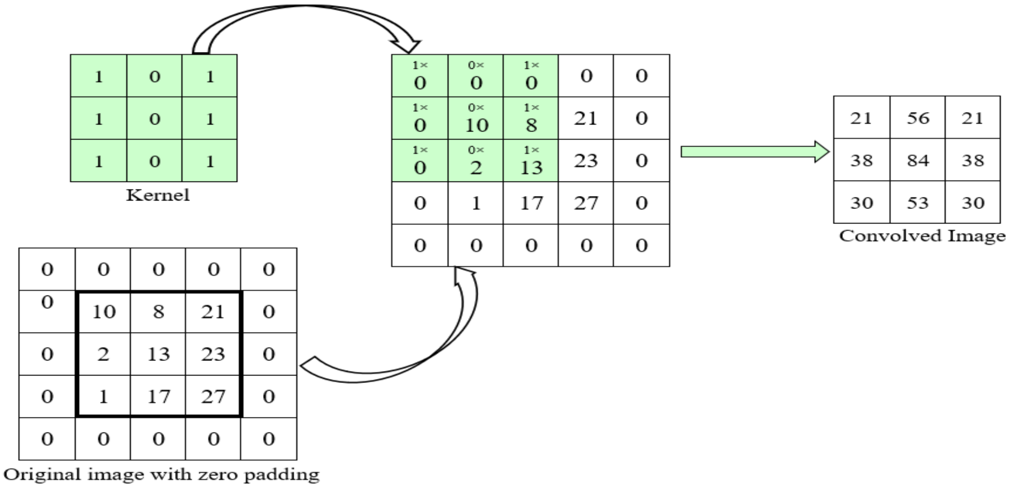


FIGURE 4 Proposed CNN Architecture

Description of each layer is as follows: The input layer contains the images that are passed on through the previous stage (from pre-processing stage in case of the detection step and from detection stage in case of classification step). It's basic function is to check if the images that are passed to this stage, have the required dimensions [4].

The input layer is followed by a convolution layer and their respective activation functions. The convolution layers help in creating a feature map by passing a matrix which is also called as kernel (K) over the input matrix of size (M × N) [12]. The convolution layers are all 3 × 3 convolution layers with stride (S) 1 and the same padding (P) [13]. These kernels are also used as feature detectors. These in the early stages identify simple features (lines and edges) and complex features during the advanced stages [14]. In the proposed architecture the filter size was chosen as 3 × 3 as it does not act to be very extreme. When the filter size 1 × 1 is used, the entire input image and its features cannot be taken into consideration. Whereas, on the other hand when a 5 × 5 or a higher kernel size is used, there are higher chances that some features may not be extracted at all. The system uses same padding in order to maintain homogeneity amongst the available input images.



1	0	1
1	0	1
1	0	1

1× 0	0× 0	1× 0	0	0
1× 0	0× 10	1× 8	21	0
1× 0	0× 2	1× 13	23	0
0	1	17	27	0
0	0	0	0	0

21	56	21
38	84	38
30	53	30

0	0	0	0	0
0	10	8	21	0
0	2	13	23	0
0	1	17	27	0
0	0	0	0	0

FIGURE 5 Example of a convolutional layer (input: 3 × 3, zero padding: 1, kernel size: 3 × 3, stride: 1, output: 3 × 3)

FIGURE 5 depicts an example of a convolutional layer, where a kernel of size (3 × 3) is applied over the original input image. The output is a 3 × 3 image which is produced after kernel sliding and dot product.

ReLU, a non-saturated activation function follows the convolutional layers. It majorly contributes in decreasing the time required for training as compared to the other available activation functions [15], [16]. ReLU is used as the activation function here, because unlike other activation functions it preserves information about relative intensities as the information passes through the architecture layer [15]. As compared to linear models, the use of an activation function makes the model better in terms of training time and computations. FIGURE 6 represents ReLU graphically. ReLU projects the negative values to zero and the function is defined as:

$$relu(x) = \begin{cases} xifx \geq 0 \\ 0\ ifx < 0 \end{cases} \tag{1}$$

| DOI: 10.15680/IJIRCCE.2022.1007109 |

Leaky ReLU in many cases performs better than ReLU. It allows some small non-zero gradient even when the function is passive. It is defined as:

$$leaky_{relu(x)} = \begin{cases} xifx \geq 0 \\ \alpha xifx < 0 \end{cases} \quad (2)$$

