Brain Tumor Classification with Convolutional Neural Network

Akshada Borhade
(D.Y. Patil International University, Pune, Maharashtra.
20190802019@dypiu.ac.in)

Nikita Verma
(D.Y. Patil International University, Pune, Maharashtra.
20190802053@dypiu.ac.in)

Dr. Maheshwari Biradar
(D. Y. Patil International University, Pune, Maharashtra.
maheshwari.biradar@dypiu.ac.in)

Sulaxan Jadhav
(D. Y. Patil International University, Pune, Maharashtra.
sulaxan.jadhav@dypiu.ac.in)

Abstract: Brain tumors are one of the top causes of death throughout the world. Brain tumor diagnosis and classification are critical for efficient treatment planning. Convolutional Neural Networks (CNNs) have demonstrated outstanding performance in image classification applications, including medical picture analysis. In this study, a CNN model is created to categorise brain tumors into four types: glioma, meningioma, pituitary, and no tumor. To improve its performance, the model is trained on a dataset of brain MRI images that has been pre-processed and supplemented. The suggested approach makes advantage of normalisation from a previously trained model as well as fine-tuning on the brain tumor data-set. On the test set, the model achieves an accuracy of 0.9, demonstrating its efficacy in brain tumor classification. The findings demonstrate that CNN models may be used to accurately classify brain tumors, which can help medical practitioners make better treatment decisions.

Keywords: Artificial Intelligence, Deep Learning, Machine Learning, CNN Model, Brain Tumor Classification.
Categories:
DOI:

1. Introduction

Brain tumors are abnormal growths of cells in the brain. The diagnosis of brain tumors is an important role in medical imaging since it aids in better treatment planning and patient care. Histopathology investigation, which is time-consuming and intrusive, is one of the classic methods of diagnosis. Non-invasive identification of brain tumors is now possible because of medical imaging techniques such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT). Deep learning techniques, notably Convolutional Neural Networks (CNNs), have shown considerable promise in medical picture categorization applications in recent years. CNNs are meant to learn the features of input data automatically and have been demonstrated to outperform traditional machine learning algorithms in many picture classification tasks. Glioma is the most prevalent kind of brain tumor, accounting for around 30% of all cases. Meningioma is the second most frequent kind of brain tumor, originating in the brain and spinal cord lining. Pituitary tumors are a form of brain tumor that develops from the pituitary gland, which is in charge of regulating the body's hormonal processes. The classification of these tumors is critical for proper treatment planning, as each type of tumor necessitates a particular treatment modality. A data-set of brain MRI images is used in this project to train and evaluate the CNN model. To increase the model's performance, the data-set is pre-processed and augmented. The suggested CNN model makes advantage of normalisation from a previously trained model on the brain tumor data-set.

2. Existing models and their limitations

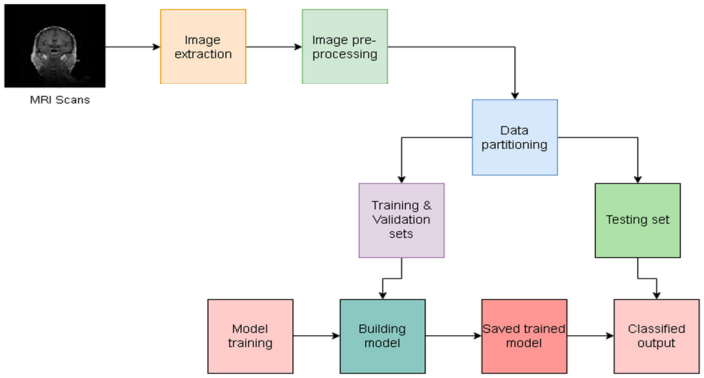
In the deep analysis of brain tumor [Mahmud 23] the CNN model consisted of many layers, compilation took a very long time. Also they lacked a decent GPU. If the dataset consists of many images then it will be a time consuming task. For the multimodal brain tumor detection [Maqsood 22] the methodology was developed for 2-D MRI scans, and the feature selection method takes time. In the Biologically Inspired Orthogonal Wavelet Transform and Deep Learning Techniques for brain tumor detection [Arif 22] the computational time, system complexity, and memory space required to execute the algorithms should all be decreased further. In the Enhanced Watershed Segmentation Algorithm-Based Modified ResNet50 Model for Brain Tumor Detection [Sharma 22] the process is time consuming due the model and its number of layers. In the Multiclass Brain tumor detection [Tiwari 22] the process faced time complexity issues. For Ensemble Transfer Learning and Quantum Variational Classifier for detecting brain tumor [Amin 22] the usage of data and images is limited. The model used for Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture [Rehman 21] gave accurate results but took a lot of time for the entire process. The model proposed by [Sadad 21] in Brain tumor detection and multi-classification using advanced deep learning techniques has a high computational complexity. In the brain tumor detection using hybrid deep neural network in MRI by adaptive squirrel search optimization [Deb 21] optimization and classifier took time. [Islam 21] used a limited dataset for Brain tumor detection in MR image using superpixels, principal component analysis and template based K-means clustering algorithm. For brain tumor detection based on

