



Brain tumor detection using CNN based on a Standard Deep Learning Model

Dr. Irene Getzi¹, Ashmika Shandilya²

Assistant Professor, Jyoti Nivas College Autonomous, Bangalore, India¹

Student, Jyoti Nivas College Autonomous, Bangalore, India²

Abstract: This research examines the development and application of an automated brain tumour detection system for robust imaging modalities such as MRI and CT scan using convolutional neural networks for exploratory diagnosis. Medical images or anatomical imaging are derived from MRI and CT scans, which play a significant role in this method. The architecture involves a series of processes such as injecting features to medical images specifically, applying pre-processing to enhance discriminative power in the input data. Deep learning architectures entail the utilisation of several network architecture such as CNN and various other state-of-the-art models.

1. INTRODUCTION

This research uses a kind of machine learning that combines traditional medical science and advanced computational approaches to diagnosis. By exploiting patterns in data from medical imaging, the research aims to use machine learning to classify and distinguish images of normal brain tissue from brain tumors.

2. LITERATURE SURVEY

G.Hemanth, M.Janardhan,[1], The research proposes an automatic segmentation method that relies upon CNN (Convolution Neural Networks), determining small 3 x 3 kernels.

The proposed approach yields assuring outcome as per the standard medical image benchmarks. CNN (Convolutional Neural Network) or ConvNet is a deep machine learning algorithm adopted to examine the Image. It utilizes various multilayer perceptions framed to gain comparatively reduced pre - processing time.

Gopal S. Tandel, Antonella Balestrieri, Tanay Jujaray, Narender N. Khanna, Luca Saba, Jasjit S. Suri,[2], The Authors developed five clinically relevant multiclass datasets (two- to six-class) and, further, proposed a transfer-learning based Artificial Intelligence scheme for brain tumour grading/classification from MRI data using a CNN. We benchmarked the CNN using six traditional machine learning methods (Decision Tree, Linear Discrimination, Naive Bayes, Support Vector Machine, K-nearest neighbor, and Ensemble). To this end, we found that CNN-based AlexNet transfer learning model outperformed. It provided mean accuracies of 100%, 95.97%, 96.65%, 87.14%, and 93.74% for the developed datasets using three cross-validation protocols (K2, K5, and K10).

R. Pugalenth, M.P. Rajakumar, J. Ramya, V. Rajinikanth,[3], This work pretends to use a Machine-Learning-Technique (MLT) on the brain MRI slices in order to discover which regions have low/high grade. The MLT uses depth sequences for three dimensional classification. The depth sequences are build following preprocessing, postprocessing and classification procedures. The preprocessing improves the tumor zone with Social Group Optimization(SGO) algorithm assisted Fuzzy-Tsallis thresholding.

Ankit Ghosh,[4], Nine popular algorithms are used to classify MRI images such as SVM, Logistic Regression, KNN, Naïve Bayes, Decision Tree, Random Forest, XGBoost , Stochastic Gradient Descent & Gradient Boosting. Their comparative analysis is done upon their accuracy, recall , precision, f1-score, AUC-ROC and AUC-PR. Gradient Boosting has out performed the others. Obtained the value of 92.4% for Accuracy, 94.4% for recall, 85% for precision, 89.5% for f1-score, 97.2% for AUC-ROC and 91.4% for Auc-pr. The XGBoost classifier has outperformed all others.

Komal Sharma, Akwinder Kaur, Shruti Gujral,[5], This paper works on a system to detect brain tumor using machine learning algorithms. It extracts texture features of the image using GLCM (Grey Level Co-occurrence Matrix). The texture features of the image under consideration are energy, contrast, correlation and homogeneity.



Dena Nadir George, Hashem B. Jehlol, Anwer Subhi Abdulhussein Oleiwi,[6], The proposed work describes detection of brain tumor in MR image (Magnetic Resonance Image) of human brain. It includes sigma filtering of MR signal, adaptive thresholding of sigma filtered signals and detection of brain regions through region based segmentation. Shape features such as Major and Minor Axis Length, Euler Number, Solidity, Area and Circularity are calculated out of brain regions which is used as input to two classifiers C4.5 decision tree and Multi-Layer Perceptron (MLP) for classifying human brain as normal or abnormal which further decipher into one benign and five malignant tumors class. The most skilled algorithm MLP computes a maximum precision of 95% with 174 brain MR images.

