Proceedings of the Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021)
IEEE Xplore Part Number: CFP21K74-ART; ISBN: 978-0-7381-1327-2

Convolutional Neural Network based Brain Tumor Detection

Shraddha S. More[1], Mansi Ashok Mange[2], Mitheel Sandip Sankhe[3], Shwethali Santosh Sahu[4]
[1,2,3,4] Department of Information Technology
[1,2,3,4] St. John College of Engineering and Management, Palghar, India
[1]shraddham@sjcem.edu.in , [2]mansimange1234@gmail.com , [3]mitheel3@gmail.com , [4]shwetalisahu20@gmail.com

2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) | 978-1-6654-1272-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICCS51141.2021.9432164

Abstract—**One of the most important and demanding tasks in the field of medical image processing is Brain Tumor Detection, as inaccurate prediction and diagnosis can result from human-assisted manual classification. One of the functions of Artificial Intelligence is Deep Learning which mimics the work of the person's brain. It is used to detect artifacts in the processing of data, recognize the voice, translate languages, and make decisions. Without human administration, it may understand, demonstrating from data that is both unorganized and unlabelled. A Convolutional Neural Network is a form of deep neural network which is used most commonly in optical representation analysis in deep learning. The current situation provides systems that detect brain tumors, but only use small datasets and image processing techniques. The proposed system majorly consists of 3 parts namely: Augmentation, Image pre-processing and applying Convolutional Neural Network (CNN). Our approach is to propose a system in which we will use a large dataset and deep learning algorithm. Results demonstrate that the CNN has 87.42% training accuracy with low difficulty, which sets it apart from all other state-of-the-art approaches.**

Keywords—Augmentation, Brain Tumor detection; Convolutional Neural Network (CNN); Deep Learning; Image Pre-processing; Magnetic Resonance Image

I. INTRODUCTION

A tumor inside the brain is an irregular development of a tissue that can cause problems with brain function. Doctors classify tumors based on where the tumor cells arose from and whether or not they are cancerous. Benign tumors may grow and influence the remaining healthy brain tissue. Malignant tumors are generally called brain cancer that grows outside the brain [1]. In the United States currently, an approximate 700,000 people have a primary brain tumor, with an estimated 85,000 predicted to be diagnosed by 2021. Brain tumors can be lethal, have a significant impact on people's lives, and fully change the lives of both the patient and their families. They effect male, female, and infants of all cultures and nationalities, without regard for race or ethnicity. [2]. In brain tumor diagnosis and treatment, imaging technology plays a key role. Imaging of tumors can be done in many different forms, including CT scanning, MRI, etc. A Brain MR image contains an enormous volume of information about the spatial brain structure and is suitable for medical diagnosis. Tumors of the brain are considered one of the dangerous and most complex for diagnosis and treatment. It can be seen that different methods are used to identify brain tumors from MR images [3]. A CNN is an artificial neural network which is specifically designed to undertake pixel data for image recognition and processing. CNN's are strong image processing and artificial intelligence (AI) systems that use deep learning to perform both generative and descriptive functions, often involving machine vision, such as image and video recognition, as well as recommender systems and natural language processing (NLP)[4]. Brain MRI images are primarily used for tumor detection and tumor progression modeling. This knowledge is specifically used in the diagnosis and treatment of tumors. On the given medical picture, an MRI image provides more data than computed tomography or ultrasound image. An MRI image provides detailed information about brain structure as well as the detection of anomalies in brain tissue. Magnetic resonance imaging (MRI) is a radiology-based medical imaging technique that produces representations of the body's structure and physiological processes in health and disease. Powerful magnetic fields, radio waves, and field gradients are used in MRI scanners to produce images of the inside of the body. The learning of nuclear magnetic resonance is the foundation of MRI. When certain atomic nuclei are exposed to an external magnetic field, they can absorb and produce radiofrequency energy [5].

II. LITERATURE REVIEW

In 2014, Komal Sharma, Akwinder Kaur, Shruti Gujral [6] proposed a study to develop a machine learning-based brain tumor detection system. The Multilayer Perceptron and Naïve Bayes machine learning algorithms are used for classification, and the maximum accuracy is achieved by using 212 samples of brain MR images. The experiment was carried out on 212 brain MR images. The MLP (Multi-Layer Perceptron) offers greater precision but takes longer to build. Considering a large data set would almost certainly improve this precision.