extreme learning [Sharif 21] drawbacks were noted in the segmentation accuracy. Considerate improvement is needed for the feature-concatenation technique [Rajinikanth 20] in a customized VGG19 Network with Concatenation of Deep and Handcrafted Features for Brain Tumor Detection

3. Methodology

3.1 Proposed model

Using a Kaggle dataset, a deep learning-based algorithm for brain tumor classification is built. The scans in the collection are divided into four categories: glioma tumor, meningioma tumor, no tumor, and pituitary tumor.
The scans in the dataset are pre-processed by reading them with OpenCV and resizing them to a fixed size of 150x150 pixels. The labels for each image are assigned based on the tumor class. After that, the data is shuffled and divided into training and testing sets. The labels are numerically encoded and then turned into one-hot encoded vectors. A convolutional neural network (CNN) model is built for brain tumor classification. The model architecture consists of numerous convolutional layers with different kinds of filters, followed by max-pooling and dropout layers to prevent overfitting. The last layers are fully coupled dense layers with ReLU activation. The model is then compiled using the Adam optimizer and the categorical cross-entropy loss function.
The training phase entails fitting the model to the training data over a set number of epochs. During training and validation, the accuracy and loss metrics are tracked. The model's performance is then evaluated using the testing set, and the accuracy and loss curves are visualised using the matplotlib and seaborn libraries.
A sample image is loaded from the testing folder, pre-processed, and put into the model for prediction to demonstrate the applicability of the trained model. The predicted output is obtained by selecting the softmax output index with the highest probability. The label for the identifying tumor class is then put forth.
The learned model is stored for later usage and loaded to infer new MRI data.



labels = ['glioma_tumor', 'meningioma_tumor', 'no_tumor', 'pituitary_tumor']

for each label in labels:
  for each image in the training folder for the current label:
  - Read the image using cv2
  - Resize the image to the specified image_size
  - Append the image to X_train
  - Append the label to Y_train

for each label in labels:
  for each image in the testing folder for the current label:
  - Read the image using cv2
  - Resize the image to the specified image_size
  - Append the image to X_train
  - Append the label to Y_train

Convert X_train and Y_train to numpy arrays Shuffle X_train and Y_train using a random state X_train, X_test, y_train, y_test = train_test_split Convert y_train and y_test to categorical format
model = Sequential() Add Conv2D, MaxPooling2D, Dropout, and Dense layers to the model Compile the model with loss function, optimizer, and metrics
Plot training accuracy and validation accuracy from history Plot training loss and validation loss from history
Make a prediction using the model and the test image
Make predictions on the X_test data using the saved model
Get the predicted classes and true classes by getting the indices of the highest values
Plot the confusion matrix

3.2 Pre-processing

The dataset is loaded and then the data is read and resized. The data is converted to arrays and then it is shuffled and split. The data labels are converted to categorical. The categorical data is then converted to numerical data using one hot encoding technique which is easy for the model to understand.

3.3 Proposed CNN model

The model is a convolutional neural network (CNN) that has been trained to perform image categorization tasks. It is made up of many layers that extract information from the input photos and create predictions about the class labels.

The first layer is a 3x3 kernel convolutional layer with 32 filters. It introduces nonlinearity by using the ReLU activation function. The input form is (150,150,3), indicating that the input images are 150x150 pixels in size and have three colour channels (RGB).