Here, α generally has a value of 0.3. Exponential Linear Unit (ELU) has recently made some moves as it gives higher classification accuracy and takes lesser training time [17]. It is defined as:

$$elu(x) = \begin{cases} xifx \geq 0 \\ a(e^{x} - 1) \quad ifx < 0 \end{cases} \quad (3)$$

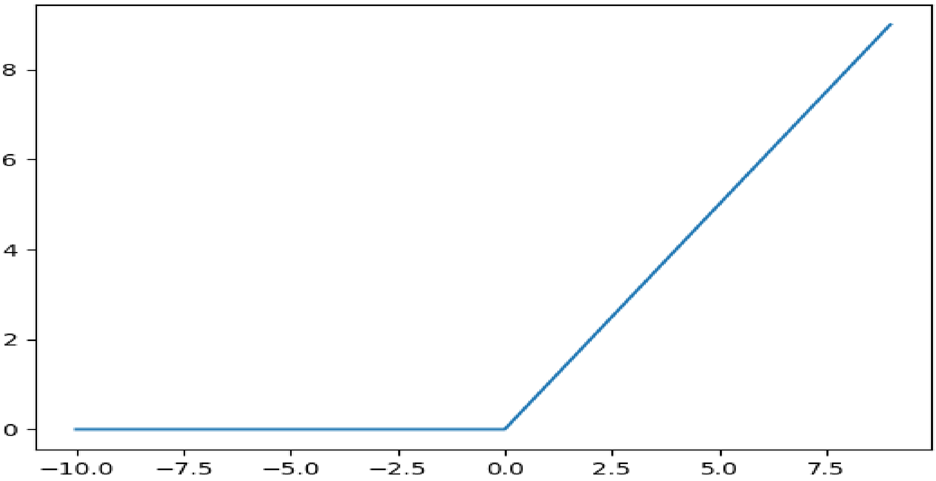


FIGURE 6 ReLU Activation Function

The convolutional layers and their activation functions are followed by a Normalization Layer. Using the proposed architecture, various normalization techniques were explored. It was observed that the best results were obtained when batch normalization was used. Normalization technique is generally used in order to adjust the related activation functions and scale the input [4]. It is also used in order to generalize the model in terms of the input image. Layer Normalization and l_p normalization were also taken into consideration but it was found that batch normalization provided the dataset with a lot more stability as compared to the two former techniques [23]. Another reason that contributes majorly to the use of Batch Normalization is its compatibility with the ReLU activation function.

Max-Pooling Layer follows the Normalization layer in the architecture shown above. Max-Pooling layers are generally present in neural networks to reduce the resolution of feature maps and eventually achieve spatial invariance [18]. Max-pooling layer not only contributes in achieveing spatial variance but also helps in down-sampling the the image by K_x and K_y along each direction. It also helps to attain a faster convergence rate which results in ameliorates generalized performance of the network [19]. The max pooling function can be given as:

$$a_j = \tanh(\beta \sum_{N \times N} a_i^{n \times n} u(n, n)) \quad (4)$$

Where, $u(x, y)$, a window function is applied to input patch which computes the maximum in a give set of values [18]. FIGURE 7 depicts an example of a Max-Pooling layer.

In order to avoid overfitting, the max-pooling layer is followed by a dropout layer. Dropping out of units is generally referred to as “dropout” [7]. Dropout technique not only helps in generalizing the model but also results in increasing the training rate. For the proposed network, 25% dropout probability was found to be most suitable. FIGURE 8 depicts and example of a dropout layer.