In 2017, S. S. Hunnur, A. Raut, and S. Kulkarni [7] demonstrated brain tumor detection by using thresholding algorithms and described the comparative study about tumor detection. The obtained results are shown, which demonstrate efficient tumor detection as well as tumor boundary extraction using the Sobel edge detection operator. The tumor's size and stage are identified. MRI images are the most effective for detecting brain tumors. Digital imaging techniques are important in this study for detecting a brain tumor in MR images.

In 2019, Gökalp Çınarer, Bülent Gürsel Emiroğlu [8] in their study SVM (Support Vector Machines) has a high sensitivity and precision, and they are much more effective than other classification techniques. It is almost assured that using a large data set and deducting density-based factors in addition to texture-based features would improve accuracy.

In 2019, Sunanda Das, O. F. M. Riaz Rahman Aranya, Nishat Nayla Labiba [9] worked on making a CNN classifier that can differentiate between 3 types of tumors (glioma, meningioma, pituitary). Pre-processing includes applying a Gaussian filter to images and implementing the histogram equalization method to the sieved images. The CNN model is then used to classify the videos. The model has an excessive number of parameters, and it was experienced on a very limited quantity of data. As a result, there is a chance of overfitting.

In 2021, Neha Sharma, Mradul Kumar Jain, Nirvikar and Amit Kumar Agarwal [10] concentrated on the primary goal of this research, which is to create a useful automated brain tumor classification system with high precision, speed, and simplicity. The accuracy can be increased by using transfer learning on large datasets. And can use a better preprocessing technique to enhance the performance of the model.

In 2020, S. Smys, Joy Iong Zong Chen, and Subarna Shakya [25] this study of literature sheds light on deep learning neural networks. Deep networks are classified into generative architectures, discriminative architectures, and mixed architectures based on their implementations and design methods. Various deep learning models are examined in these three groups, in addition to its more recent application-based methods. It is suggested that the CNN or deep CNN to be improved in this study.

III. Deep Learning Techniques

Deep learning is an AI function which mimics how the human brain interprets data and creates patterns to make decisions. It is a branch of machine learning that uses neural networks to learn unsupervised from unstructured or unlabelled data [11]. Deep learning techniques that can be used to solve a wide range of problems [12].

1. Fully Connected Neural Networks

For a completely connected neural network, each neuron in the previous layer is attached to any neuron in the succeeding layer, and each neuron has an activation mechanism that influences the neuron's output based on its input.

2. Convolutional Neural Networks

It is a kind of deep neural network that is specifically designed to classify images. It consists of an input layer which is typically a two-dimensional array of neurons and an output layer which is a one - dimensional set of output neurons.

3. Recurrent Neural Network

The current prediction of a recurrent neural network is based on knowledge of the previous state, and the process can be repeated for an undetermined number of steps.

4. Generative Adversarial Network

It is a deep learning neural network that combines two deep learning networks: a Discriminator Network and a Generator Network. The Generator Network creates fake data, while the Discriminator Network tries to tell the difference between real and fake data. These two networks are adversaries in the way that they are both trying to outperform each other. The Generator is attempting to create simulated data that is indistinguishable from actual data, while the Discriminator is attempting to improve its ability to detect fake data.

5. Deep Reinforcement Learning

Reinforcement learning comprises an agent interacting with a surrounding. Inside the world, the agent is attempting to accomplish some form of objective. The agent will observe the state of the environment. For detecting brain tumours, MRI images are the best choice.

IV. Proposed System

The proposed scheme for brain tumour detection using the Convolutional Neural Network (CNN) technique is depicted in Figure 1. In this suggested technique, the MRI image in JPEG format is provided to the system. The system then detects the brain tumor in the image. We have created our own large dataset by using augmentation. Thus, we get more samples of the same MRI image with different angles by augmentation in order to produce accurate results. The enhanced image is then analyzed using the CNN algorithm to determine if the brain is compromised by the tumor or not. In less time, this mechanism helps the neurologists to have detailed results.