Following the first convolutional layer is another convolutional layer with 64 filters and a 3x3 kernel size, both of which use the ReLU activation function.

ReLU or rectified linear unit activation function is represented as- $f(x) = \max(0,x)$

To minimise the spatial dimensions of the output, a max pooling layer with a pool size of 2x2 is added. This layer aids in the capture of crucial features while minimising computation.

To prevent overfitting, a dropout layer with a rate of 0.3 is implemented during training, randomly changing a fraction of input units to 0. The dropout layer contributes to the model's regularisation and generalisation capability.

Additional convolutional layers (64 filters, 3x3 kernel) and dropout layers (rate of 0.3) are added before max pooling layers (2x2 pool size). In subsequent convolutional layers, the number of filters is raised to catch increasingly complex patterns in the images.

A flatten layer is inserted after the final convolutional layer to turn the 2D feature maps into a 1D vector. This vector is then supplied into layers that are fully connected. The first fully connected layer is made up of 512 units with the ReLU activation function, followed by another layer with 512 units and ReLU activation.

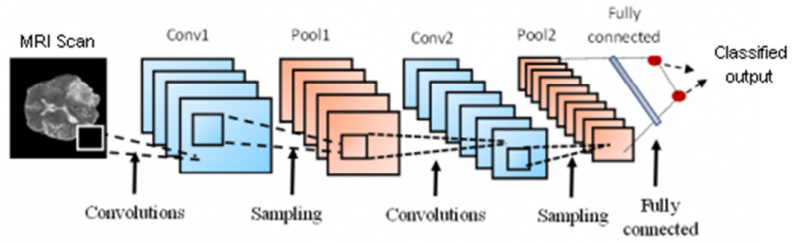
To avoid overfitting even more, a dropout layer with a rate of 0.3 is added after the fully connected layers. Finally, as the output layer, a dense layer with four units and softmax activation is incorporated. The softmax activation assigns probabilities to each class label, indicating how likely each class is.

Softmax activation function is represented as-

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1,\ldots,K$$

Overall, the hierarchical organisation of convolutional and pooling layers in this model architecture seeks to capture and learn discriminative characteristics from input

images. The fully connected layers at the model's end use these learned features to predict the class labels of the images.



Layer	Output Shape	Number of Parameters
conv2d (Conv2D)	(None, 148, 148, 32)	896
conv2d_1 (Conv2D)	(None, 146, 146, 64)	18496
max_pooling2d	(None, 73, 73, 64)	0
dropout (Dropout)	(None, 73, 73, 64)	0
conv2d_2 (Conv2D)	(None, 71, 71, 64)	36928
conv2d_3 (Conv2D)	(None, 69, 69, 64)	36928
dropout_1 (Dropout)	(None, 69, 69, 64)	0
max_pooling2d_1	(None, 34, 34, 64)	0
dropout_2 (Dropout)	(None, 34, 34, 64)	0
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73856
conv2d_5 (Conv2D)	(None, 30, 30, 128)	147584
conv2d_6 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_2	(None, 14, 14, 128)	0
dropout_3 (Dropout)	(None, 14, 14, 128)	0
conv2d_7 (Conv2D)	(None, 12, 12, 128)	147584
conv2d_8 (Conv2D)	(None, 10, 10, 256)	295168

max_pooling2d_3	(None, 5, 5, 256)	0
dropout_4 (Dropout)	(None, 5, 5, 256)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 512)	3277312
dense_1 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0

Total parameters: 4,447,044

Trainable parameters: 4,447,044

Non-trainable parameters: 0

3.4 Confusion Matrix

The confusion matrix is a table that summarises a classification model's performance by counting the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.

Confusion matrix is represented as:

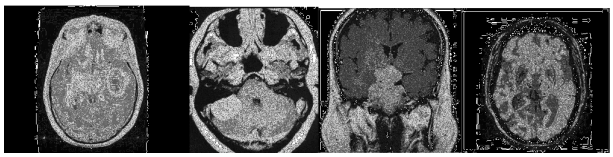
1. Accuracy (everything correct / everything) = TP + TN / TP + TN + FP + FN
2. Misclassification (everything incorrect / everything) = FP + FN / TP + TN + FP + FN
3. Precision = TP / TP + FP (actual positives / projected positives).
4. Sensitivity, often known as recall (true positives / all actual positives)= TP / TP + FN.
5. Specificity (all actual negatives / true negatives) =TN / TN + FP

4. Results and Discussions

The proposed model gives an accuracy of 0.9. The model takes approximately 3 minutes to compile as it uses GPU acceleration runtime. The whole process takes 5-6 minutes to give results. The dataset consists of 3274 MRI scans collected from kaggle.