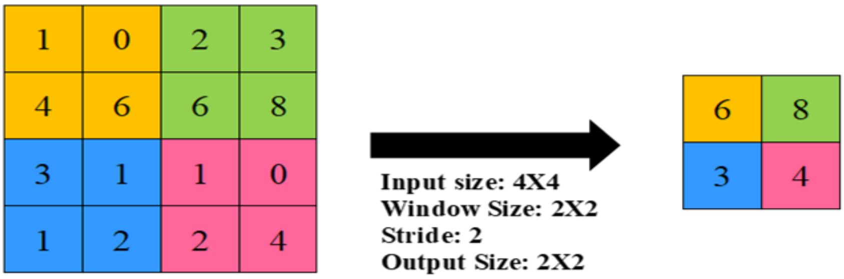


FIGURE 7 An example of max-pooling layer

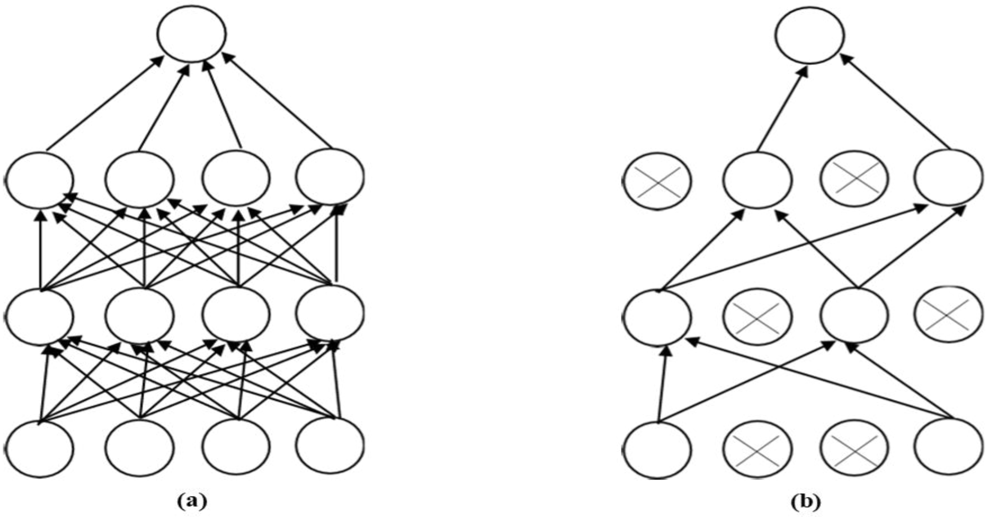


FIGURE 8 (a) Standard Neural Net, (b) After applying dropout

Finally, a total of 3 advanced layers were used: 1 Fully Connected layer, 1 Softmax layer and 1 Classification layer. The fully-connected layer connects all the neurons from the previous layers to all the neurons of the next one. It will produce three classes as its output. It is followed by a softmax layer which is also known as normalized exponential function [5]. The main function of softmax layer is to get all the predicted values in the range of 0-1 with the total sum of these values being equal to 1 or 100% [4]. FIGURE 9 shows an example of a softmax layer. The output of this function can be calculated by the equation given below:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (5)$$

Where, ‘i’ is a class. The probability of this class can be calculated over ‘j’ different classes as a function of σ (z) and their summations adding up to 1.

Loss function is another important aspect that needs to be taken into consideration while designing any neural network. Lower the value of the loss function higher will be the accuracy attained by the network. The proposed architecture uses categorical cross-entropy to calculate the loss. It is an efficient way of calculating loss over models with unevenly distributed datasets. This function is defined for two distributions (p and q) over a discrete variable y and is given by:

$$H(p,q) = -\sum_x p(y)\ln(q(y)) \qquad (6)$$

Where, $q(y)$is the estimate for true distribution $p(y)$.