3. PROPOSED METHODOLOGY

The proposed research methodology for the detection of brain tumor on MRI images utilizes Convolutional Neural Network (CNN) based on a standard deep learning model. The CNN model is compared with the Support Vector Machine (SVM) model as well. A standard deep learning framework is executed which consists of seven steps: collecting a dataset of MRI images for observation, image preprocessing like image normalization and data augmentation to improve data quality, splitting the dataset into training, validation, and testing sets.

With the help of hyperparameter tuning, CNN model will be trained on the training set (consisting all sample images). Then, trained CNN model together with SVM model (which learn the underlying structural patterns from the same set) will be evaluated on their line charts using metrics, such as accuracy, sensitivity, specificity and f1-score. This research will develop an accurate and effective automated detection process of brain tumors using the best features of CNN and SVM models. The last step will be testing these models on a set of independent MRI images to explore if the result is generalizable in real life situations or not.

3.1 Flow Diagram with Explanation

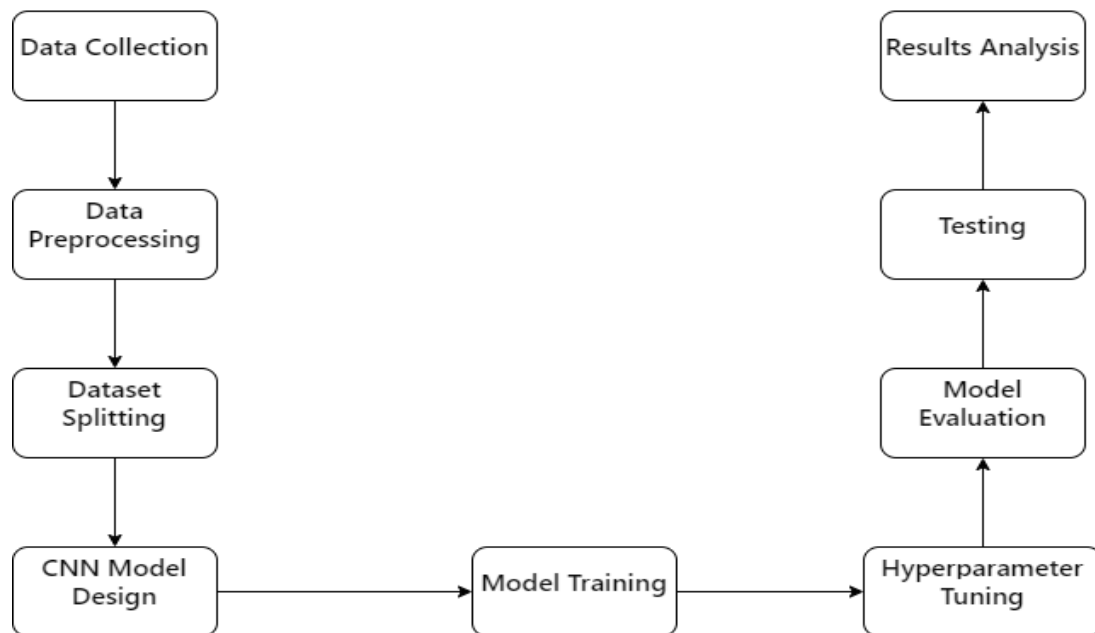


Figure 1. Flow Diagram of the Methodology

1. Data extraction: Create a massive collection of MRI images encompassing a wide variety of tumors, different stages of growth, and images without tumors at all.

2. Preprocessing the Data: Image intensities are normalised and images are augmented by rotating, scaling, and flipping them in order to make models more generalisable.

3. Split data: The dataset is divided into training set and test set. The model is trained on the training set. The model's effectiveness is tested using the test set.