4.1 Dataset samples-

The following images show glioma, meningioma, pituitary and no tumor sample images respectively from the kaggle dataset.



4.2 Obtained Output

1. Glioma Tumor- When the test image of a scan having glioma tumor is fed to the model, the output we get matches the input-

```
[36] img = cv2.imread('/content/drive/MyDrive/archive/Testing/glioma_tumor/image(12).jpg')
     img = cv2.resize(img,(150,150))
     img_array = np.array(img)
     img_array=img_array.reshape(1,150,150,3)

[37] model.save('braintumor.h5')

[38] saved_model=keras.models.load_model("braintumor.h5")

     a=model.predict(img_array)
     indices = a.argmax()
     indices
     if indices==0:
       print("Glioma_tumor")
     elif indices==1:
       print("Meningioma_tumor")
     elif indices==2:
       print("No Tumor")
     elif indices==3:
       print("Pituitary_tumor")

     1/1 [==============================] - 0s 114ms/step
     Glioma_tumor
```

2. Meningioma Tumor- When the test image of a scan having meningioma tumor is fed to the model, the output we get matches the input-

```
[40] img = cv2.imread('/content/drive/MyDrive/archive/Testing/meningioma_tumor/image(102).jpg')
     img = cv2.resize(img,(150,150))
     img_array = np.array(img)
     img_array=img_array.reshape(1,150,150,3)

[41] model.save('braintumor.h5')

[42] saved_model=keras.models.load_model("braintumor.h5")

     a=model.predict(img_array)
     indices = a.argmax()
     indices
     if indices==0:
       print("Glioma_tumor")
     elif indices==1:
       print("Meningioma_tumor")
     elif indices==2:
       print("No Tumor")
     elif indices==3:
       print("Pituitary_tumor")

     1/1 [==============================] - 0s 111ms/step
     Meningioma_tumor
```

3. Pituitary Tumor- When the test image of a scan having pituitary tumor is fed to the model, the output we get matches the input-

```
[44] img = cv2.imread('/content/drive/MyDrive/archive/Testing/pituitary_tumor/image(21).jpg')
     img = cv2.resize(img,(150,150))
     img_array = np.array(img)
     img_array=img_array.reshape(1,150,150,3)

[45] model.save('braintumor.h5')

[46] saved_model=keras.models.load_model("braintumor.h5")

     a=model.predict(img_array)
     indices = a.argmax()
     indices
     if indices==0:
       print("Glioma_tumor")
     elif indices==1:
       print("Meningioma_tumor")
     elif indices==2:
       print("No Tumor")
     elif indices==3:
       print("Pituitary_tumor")

     1/1 [==============================] - 0s 116ms/step
     Pituitary_tumor
```

4. No tumor- When the test image of a no tumor scan is fed to the model, the output we get matches the input-

```
img = cv2.imread('/content/drive/MyDrive/archive/Testing/no_tumor/image(15).jpg')
img = cv2.resize(img,(150,150))
img_array = np.array(img)
img_array=img_array.reshape(1,150,150,3)
```