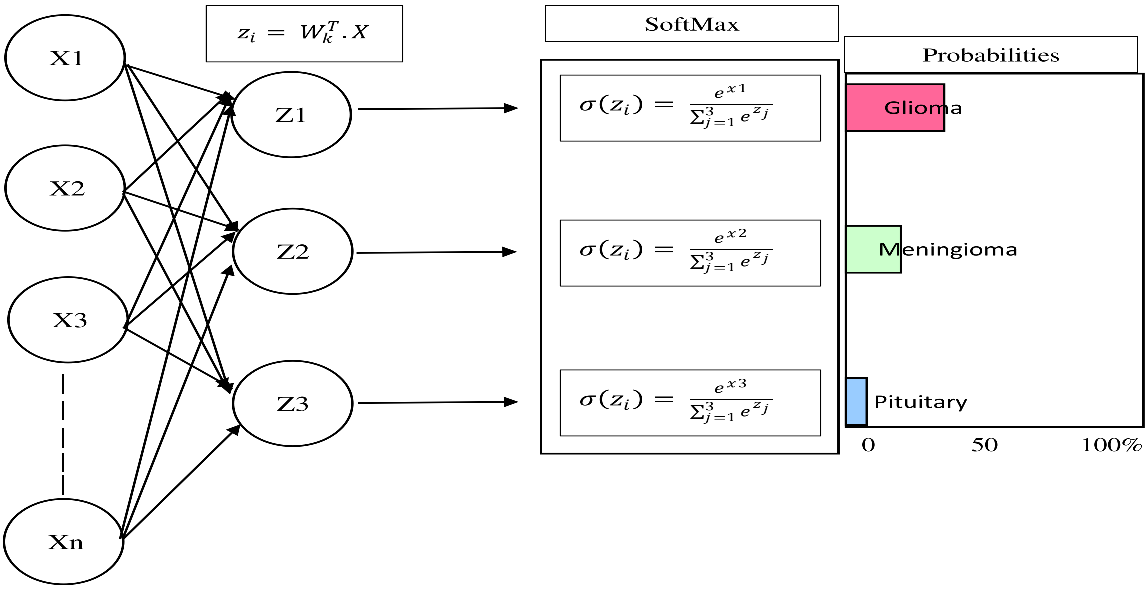

X1
X2
X3
Xn
$z_i = W_k^T.X$
Z1
Z2
Z3
SoftMax
$\sigma(z_i) = \frac{e^{x1}}{\sum_{j=1}^{3} e^{z_j}}$
$\sigma(z_i) = \frac{e^{x2}}{\sum_{j=1}^{3} e^{z_j}}$
$\sigma(z_i) = \frac{e^{x3}}{\sum_{j=1}^{3} e^{z_j}}$
Probabilities
Glioma
Meningioma
Pituitary
0
50
100%

FIGURE 9 Example of a Soft-max Layer

D. *REGULARIZATION TECHNIQUES AND OPTIMIZATION ALGORITHM*

The term Regularization refers to fitting a solving solution well during training while preventing system overfitting [4]. Some techniques have been used during the training and preprocessing phases to avoid overfitting. Firstly, data augmentation is used. Image orientation and geometric distortion is performed on the images in order to avoid overfitting. Then a dropout layer has been used to stochastically remove the invisible weights. Finally, to monitor the validation and testing performance of the system, early stop technique has been used. It is used to monitor the performance of the system and it stops before completion of all the epochs when either the system attains stability or before the system starts to overfit [20].

In order to minimize the loss function and update the network parameters optimization technique is used. Stochastic gradient descent (SGD) is a widely used optimizer in deep neural networks. Adaptive moment estimation (Adam) is a recently presented stochastic optimization technique. It works better than the already available customary optimization algorithms [21]. Generally, the learning rate remains constant for the SGD algorithms. However, Adam as an optimizer works differently. It first computes the moment (the mean) and then the second moment (the uncentered variance) of the gradients [1]. Other optimizers like Adagrad, Nadam, Adamax were also examined during the training process. Unlike these optimizers, Adam performed the best giving the best accuracy and the least loss.

IV. Simulation Results

A. *Evaluation of the Detection Stage*

FIGURE 10 shows the progress of the system with accuracy as its parameter. The figure shows that the accuracy for the training set is almost 99% right after the 10th epoch and the accuracy for the validation set ranges from 94% to 99%.

FIGURE 11 shows that the loss for the training set drops sharply and becomes constant after the 8th epoch. On the other hand, for the validation set, the loss graph has some fluctuations throughout 100 epochs as a mini-batch of small size 32 images is used.