4. CNN Architecture Design: We define a CNN architecture for the image classification task and specify the number of layers, filter sizes, and activation functions such that hierarchical representations of space are captured.

5. Model Training: It trains the CNN using an error-minimisation algorithm (e.g Batch stochastic gradient descent) which aims to reduce the difference between predicted labels and actual labels.

6. Hyperparameter tuning: Hyperparameters (eg, learning rate, batch size) are tuned via validation on a subset of the training data to optimise model performance.

7. Model Evaluation: The trained model is evaluated on the validation set to assess performance using criteria such as accuracy, sensitivity, specificity, and AUC-ROC.

8. Testing: The model is evaluated on the test set so that the model generalises to real-world situations; that is, it works well when fed new, unseen data.

9. Post-results Analysis: Here we examine the model's strengths and weaknesses, and propose ways of tweaking or refining it to perform better.

4. DESIGN AND IMPLEMENTATION

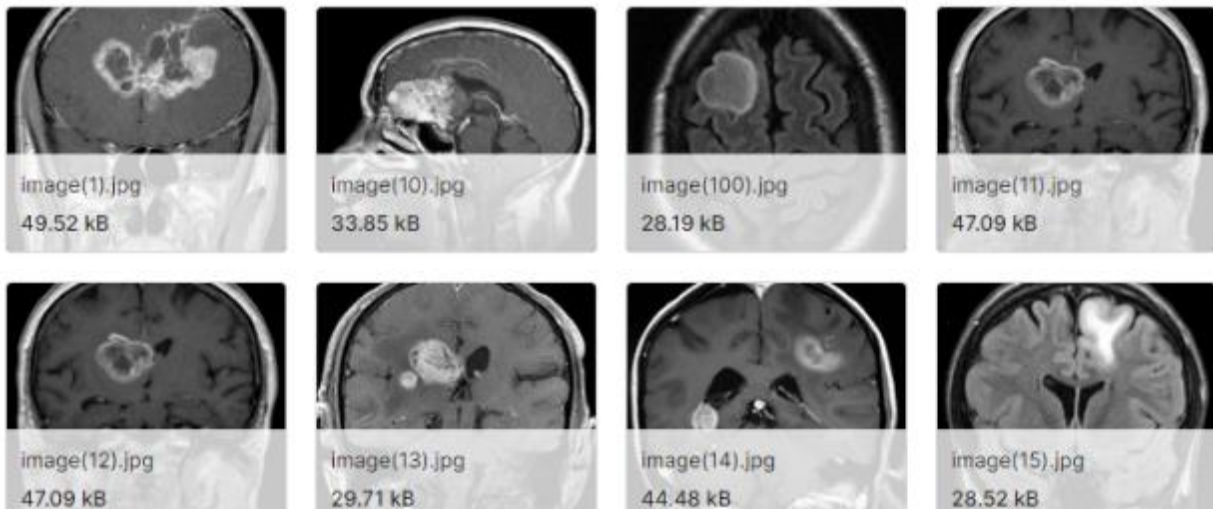
4.1 ENVIRONMENTAL SETUP

1. Library Dependencies: Python libraries used for the various elements of this research: 1. Machine learning: Keras, scikit-learn, TensorFlow 2. Other libraries: Pandas, NumPy, IPywidgets, PIL, tqdm, OpenCV, Matplotlib, Seaborn.