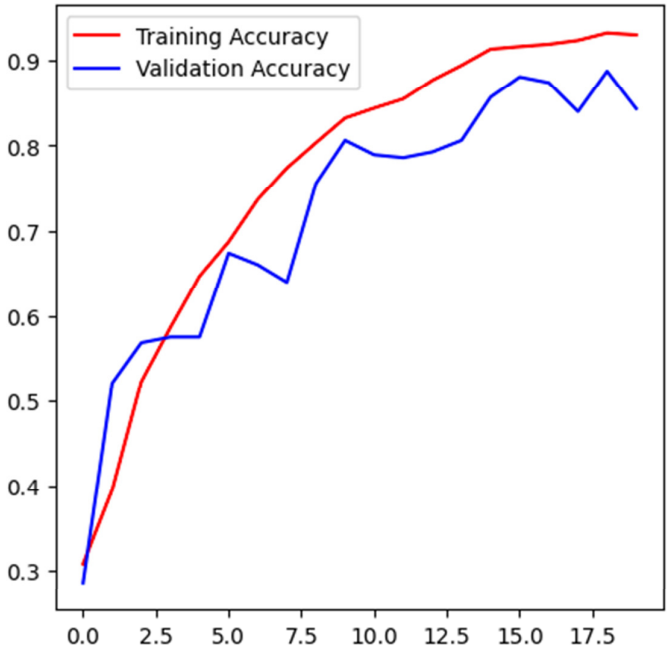
```
[31] model.save('braintumor.h5')
```

```
[32] saved_model=keras.models.load_model("braintumor.h5")
```

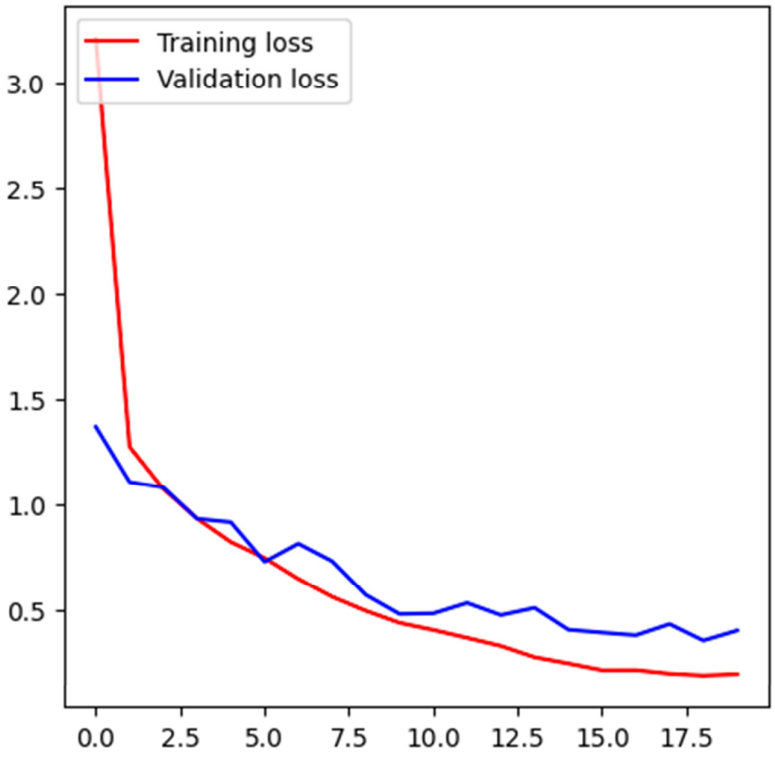
```
a=model.predict(img_array)
indices = a.argmax()
indices
if indices==0:
  print("Glioma_tumor")
elif indices==1:
  print("Meningioma_tumor")
elif indices==2:
  print("No Tumor")
elif indices==3:
  print("Pituitary_tumor")
```

```
1/1 [==============================] - 0s 69ms/step
No Tumor
```