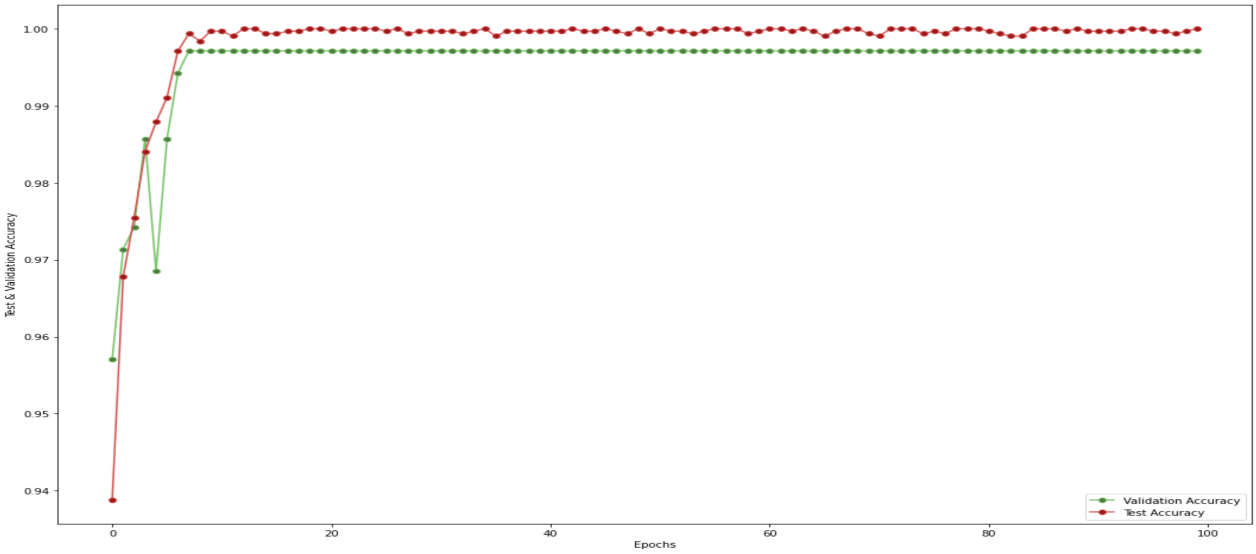


FIGURE 10 Testing and Validation Accuracy over all the iterations of study

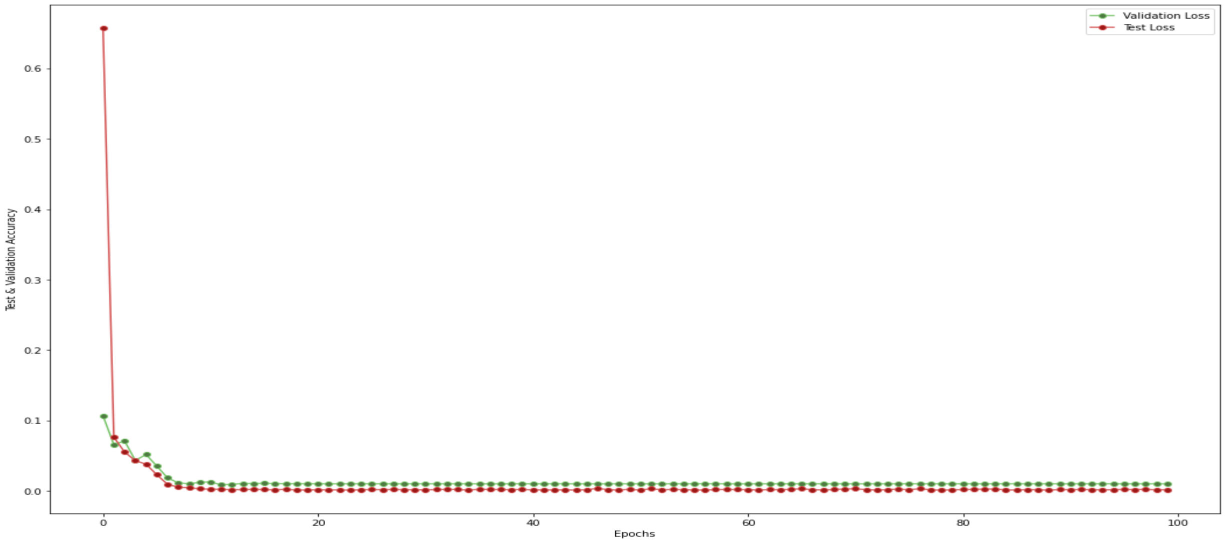


FIGURE 11 Testing and Validation loss over all the iterations of study

B. *Evaluation of the Classification Stage*

FIGURE 12 shows the progress of the system with accuracy as its parameter. The figure shows that the accuracy for the training set is almost 99% right after the 10th epoch and the accuracy for the validation set ranges from 58% to 95%.

FIGURE 13 shows that the loss for the training set drops sharply and becomes constant after the 15th epoch. On the other hand, for the validation set, the loss graph has some fluctuations throughout 100 epochs as a mini-batch of small size 32 images is used.