2. Dataset Used:

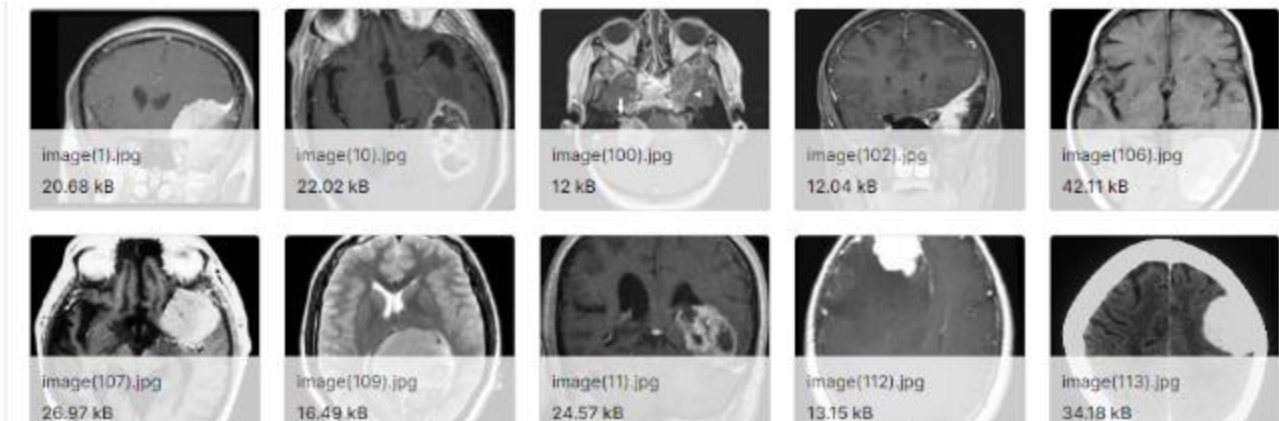
glioma_tumor: 100 files

A glioma is a tumour arising from glial cells within the brain. There are different grades of glioma, depending on their grade of malignancy.



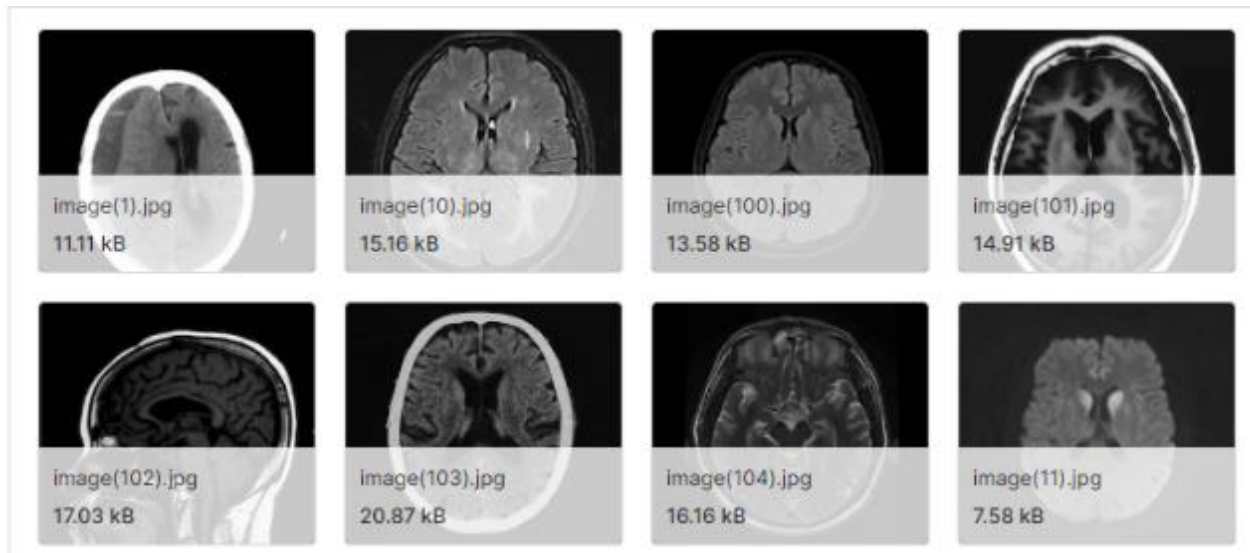
meningioma_tumor: 115 files

Meningiomas arise from the meninges, the connective tissue surrounding the brain and spinal cord. Meningiomas are typically benign. However, larger and locally aggressive tumors can affect the nearby brain tissue.