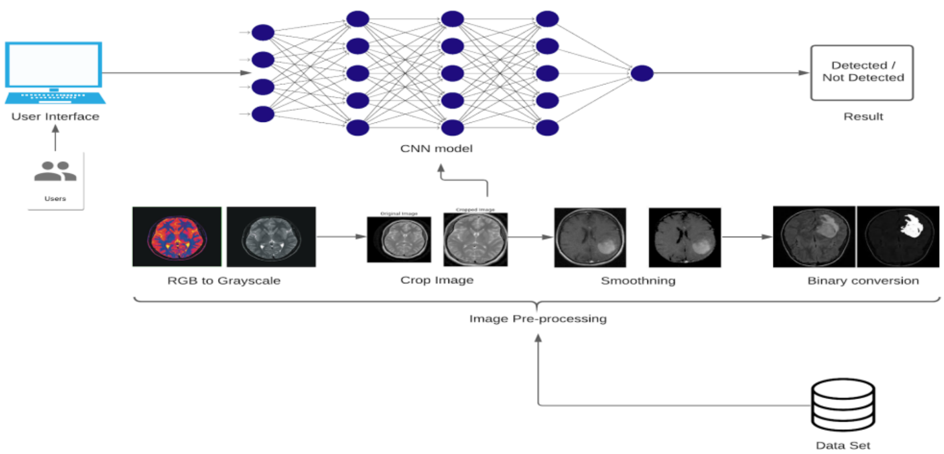

Fig 1: Block diagram of the proposed system.

A. Augmentation

Every Machine Learning model needs some data for training purposes. The data is extremely important because the future predictions entirely depend upon the training data. If the dataset is not filtered properly the resultant predictions would be abnormal. The dataset available over google is a small dataset of around 200 images only. To convert the small data set into a large data set we have used augmentation. Dataset augmentation is the process of applying simple and complex transformations like flipping or style transfer to your data which can help overcome the increasingly large requirements of Deep Learning models. In our dataset, a single image is augmented into 7 images with different angles and rotation of the original image. This large data set is used to train the model.

Table I: Dataset Statistics

Class	No. of images
Brain Tumor present	1085
Brain Tumor not present	980
Total	2065

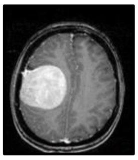

Fig 2: Original Image

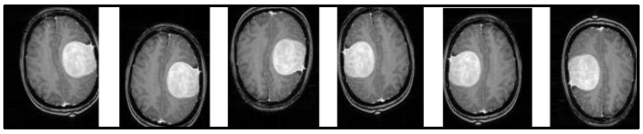

Fig 3: Dataset (Augmented Images)

B. Image Pre Processing

Pre-processing of images applies to all the transformations done on the raw data before it is provided to the deep learning algorithm. The primary goal of pre-processing is to maximize image data by suppressing unnecessary distortions or enhancing specific image features that are required for any further processing. Pre-processing of the dataset is a very important step because it enhances the quality of the image and only reflects the vital parts of the brain [13].

The proposed system uses four crucial steps of pre-processing.

- **RGB to Grayscale**

```
gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
```

Pseudo code 1: RGB to Grayscale

cvtColor() function used in pseudo code 1 is used to convert RGB (red blue green) to grayscale. This technique transforms an image from one color space to another. The OpenCV-Python library is a collection of Python bindings for solving computer vision problems. To convert an image from one color space to another, use the cv2.cvtColor() process. In OpenCV, there are more than 150 color-space conversion methods.

- **Crop Image**

Dataset images not only contain parts of the brain but also the portion which is not really required to detect the tumor. To get rid of unwanted black portions, the images in the dataset are cropped. The image is cropped from left, right, top and bottom leaving only the part of the brain in the image.

```
ExtremeLeft = tuple(c[c[:, :, 0].argmin()][0])
ExtremeRight = tuple(c[c[:, :, 0].argmax()][0])
ExtremeTop = tuple(c[c[:, :, 1].argmin()][0])
ExtremeBottom = tuple(c[c[:, :, 1].argmax()][0])
NewImg   =   image[ExtremeTop[1]:ExtremeBottomt[1],
ExtremeLeft[0]:ExtremeRight[0]]
```

Pseudo code 2: Crop Image

In pseudo code 2 the extreme most point from the left side of the MR image of the brain is identified and stored in the variable ExtremeLeft. Similarly, the extreme most points from the right, top and bottom side of the MR image of the brain are identified and stored in the variable ExtremeRight, ExtremeTop, ExtremeBottom respectively. All the four points are stored in the variable NewImg and then transferred to main function.