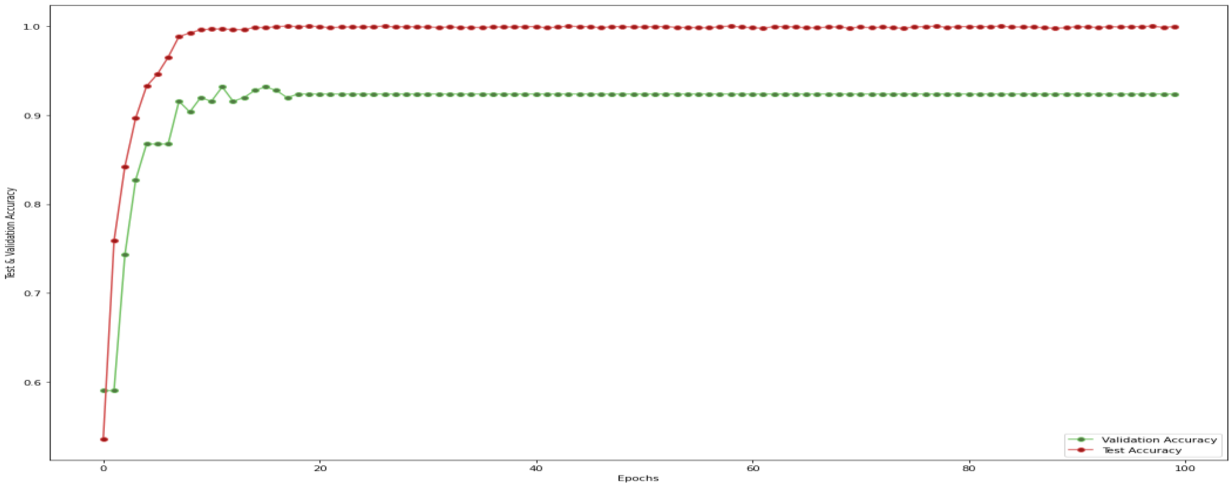


FIGURE 12 Testing and Validation accuracy over all the iterations of study

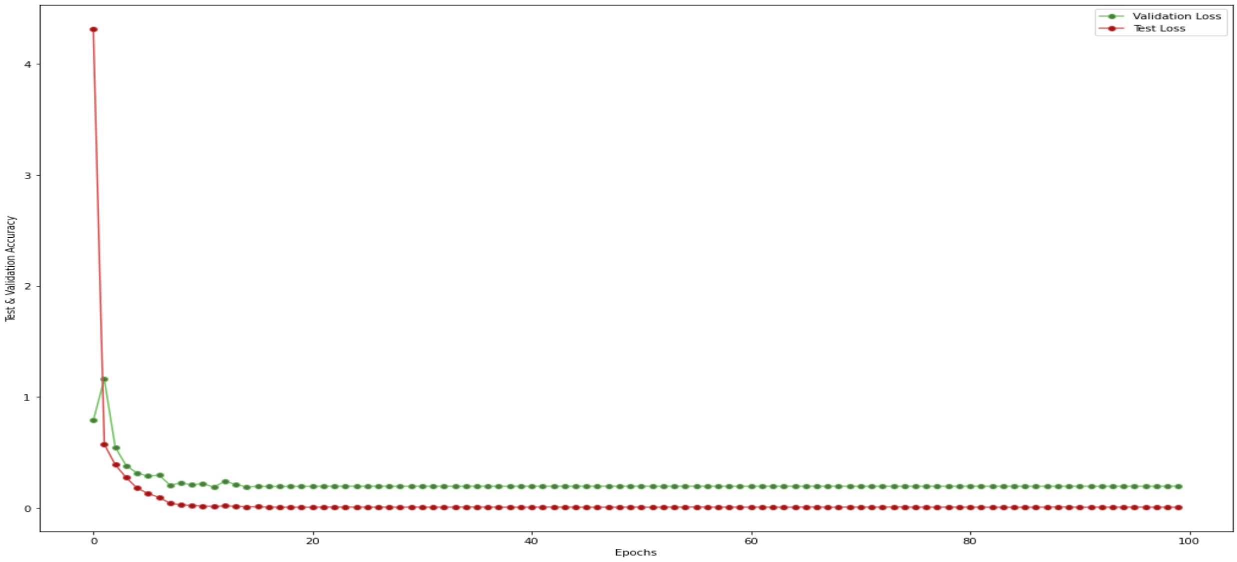


FIGURE 13 Testing and Validation loss over all the iterations of study

C. *CONFUSION MATRIX*

FIGURE 14 shows the confusion matrix that was obtained while performing the detection of the tumors and FIGURE 15 shows the confusion matrix that was obtained while performing the classification of the tumors. This gives a clear insight of the system's performance. The X-axis represents the predicted label i.e. the output of the system whereas the Y-axis represents true labels i.e. the actual labels. Precision, Recall, Specificity, Accuracy and F1 Score have been calculated with the help of the formulae listed below:

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$Specificity = \frac{TP}{(TN + FN)}$$

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

$$F1 - Score = \frac{2\,(Recall\ \times Precision)}{(Recall + Precision)}$$

Where,

True Positive(TP) is the number of positive predicted cases and they are actually positive.
True Negative(TN) is the number of negative predicted cases and they are actually negative.
False Positive(FP) or Type one error is the number of positive predicted cases when they are actually negative.
False Negative(FN) or Type two error is the number of negative predicted cases when they are actually positive.