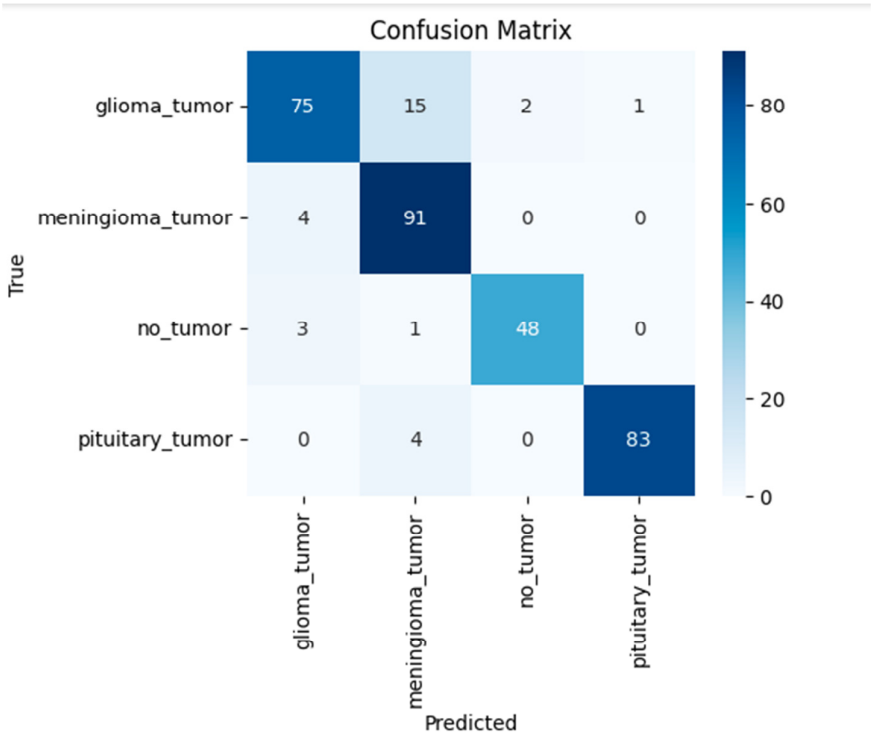
5. Accuracy Graph-



6. Loss Graph-



7. Confusion Matrix-



4.3 Applications

The trained model can be employed in medical diagnosis for automated brain tumour categorization. Based on medical pictures, it can help clinicians precisely diagnose and categorise various types of brain tumours. The model can assist healthcare practitioners in making decisions. It can serve as an expert second opinion.

8. Conclusion

In this paper, we have suggested a method for classifying four types of brain tumours namely glioma, meningioma, pituitary and no tumor respectively using a basic CNN model. The entire process is implemented on Google colab with GPU runtime acceleration taking roughly about 5-6 minutes. When this process is executed without a GPU setup it takes roughly 3 hours minimum to give results. We also noted that when we use normalisation in pre-processing the accuracy is less. Without using normalisation we get an accuracy of 0.9. When any random image is fed to the model the output is given correctly as no tumor. Overall, the method that we suggested in this paper is reliable and can be used by doctors for saving lives.

9. Future Work

The project focuses only on brain tumor, in the future we can use it for different brain diseases, Alzheimer's and other cancer diseases too.

References

[Mahmud 23] Mahmud, M.I.; Mamun, M.; Abdelgawad, A. A Deep Analysis of Brain Tumor Detection from MR Images Using Deep Learning Networks. *Algorithms* 2023, *16*, 176. https://doi.org/10.3390/a16040176

[Asad 23] Asad, R.; Rehman, S.u.; Imran, A.; Li, J.; Almuhaimeed, A.; Alzahrani, A. Computer-Aided Early Melanoma Brain-Tumor Detection Using Deep-Learning Approach. *Biomedicines* 2023, *11*, 184. https://doi.org/10.3390/biomedicines11010184

[Ali 23] Ali, M.U.; Hussain, S.J.; Zafar, A.; Bhutta, M.R.; Lee, S.W. WBM-DLNets: Wrapper-Based Metaheuristic Deep Learning Networks Feature Optimization for Enhancing Brain Tumor Detection. *Bioengineering* 2023, *10*, 475. https://doi.org/10.3390/bioengineering10040475

[Maqsood 22] Maqsood, S.; Damaševičius, R.; Maskeliūnas, R. Multi-Modal Brain Tumor Detection Using Deep Neural Network and Multiclass SVM. *Medicina* 2022, *58*, 1090. https://doi.org/10.3390/medicina58081090

[Arif 22] Muhammad Arif, F. Ajesh, Shermin Shamsudheen, Oana Geman, Diana Izdrui, Dragos Vicoveanu, "Brain Tumor Detection and Classification by MRI Using Biologically Inspired Orthogonal Wavelet Transform and Deep Learning Techniques", *Journal of Healthcare Engineering*, vol. 2022, Article ID 2693621, 18 pages, 2022. https://doi.org/10.1155/2022/2693621

[Sharma 22] Arpit Kumar Sharma, Amita Nandal, Arvind Dhaka, Deepika Koundal, Dijana Capeska Bogatinoska, Hashem Alyami, "Enhanced Watershed Segmentation Algorithm-Based Modified ResNet50 Model for Brain Tumor Detection", BioMed Research International, vol. 2022, Article ID 7348344, 14 pages, 2022. https://doi.org/10.1155/2022/7348344

[Tiwari 22] Tiwari P, Pant B, Elarabawy MM, Abd-Elnaby M, Mohd N, Dhiman G, Sharma S. CNN Based Multiclass Brain Tumor Detection Using Medical Imaging. Comput Intell Neurosci. 2022 Jun 21;2022:1830010. doi: 10.1155/2022/1830010. PMID: 35774437; PMCID: PMC9239800.