- **Smoothing**

Low pass filtering (also known as smoothing) is used to remove high spatial frequency noise from a digital image. Low-pass filters often employ the moving window operator, which changes the value of one pixel at a time in response to some aspect of a local area (window) of pixels. When the operator passes over the file, all of the pixels are affected.

```
gray=cv2.GaussianBlur(gray,(5,5),0)
```

Pseudo code 3: Smoothning

Gaussian blur is a technique for reducing random noise in images. Gaussian blur is the product of applying a convolution between an image and a Gaussian function. The formula for a 1D Gaussian distribution is:

$$f_g(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-a)^2}{2\sigma^2}} \quad \text{.........Equation (1)}$$

In equation 1 σ is referred to as the standard deviation of the distribution. The formula for a 2-D isotropic Gaussian is:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{.........Equation (2)}$$

We must first construct a discrete approximation of the Gaussian function before starting with the convolution since the image is made up of pixels. We need a big convolution kernel with no limits since the Gaussian distribution is non-zero everywhere. However, since the Gaussian distribution is essentially zero rather than three standard deviations from the mean in fact, we can shorten the kernel. The standard deviation value is used to measure the degree of smoothing. The Gaussian generates a 'weighted average' of each pixel's area, with the mean biased against the central pixels' meaning [15].

- **Binary Conversion**

Thresholding is a basic segmentation process that divides image regions into regions that correspond to the objects we want to study. The disparity in contrast between the object pixel and the surrounding pixels is used to separate the objects. The brain is isolated from the picture context using this feature [14]. The function cv2.threshold in pseudo code 4 is used for thresholding in OpenCV with Python. The same threshold value is used for each pixel. If the pixel value is less than or equal to the threshold, it is set to 0, otherwise it is set to the maximum value.

```
fin=cv2.threshold(gray,45,255,cv2.THRESH_BINARY)
[1]
```

Pseudo code 4: Binary Conversion

C. CNN

A CNN, or convolutional neural network, is a deep learning neural network designed to process organized arrays of data like images. Convolutional neural networks are commonly used in computer perception and have become the state-of-the-art for many visual applications such as image recognition, as well as natural language processing for text categorization. Convolutional neural networks can quickly distinguish patterns in the input signal, such as arcs, gradients, circles, or even eyes and faces. Convolutional neural networks are so powerful in computer vision because of this property. A feed-forward neural network of up to 20 or 30 layers is known as a convolutional neural network. In convolutional neural networks, several convolutional layers are stacked on top of one another, each capable of understanding more complex shapes.

- **Input Layer:**

Image data is stored in CNN's input layer. A three-dimensional matrix is used to represent image data. It needs to be remodelled into a single column. If you have a 28 x 28 = 784-pixel image, you must convert it to 784 x 1 before feeding it into the input. The dimension of input would be "m" if you have "m" training examples (784, m) [17].

```
x = Input(shape = (32,))
```

Pseudo code 5: Input Layer

- **Zero Padding:**

The zero-padding process involves adding zeroes symmetrically to the input matrix. It is a common change that makes it possible to adapt the size of the input to our needs. It is usually used to design the CNN layers if the input volume dimensions must be kept in the output volume [17].

```
X = ZeroPadding2D (padding = (1, 1), data_format =
None)
```

Pseudo code 6: Zero Padding

- **Conv2D:**

The number of filtrations from which convolutional layers can learn is a necessary Conv2D parameter. The convolution's integer value and number of output filters are also calculated.

model.add (Conv2D (32, (3, 3), padding = "same", activation = "relu"))

model.add (MaxPooling2D (pool_size = (2, 2)))

We're learning 32 filters here and then using Max Pooling to reduce the output volume's spatial dimensions. As far as choosing the suitable value for no. of filters, it is always suggested to use powers of 2 as the values [18].

```
Y = tf.keras.layers.Conv2D(2,3,padding = "same")(x)
```

Pseudo code 7: Conv2D

- **Batch Normalization:**

Batch Normalization is a normalization approach done between the layers of a Neural Network rather than in the raw data. Instead of a full data set, it is done with mini-batches. It accelerates training and uses higher rates to facilitate learning.