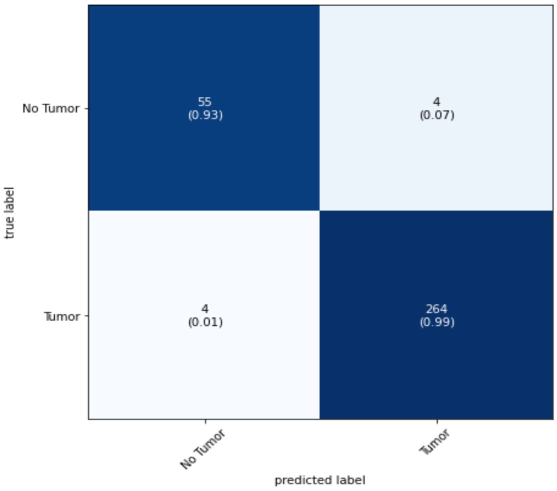


FIGURE 14 Confusion Matrix for Detection Stage

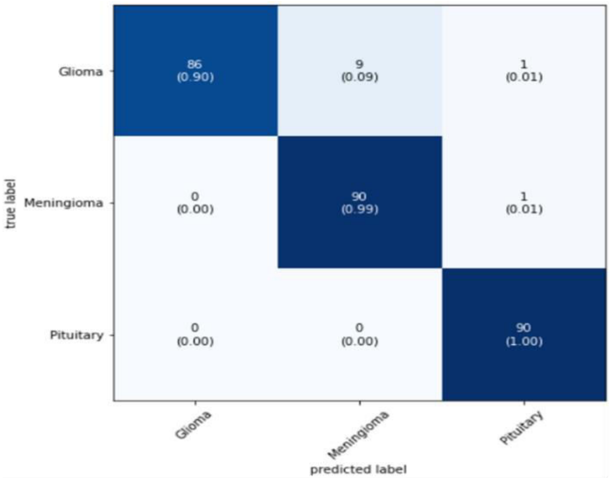


FIGURE 15 Confusion Matrix for the Classification Stage

TABLE 3 Accuracy metrics in terms of TP, TN, FP, FN, precision, recall, accuracy, and f1-score for the detection stage.

Metrices / Tumor Type	TP	FP	TN	FN	Weighted Precision	Weighted Recall	Weighted Accuracy	Weighted F1-Score
Tumor	55	4	264	4	0.93	0.93	97.55%	0.93
No Tumor	264	4	55	4	0.99	0.99	97.55%	0.99

TABLE 4 Accuracy metrics in terms of TP, TN, FP, FN, precision, recall, accuracy, and f1-score for the classification stage.

Metrices / Tumor Type	TP	FP	TN	FN	Weighted Precision	Weighted Recall	Weighted Accuracy	Weighted F1-Score
Glioma	86	10	181	0	0.89	**1.00**	96.3%	0.94
Meningioma	90	1	177	9	0.97	0.90	96.3%	0.94
Pituitary	90	0	185	2	**1.00**	0.97	**99.2%**	**0.98**

D. *EMPIRICAL ARCHITECTURE AND HYPER PARAMETERS*

This part consists of an overview of the different architecture parameters involved during the evolution and selection of the final model.

TABLE 5 shows the different parameters that were tested during the model building process before reaching the final model. The model that is selected gives the best performance.

TABLE 5 Different architectures and hyper-parameters tested and tried before reaching the final model.

Factor(s)	Values
Number of drop out layers	1, 2, 3
Maximum epochs	15, 20, 30, 40, 45, 50
Number of convolutional kernels	32, 64, 128, 256, 512
Kernel Size	3, 5, 7
Pooling Layer window size	2, 3, 4
Optimizers	SGD, Adam, Adamax
Mini-batch size	8, 16, 32, 64, 128
Dropout Rate	0.1, 0.15, 0.2, 0.25
Initial learning rate	0.001, 0.0001

E. *TOOLS AND TIME CONSUMPTION*

The training of the proposed architecture is done on Intel i7-10750H CPU (2.6 GHz), NVIDIA GeForce GTX 1650 (4GB) GPU, 16GB RAM, Visual Studio Code (2022) and Python. The training time for the detection stage was 60 minutes for 2870 images and for the classification stage it was observed to be 34.8 minutes for 2475 images in the study. The average test execution time for the detection stage was [60 seconds per step] and for the classification stage was [58 seconds per step].

V. Conclusion and Future Work

In this work, we have presented a computer-aided diagnosis system for the detection and classification of brain tumor via MR images into the following three types: Meningioma, Glioma and Pituitary tumors using a CNN architecture. The network proposed is built from 13 layers. These layers start from the input layer which holds the augmented and preprocessed images through the convolution layers and their respective activation functions (3 Convolution layers, 1 Normalization layer, 3 Max-Pooling layers and 2 Dropout layers) followed by a Fully Connected Layer. Finally, a softmax layer and a classification layer is used to get a classified output from the system. Data augmentation helped show better results even when the dataset used was relatively small in size. The proposed structure has achieved the highest accuracy of 97.55% for the detection stage and 96.3% for the classification stage.