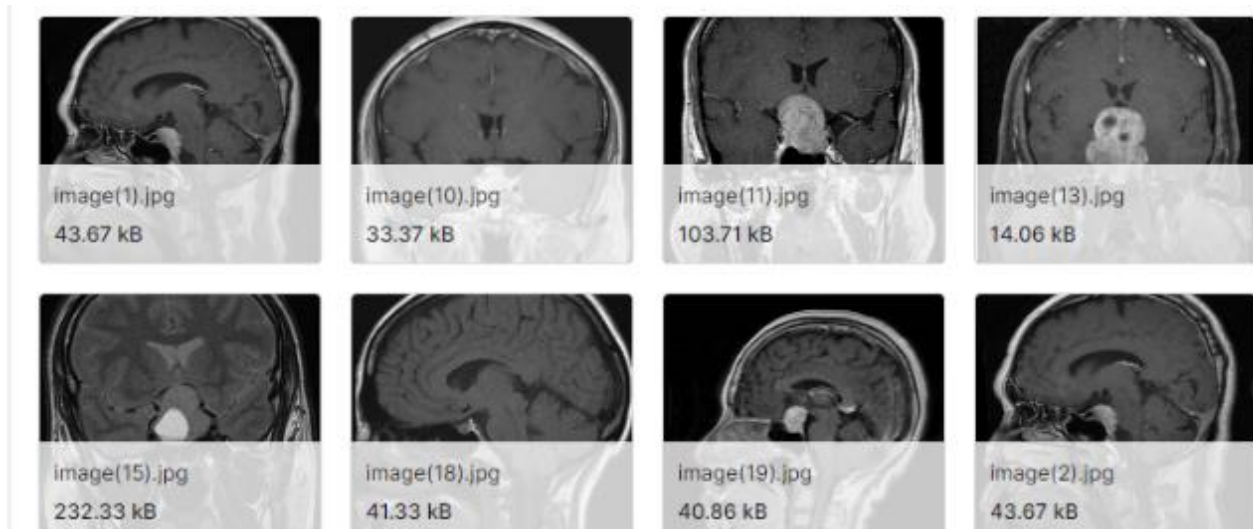
no_tumor: 105 files

This class usually shows images of brains without any visible tumour, and the normal brain represents the null hypothesis (or the reference class) by which the tumour is determined using the model.



pituitary_tumor: 74 files

Pituitary tumors develop in the pituitary, a small gland at the base of the brain that produces hormones. Tumours in the pituitary disrupt hormone regulation. Depending on type and size, a pituitary tumour can cause health problems.



4.2 WORKFLOW / IMPLEMENTATION STEPS

1. Data Preparation

1.1 Load and preprocess training and testing images

X_train and Y_train are initialized as empty lists to store training images and their corresponding labels.

The image_size is set to 150, representing the target size for resizing images.

The labels list contains four class labels for the brain tumor dataset: 'glioma_tumor', 'meningioma_tumor', 'no_tumor', and 'pituitary_tumor'.

1.2 Load training and testing images

Iterates over each label in the labels list.

Forms the path to the corresponding folder for that label in the training dataset.

Iterates over each file in the folder.

Use OpenCV (cv2.imread) to read every image and resize it to the image_size, and append it to X_train.

Appends the label i to Y_train for each corresponding image.

1.3 Convert lists to numpy arrays

np.array() Conversion: a function of the NumPy library that transforms a Python list into a NumPy array. This conversion is typically accompanied by the fact that NumPy arrays provide faster numerical operations and are used as the data type of machine learning libraries such as TensorFlow and scikit-learn.

1.4 Shuffle the data

`X_train, Y_train = shuffle(X_train, Y_train, random_state=101)`

shuffle() Function: function provided by scikit-learn (a machine learning library) (from sklearn.utils) that will shuffle/randomly permute the input array or list along its first axis. Shuffling avoids any bias effects (if the data has some inherent order) when training neural networks.