Following the technique explained in the previous section, we define the normalization formula of Batch Normalization as

$$z^N = \left(\frac{z-m_z}{s_z}\right) \text{.........Equation (3)}$$

Being m_z the mean of the neurons' output and s_z the standard deviation of the neurons' output [19].

```
X = BatchNormalization(axis=1, center=True,
scale=True)(X)
```

Pseudo code 8: Batch Normalization

- **Relu Activation:**

In a neural network, the activation function converts the node's weighted summed input into node or outcome activation for that input. The rectified linear activation function, or ReLU, is a piecewise linear function that provides the output as the input directly if the input is positive otherwise gives zero output. For a wide range of neural networks, it has become the default activation function, as a model that is easy to train and often performs much better [20].

```
X = Activation('relu')(X)
```

Pseudo code 9: Relu Activation

- **MaxPooling2D:**

Max pooling is a pooling activity that selects the maximum element from the area of the filter-related feature map. Thus, the result following the maximum pooling layer is a map with the most notable features on the preceding function map [21].

```
X = MaxPooling2D (pool_size = (2, 2), strides =
None, padding = "valid") (X)
```

Pseudo code 10: MaxPooling2D

- ***Flatten*:**

Flattening is the process of transforming data into a one-dimensional array for providing input in the next layer. We flatten the contribution of the convolutional layers to create a single long feature vector. It is associated with the final classification model, known as a fully connected layer [22]. Flatten layers allow you to transform data from a vector of 2d matrices (or, more accurately, nd matrices) into the correct format for a dense layer to understand. It simply

```
X = Flatten(data_format = None)(X)
```

allows this different layer type to operate on the data.

Pseudo code 11: Flatten

- **Dense Layer:**

The Dense Layer, which is the most common and frequently used layer, is a comprehensive strongly connected neural network layer. The following procedure is performed on the input by the dense layer, and the output is returned. [23].

```
y = Dense(16, activation='sigmoid')(x)
```

Pseudo code 12: Dense Layer

D. Performance Evaluation

The F1 score is used to test the system's performance.
In relation to a particular positive class, the F1 score blends precision and recall. The F1 score is a measured average of precision and recall, with 1 representing the best and 0 representing the worst [24].
Formula for F1 score is given as.
F1 = 2 * (precision * recall) / (precision + recall)
The F1 score evaluated for the proposed system is;
F1 score: 89.16%

Table II: Data Sample

	Total Dataset	Training Data	Validation Data	Testing Data
Number of examples	2064	1444	310	310
% of positive examples	52.51%	52.97%	54.83%	48.06%
Number of positive examples	1084	765	170	149
% of negative examples	47.48%	47.02%	45.16%	51.93%
Number of negative examples	980	679	140	161

V. Experimental Results

In this part, we will look at the results of the CNN algorithm, which is a classification technique. A total of 2065 brain MR images were used in this study. The system detects the tumor in the brain and provides us the output in the form of text such as tumor detected or tumor not detected. A large dataset is used in the framework to provide a high efficiency rate. A total of 80% of this dataset is used to train the device, while the remaining 20% is used to test for the existence of a brain tumour. At the end we get a system with better performance and more precision. The accuracy and f1 score of the classifier's results were used to test its efficiency.

Table III: Model Performance

Model	Accuracy	F1 score	Training time
Custom made model using CNN	87.42%	89.16%	65 sec

Table IV: Confusion Matrix

		Estimation Class	
Actual Class		Positive	Negative
	Positive	TP	TN
	Negative	FP	FN

A. Accuracy

To evaluate classification performances, the most widely used classification accuracy is used to assess the classifier's overall effectiveness.

$$Accuracy = \frac{TN+TP}{TP+FP+TN+FN} \text{........Equation (4)}$$

There is an accuracy of 87.42%. Based on this Training accuracy and validation accuracy is displayed in fig 4.

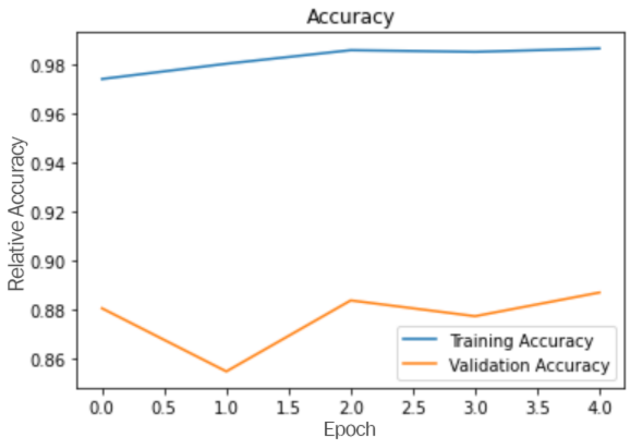

Fig. 4: Graphical representation of the developed system differentiating between Training Accuracy and Validation Accuracy

Figure 4 depicts a graphic representation of the established framework that distinguishes between training and validation accuracy. Furthermore, the highest level of accuracy achieved is 87.42%.