[Amin 22] Javeria Amin, Muhammad Almas Anjum, Muhammad Sharif, Saima Jabeen, Seifedine Kadry, Pablo Moreno Ger, "A New Model for Brain Tumor Detection Using Ensemble Transfer Learning and Quantum Variational Classifier",

Computational Intelligence and Neuroscience, vol. 2022, Article ID 3236305, 13 pages, 2022. https://doi.org/10.1155/2022/3236305

[Woźniak 21] Woźniak, M., Siłka, J. & Wieczorek, M. Deep neural network correlation learning mechanism for CT brain tumor detection. Neural Comput & Applic (2021). https://doi.org/10.1007/s00521-021-05841-x

[Rehman 21] Rehman, A, Khan, MA, Saba, T, Mehmood, Z, Tariq, U, Ayesha, N. Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture. Microsc Res Tech. 2021; 84: 133– 149. https://doi.org/10.1002/jemt.23597

[Sadad 21] Sadad T, Rehman A, Munir A, Saba T, Tariq U, Ayesha N, Abbasi R. Brain tumor detection and multi-classification using advanced deep learning techniques. Microsc Res Tech. 2021 Jun;84(6):1296-1308. doi: 10.1002/jemt.23688. Epub 2021 Jan 5. PMID: 33400339.

[Liu 21] Liu T, Yuan Z, Wu L, Badami B. An optimal brain tumor detection by convolutional neural network and Enhanced Sparrow Search Algorithm. Proc Inst Mech Eng H. 2021 Apr;235(4):459-469. doi: 10.1177/0954411920987964. Epub 2021 Jan 13. PMID: 33435847.

[Kumar 21] Kumar, D. & Satyanarayana, D. & Prasad, M.. (2021). MRI brain tumor detection using optimal possibilistic fuzzy C-means clustering algorithm and adaptive k-nearest neighbor classifier. Journal of Ambient Intelligence and Humanized Computing. 12. 1-14. 10.1007/s12652-020-02444-7.

[Deb 21] Deb, Daizy & Roy, Sudipta. (2021). Brain tumor detection based on hybrid deep neural network in MRI by adaptive squirrel search optimization. Multimedia Tools and Applications. 80. 1-25. 10.1007/s11042-020-09810-9.

[Islam 21] Md Khairul Islam, Md Shahin Ali, Md Sipon Miah, Md Mahbubur Rahman, Md Shahariar Alam, Mohammad Amzad Hossain, Brain tumor detection in MR image using superpixels, principal component analysis and template based K-means clustering algorithm, Volume 5, 2021, 100044, ISSN 2666-8270, https://doi.org/10.1016/j.mlwa.2021.100044

[Sharif 21] Sharif, M., Amin, J., Raza, M. et al. Brain tumor detection based on extreme learning. Neural Comput & Applic 32, 15975–15987 (2020). https://doi.org/10.1007/s00521-019-04679-8

[Rajinikanth 20] Rajinikanth, V.; Joseph Raj, A.N.; Thanaraj, K.P.; Naik, G.R. A Customized VGG19 Network with Concatenation of Deep and Handcrafted Features for Brain Tumor Detection. Appl. Sci. 2020, 10, 3429. https://doi.org/10.3390/app10103429

[Gumaste 20] Gumaste P. P, Bairagi. V. K. A Hybrid Method for Brain Tumor Detection using Advanced Textural Feature Extraction .Biomed Pharmacol J 2020;13(1).

[Suresha 20] D. Suresha, N. Jagadisha, H. S. Shrisha and K. S. Kaushik, "Detection of Brain Tumor Using Image Processing," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 844-848, doi: 10.1109/ICCMC48092.2020.ICCMC-000156.