1.5 Split the data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X_train, Y_train, test_size=0.2, random_state=101)
```

This `test_size=0.2` means that 20 per cent of the data will be used for testing, and the rest for training.

2. Model Definition and Training

2.1 Convolutional neural network

1. The sequential model is initialized with `model = Sequential()`.
2. Conv2D layer with increasing filters (32, 64, 128, 256), kernel size (3,3), activation function ReLU and input shape (150,150,3), respectively is added.
3. MaxPooling2D (2,2) layers is added for down-sampling and Dropout layers with rate 0.3 to prevent overfitting.
4. A Flatten layer then Dense layers of 512 neurons is added, ReLU activation and Dropout (0.3) to avoid overfitting.
5. For multi-class classification, the Dense layer has 4 neurons with softmax activation.

2.2 Model Training

1. `model.fit(X_train, y_train, epochs=20, validation_split=0.1)` # 20 epochs, 10% of training data for validation.

2. An object called history is returned, including loss and accuracy scores for both the training and validation sets for each epoch.

3. Model Evaluation and Visualization

Visualizes the training and validation accuracy of a neural network model over epochs using Matplotlib.

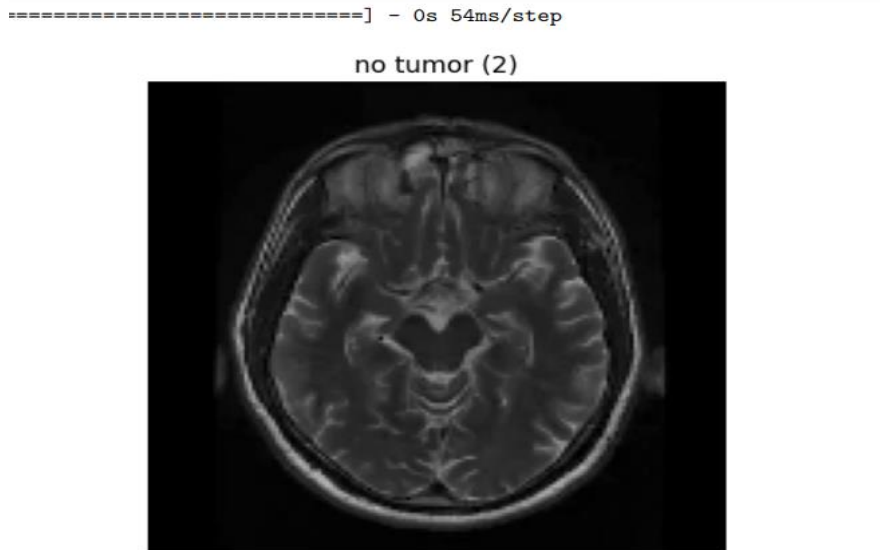


4. Image Prediction

1. Image Processing: Brain MRI example image that is loaded, resized to size (150, 150), converted to a NumPy array and reshaped to shape (1, 150, 150, 3).



2. Prediction and Display: Having made a prediction with the trained model, map the predicted index to a class label, and use the Matplotlib function to show the image with predicted information.



5. RESULT ANALYSIS

1. Prediction and Conversio: Class probabilities for the test set X_test is provided by the trained CNN, while both the test labels y_test and the predict labels y_pred_cnn, taking one-hot esncoding as a example, is converted from one-hot encoding encoding to single-dimensional labels.

2. Reporting and Scoring: The accuracy of classification is calculated, and a classification report is created using scikit-learn. The results of the operations are printed to show the model-performance report.

We compared the Convolutional Neural Network (CNN) and Support Vector Machine (SVM) models in brain tumor classification with Magnetic Resonance Imaging (MRI) data; we use the analysis of precision, recall, F1-score, and support as an assessment. The CNN performs better than SVM for all metrics.

From the chart, it can be seen that CNN is more accurate because it has a higher percentage of true positive correctly predicted cases than the total predicted positive cases. Furthermore, CNN has higher recall as it predicted a larger percentage of actual true positive cases in the dataset than in the FIR tree model. If we balance precision and recall, then the F1-score tends to be higher because CNN has a higher score compared to the FIR tree model.

In general, our experiments demonstrated that CNN is more suitable than SVM in the classification of the brain tumor instances for the reason that it can reach a more precision and comprehensive diagnostic result.

Algorithm	Accuracy
CNN	85.30%
SVM	79.09%

6. CONCLUSION