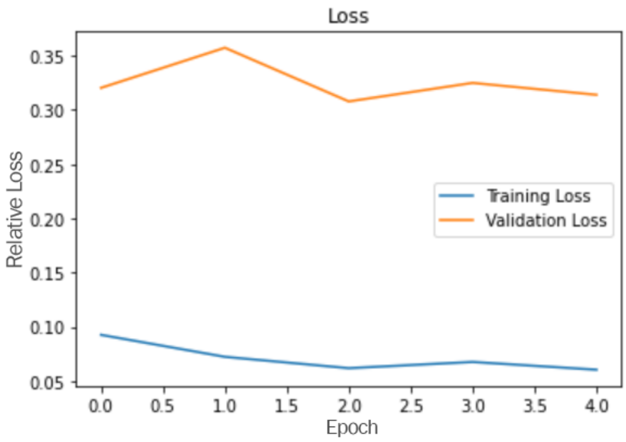

Fig. 5: Graphical representation of the developed system differentiating between Training Loss and Validation Loss

Figure 5 depicts a graphic representation of the established system that distinguishes between Training and Validation Losses. Furthermore, the total loss is 33.25%.

B. F1 Score

Results other than classification accuracy should be analyzed in order to make an accurate decision about classifier efficiency. The F1 score calculated for this reason assesses the relationship between the data's positive information and the classifier's performance.

$$f1\ score = \frac{2TP}{2TP+FP+FN} \text{.........Equation (5)}$$

The f1 score is 89.16% and gives high efficiency.

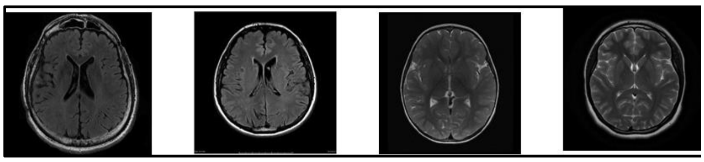

Fig. 6: Tumor not detected

Figure 6 indicates the MRI images which do not contain tumor and the system has successfully given the output.

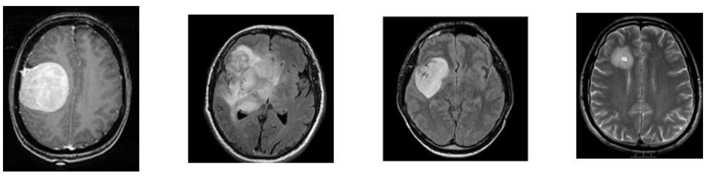

Fig. 7: Tumor detected

Figure 7 displays MRI images that reveal the presence of a tumour in the brain.

VI. Conclusion and Future Scope

Brain Tumor Detection is a very crucial part of Medical Science. In this paper, we focused on developing a system that provides the user the best possible output. To do so, a broad dataset is used in the system, which eventually leads to a high efficiency rate. Convolutional Neural Network that is a deep learning algorithm is used by the framework. This algorithm is used for image recognition and object detection. At the end we get a system with better performance and 89% of accuracy rate. Future work will focus on classifying the type of tumor in the brain which will give a clear idea of the tumor. The system will also be trained to provide the size and location of tumor in the brain so as the patient can get appropriate treatment. The system is designed in such a way to detect only the existence of the tumor in the brain. The output for the same is given in yes / no format. Web Application will be developed which will display the output and also will give some information about tumor and treatment of brain tumor as well.

References

[1] Mayo Clinic, Brain Tumor, [Online] Available: https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084 [Accessed on March 15, 2020].

[2] National Brain Tumor Society, Quick Brain Tumor Facts, [Online] Available:https://braintumor.org/brain-tumor-information/brain-tumor-facts/ [Accessed on March 15, 2020].

[3] Cancer.Net, Brain Tumor: Diagnosis, [Online], Available: https://www.cancer.net/cancer-types/brain-tumor/diagnosis [Accessed on March 16, 2020].

[4] TechTarget, Convolutional Neural Network, [Online], Available: https://searchenterpriseai.techtarget.com/definition/convolutional-neural-network [Accessed on March 17, 2020].