References

[1] A. K. Anaraki, M. Ayati, and F. Kazemi, ‘‘Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms,’’ Biocybernetics Biomed. Eng., vol. 39, no. 1, pp. 63–74, Jan./Mar. 2019. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Salabha Varghese “Brain Tumor Segmentation by FCM and Enhancement by ANN, using GLCM Based Feature Extraction ” International Journal of Science and Research (IJSR), Volume 3 Issue 11, November 2014.

[3] Damandeep Kaur, Surender Singh , “Detection of Brain Tumor using Image Processing Techniques”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8, Issue-5S3, July 2019

[4] HOSSAM H. SULTAN , NANCY M. SALEM , AND WALID AL-ATABANY, “Multi-Classification of Brain Tumor Images Using Deep Neural Network” IEEE Access, Vol. 7, date of publication May 27, 2019

[5] C. Bishop, Pattern Recognition and Machine Learning. Berlin, Germany: Springer-Verlag, 2006.

[6] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, ‘‘A survey on deep learning in medical image analysis,’’ Med. Image Anal., vol. 42, pp. 60–88, Dec. 2017

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ‘‘Dropout: A simple way to prevent neural networks from overfitting,’’ J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[8] V. Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie & Laith Farhan “Review of deep learning: concepts, CNN

architectures, challenges, applications, future directions" Alzubaidi et al. J Big Data (2021) 8:53 https://doi.org/10.1186/s40537-021-00444-8

[9] Ezra Philip Darius Davis, Miyabi Ida Gaskell, Ryan Baylor Killea "Convolutional Neural Network Visualization for fMRI Brain Disease Classification Tasks".

[10] Y. LeCun. (2015). Lenet-5, Convolutional Neural Networks. Accessed: May 2019. [Online]. Available: http://yann.lecun.com/exdb/lenet

[11] B. W. Stewart and C. P. Wild, World Cancer Report 2014. Lyon, France: IARC, 2014.

[12] Srikanth Tammina, Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images, International Journal of Scientific and Research Publications, Volume 9, Issue 10, October 2019 143 ISSN 2250-3153

[13] Qing Guan1,2, Yunjun Wang1,2, Bo Ping2,4, Duanshu Li1,2, Jiajun Du3, Yu Qin3, Hongtao Lu3, Xiaochun Wan2,4,*, Jun Xiang1,2, Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study, Journal of Cancer 2019, Vol. 10

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998

[15] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In Proc. 27th International Conference on Machine Learning, 2010.

[16] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' 2015, arXiv:1502.03167. [Online]. Available: https://arxiv.org/abs/1502.03167

[17] Clevert D-A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (elus); 2015, arXiv preprint arXiv:1511.07289.

[18] D. Scherer, A. Müller, and S. Behnke, ''Evaluation of pooling operations in convolutional architectures for object recognition,'' in Artificial Neural Networks. Berlin, Germany: Springer, 2010, pp. 92–101.

[19] J. Nagi, F. Ducatelle, G. A. D. Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, ''Max-pooling convolutional neural networks for vision-based hand gesture recognition,'' in Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA), Nov. 2011, pp. 342–347.

[20] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT press, 2016.

[21] Kingma D, Ba J. Adam: a method for stochastic optimization; 2014, arXiv preprint arXiv:1412.6980.

[22] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 2011;12(July):2121–59

[23] Zhenwei Dai and Reinhard Heckel Channel Normalization in Convolutional Neural Networks avoids Vanishing Gradients

BIOGRAPHY

Shilpa Mene is an Assistant Professor in the Department of Information Technology, K. K. Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University. She received her Master of Engineering degree in Computer Science in 2013 from SPPU, Pune, India. Her interests are Database Management System, System Programming, Information Retrieval Systems, Natural Language Processing, Machine Learning, Computer Vision etc.

Shagun Lahoti completed her Bachelor of Engineering in Information Technology at K. K. Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University in 2022. She will further complete her Master of Science in Computer Engineering from Northwestern University.

Shubham Pawar completed his Bachelor of Engineering in Information Technology at K. K. Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University in 2022. He further aspires to join the Armed Forces.

Mayuri Nari completed her Bachelor of Engineering in Information Technology at K. K. Wagh Institute of Engineering Education and Research, Savitribai Phule Pune University in 2022. She currently works with an IT company.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

☎ 9940 572 462 6381 907 438 ✉ ijircce@gmail.com

Scan to save the contact details

www.ijircce.com