In conclusion, this study examines the use of deep learning in detecting brain tumors with the use of two different models, CNN and SVM. The CNN model obtained 85.30% accuracy which made it a model with higher performance than the SVM model. Besides, CNN showed higher precision, recall and F1-score, which are also measures of accurate predictions in highly complex and image-oriented problems. In contrast, SVM with an accuracy of 79.09%, but has poor performance in the identification of complex features in medical images.



REFERENCES

- [1]. Hemanth, G., M. Janardhan, and L. Sujihelen. "Design and implementing brain tumor detection using machine learning approach." *2019 3rd international conference on trends in electronics and informatics (ICOEI)*. IEEE, 2019. Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm by Gopal S. Tandel
- [2]. Borole, Vipin Y., Sunil S. Nimbhore, and Dr Seema S. Kawthekar. "Image processing techniques for brain tumor detection: A review." *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 4.5 (2015)
- [3]. Sharma, Komal, Akwinder Kaur, and Shruti Gujral. "Brain tumor detection based on machine learning algorithms." *International Journal of Computer Applications* 103.1 (2014).
- [4]. Sapra, Pankaj, Rupinderpal Singh, and Shivani Khurana. "Brain tumor detection using neural network." *International Journal of Science and Modern Engineering (IJISME) ISSN* (2013): 2319-6386.
- [5]. Amin, Javeria, et al. "A distinctive approach in brain tumor detection and classification using MRI." *Pattern Recognition Letters* 139 (2020): 118-127.
- [6]. Abdalla, Hussna Elnoor Mohammed, and M. Y. Esmail. "Brain tumor detection by using artificial neural network." *2018 International conference on computer, control, electrical, and electronics engineering (ICCCEEE)*. IEEE, 2018.
- [7]. Amin, Javeria, et al. "Big data analysis for brain tumor detection: Deep convolutional neural networks." *Future Generation Computer Systems* 87 (2018): 290-297..
- [8]. Hossain, Tonmoy, et al. "Brain tumor detection using convolutional neural network." *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*. IEEE, 2019.
- [9]. Sharma, Pratibha, Manoj Diwakar, and Sangam Choudhary. "Application of edge detection for brain tumor detection." *International Journal of Computer Applications* 58.16 (2012).
- [10]. Nazir, Maria, Sadia Shakil, and Khurram Khurshid. "Role of deep learning in brain tumor detection and classification (2015 to 2020): A review." *Computerized medical imaging and graphics* 91 (2021): 101940.
- [11]. U. Rajendra Acharya, S. Vinitha Sree, M. Muthu Rama Krishnan, Filippo Molinari, Roberto Garberoglio, Jasjit S. Suri, Non-invasive automated 3D thyroid lesion classification in ultrasound: a class of ThyroScan™ systems, *Ultrasonics* 52 (4) (2012) 508–520, <https://doi.org/10.1016/j.ultras.2011.11.003>.
- [12]. MATLAB Central, URL: <http://www.mathworks.in/discovery/imagesegmentation.html>.
- [13]. Kapoor, Luxit, and Sanjeev Thakur. "A survey on brain tumor detection using image processing techniques." *2017 7th international conference on cloud computing, data science & engineering-confluence*. IEEE, 2017.
- [14]. Kapoor, Luxit, and Sanjeev Thakur. "A survey on brain tumor detection using image processing techniques." *2017 7th international conference on cloud computing, data science & engineering-confluence*. IEEE, 2017.
- [15]. Chahal, Prabhjot Kaur, Shreelekha Pandey, and Shivani Goel. "A survey on brain tumor detection techniques for MR images." *Multimedia Tools and Applications* 79.29 (2020): 21771-21814.
- [16]. Ratan, Rajeev, Sanjay Sharma, and S. K. Sharma. "Brain tumor detection based on multi-parameter MRI image analysis." *ICGST-GVIP Journal* 9.3 (2009): 9-17.