[5] RadiologyInfo, Magnetic Resonance Imaging (MRI), [Online], Available: https://www.radiologyinfo.org/en/info.cfm?pg=safety-mr [Accessed on March 17, 2020].

[6] Komal Sharma, Akwinder Kaur, Shruti Gujral "Brain Tumor Detection based on Machine Learning Algorithms", *International Journal of Computer Applications (0975 – 8887) Volume 103 – No.1*, October 2014, 103. 7-11. 10.5120/18036-6883.

[7] S. S. Hunnur, A. Raut and S. Kulkarni, "Implementation of image processing for detection of brain tumors," *International Conference on Computing Methodologies and Communication (ICCMC),* Erode, India, 2017, pp. 717-722, doi: 10.1109/ICCMC.2017.8282559.

[8] G. Çınarer and B. G. Emiroğlu, "Classification of Brain Tumors by Machine Learning Algorithms," *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT),* Ankara, Turkey, 2019, pp. 1-4, doi: 10.1109/ISMSIT.2019.8932878.

[9] Sunanda Das, O.F.M. Riaz Rahman Aranya, Nishat Nayla Labiba, "Brain Tumor Classification Using Convolutional Neural Network," *1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT),* Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ICASERT.2019.8934603.

[10] Neha Sharma, Mradul Kumar Jain, Nirvikar and Amit Kumar Agarwal "Brain Tumor Classification Using CNN", *Advances and Applications in Mathematical Sciences Volume 20,* Issue 3, January 2021.

[11] Investopedia, Deep Learning, [Online], Available: https://www.investopedia.com/terms/d/deep-learning.asp, [Accessed on March 20, 2020].

[12] Zeomag, Deep Learning Techniques, [Online], Available: https://www.zeolearn.com/magazine/five-important-techniques-that-you-should-know-about-deep-learning, [Accessed on March 20, 2020].

[13] Great Learning, Image Pre-processing, [Online], Available: https://www.mygreatlearning.com/blog/introduction-to-image-pre-processing/ [Accessed on April 11, 2020].

[14] OpenCV, Thresholding Operations, [Online], Available: https://docs.opencv.org/3.4/db/d8e/tutorial_threshold.html, [Accessed on April 15, 2020].

[15] Vibhakar, Ami & Tiwari, Mukesh & Singh, Dr Jaikaran. (2012). "Performance Analysis for MRI denoising using Intensity Averaging Gaussian Blur Concept and Its Comparison with Wavelet Transform Method". *International Journal of Computer Applications (IJCA)*, Volume 58, No.15, November 2012, 58. 21-26. 10.5120/9359-3694.

[16] Wikipedia, Convolutional Neural Network, [Online], Available: https://en.wikipedia.org/wiki/Convolutional_neural_network, [Accessed on April 25, 2020].

[17] Towards Data Science, Convolutional Neural Network, [Online], Available:https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529, [Accessed on April 26, 2020].

[18] GeeksforGeeks, Keras.Conv2D Class, [Online], Available: https://tinyurl.com/wxaj5dpa, [Accessed on April 26, 2020].

[19] Baeldung, Batch Normalization in CNN, [Online], Available: https://www.baeldung.com/cs/batch-normalization-cnn, [Accessed on April 26, 2020].

[20] Machine Learning Mastery, Rectified Linear Unit (ReLU), [Online], Available:https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, [Accessed on April 28, 2020].

[21] GeeksforGeeks, Introduction to Pooling Layer, [Online], Available: https://tinyurl.com/yrs5htxj, [Accessed on April 26, 2020]

[22] Towards Data Science, Convolutional Neural Network [Online], Available:https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480#:~:text=Flattening%20is%20converting%20the%20data,called%20a%20fully-connected%20layer, [Accessed on April 26, 2020].

[23] Tutorialspoint, Keras-Dense Layer, [Online], Available: https://www.tutorialspoint.com/keras/keras_dense_layer.htm#:~:text=Dense%20layer%20is%20the%20regular,input%20and%20return%20the%20output, [Accessd on April 27, 2020].

[24] Ritchie Ng, F1 Score, [Online], Available: https://www.ritchieng.com/machinelearning-f1-score/, [Accessed on July 10, 2020].

[25] Smys, S., Joy Iong Zong Chen, and Subarna Shakya. "Survey on Neural Network Architectures with Deep Learning." Journal of Soft Computing Paradigm (JSCP) 2, no. 03 (2020): 186-194.