# Project Report - ECE 176

**Tyden Rucker**
Cognitive Science UCSD
A16318316

**Siddhant Joshi**
Cognitive Science UCSD
A16878588

## Abstract

*The advent of deep learning techniques has opened the doors to numerous applications in all kinds of industries. Some important advancements have been made in healthcare, ranging from detection softwares, robot-assisted surgeries, and rapid drug testing. Brain tumor classification has been a popular problem in which deep neural networks (DNN) have made significant progress. We consider techniques from architectures in previous research to offer a holistic review of present and potential methods to detect brain tumors through magnetic resonance imaging (MRI) scans. In this paper, we explore the potential of multi-image classification to implement a generalized model that would be capable of identifying a wide variety of brain tumors as well as healthy brains. We examine modifications to convolutional neural network (CNN) architectures from empirical studies done on the topic to determine the factors that make for accurate and generalized tumor detection tools. According to the results obtained from our testing, we devised a 5-layer CNN that utilizes an Adam optimizer that achieves the best accuracy of 92.93% on test data. Similar neural network architectures are important positive steps toward a future of accurate and efficient diagnoses that would save the lives of millions.*

## 1 Introduction

Brain tumors are one of the most aggressive diseases among the human population with severe impact on the central nervous system (CNS), accounting for 85 - 90% of CNS tumor[2]. Caused by abnormal growth in brain cells, these deadly cancers can be categorized into three common types: glioma, meningioma, and pituitary. Glioma are caused by tumefaction of glial cells, such as astrocytes or ependymal cells. Given the importance of glial cells in maintaining the integrity of the CNS, damage to or caused by these cells pose serious threats for patients, accounting for the highest mortality rate of the tumors[3]. Meningioma originate in the meninges of the brain or spinal cord. This could compromise the protection these layers provide for the CNS in addition to applying enough pressure to damage brain cells. Pituitary tumors form in the pituitary gland, a critical player in development and hormone regulation[7].

A brain tumor's early and accurate detection is critical to devising life-saving treatment plans for the thousands affected every year. This variety of tumor sizes and locations make their diagnosis challenging, even for expert radiologists. Manual analysis of MRI imaging scans is quite time-consuming and leaves ample room for misinterpretation[4]. Coupling the valuable data generated from MRI technology with deep learning image classification algorithms can provide medical professionals with a necessary, automated brain tumor classification system. Automating the analysis of these vast image data sets to detect tumors with high accuracy is a significant step toward proactive oncology. This would both accelerate the diagnosis process as well as eliminate human-error in life-changing treatment plans. In this paper, we discuss and develop preliminary image classification models using staple DNN techniques in CNN architectures. CNNs are ideal for image detection problems that contain a variety of images as diverse as MRI brain scans, as they rely on convolving and pooling pixels to highlight an image's unique features. In the context of brain tumors, this could include the outline of the brain, the tumor's location within it, and the shape of the tumor. These *feature maps* are generated and fine-tuned by the network throughout the training process to be mapped over novel images and provide an accurate classification of the tumor in question[1]. The classifiers we implemented are capable of determining whether a given brain scan is healthy or cancerous as well as identify which kind of tumor is present with accuracies up to 93%.

In 2 Related Work, we will discuss previous literature on the subject, comparing and contrasting various implementations to assess the most effective methods for tackling this problem. We then address the specific architectures we tested in 3 Methods, explaining our rationale behind the adjustments we made as well as utilizing detailed figures to visualize the network layers. Finally, we address the implementation process and subsequent results in 4 Experiments before concluding with an analysis of the whole study and future potential of the produced results.

## 2 Related Work

BrainMRNET, developed by Tocagar et al., used CNN techniques to achieve 96.05% accuracy. This architecture took advantage of residual blocks to preserve details through convolutional layers, a method introduced and utilized in the novel ResNet architecture from He et al., 2015.[12] Khawaldeh et al. suggested an improvement to AlexNet, another effective CNN model composed of 8 layers (5 convolutional layers and 3 fully connected layers), to achieve 91% test accuracy[6]. Researchers M. Siar and M. Teshnehlab also used a baseline model similar to AlexNet and adjusted the classification layers to test for different accuracies. Some of the layers they tested include a Softmax Fully Connected layer, Radial Basis Function (RBF) classifier, and a Decision Tree (DT) classifier. The Softmax fully-connected layer performed the best with an accuracy of 98.67%[8]. These papers provide an excellent context for the problem we aim to tackle, however the devised networks were only trained as binary classifiers. The data we utilize in our implementation contains a variety of brain tumors, capitalizing on practicality while borrowing methods from these papers to preserve efficacy.

In 2022, Kibriya et al. proposed a 13-layer CNN architecture that achieved a highest accuracy of 97.2%. What is unique about this architecture is that it is relatively simple, containing only 13 layers, while being capable of processing data on three kinds of tumors[5]. We will employ this architecture to serve as a point of comparison for the performance of our model, as it is trained to run on multi-class data while striving for conditions similar to ours.

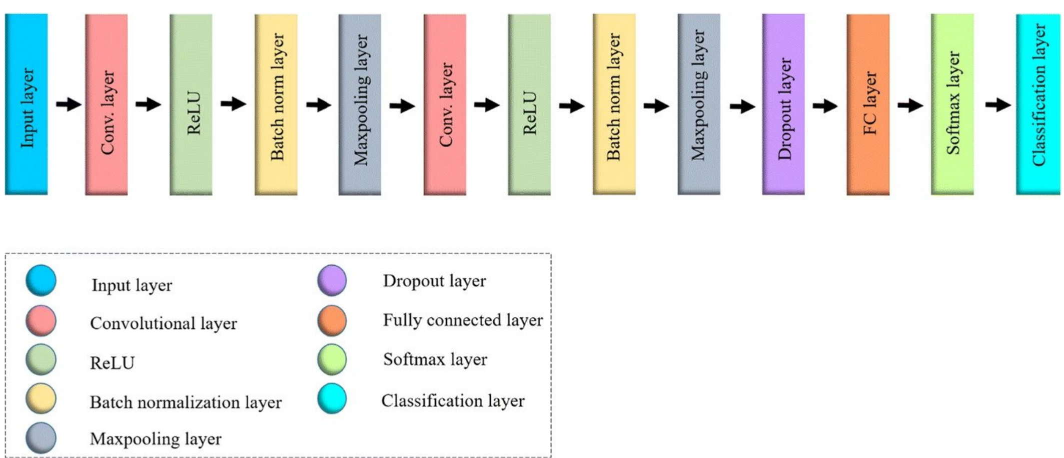


Figure 1: 13-Layer CNN architecture by Kibriya et al. (2022)[5]

## 3 Method

### 3.1 Method Structure

We began with preliminary research into well-performing convolutional neural networks. We discovered a multitude of architectures that achieved high accuracy on both binary classification of brain

| Trial | Epochs | Manipulation |
|---|---|---|
| 1 | 10 | Preliminary 5-layer CNN architecture inspired by baseline proposed by Siar & Teshnehlab, 2019 |
| 2 | 10 | Introduce batchnorm layers after each convolutional layer |
| 3 | 10 | Extra fully-connected layer |
| 4 | 10 | Adam optimizer instead of SGD |
| 5 | 10 | Removed extra fully-connected layer and added a third convolutional-ReLU layer followed by another batchnorm |
| 6 | 10 | Removed third convolutional layer and batchnorm from Trial 5 architecture |

Figure 2: Proposed manipulations across trials

tumors as well as multi-class classification without including a no-tumor class[5][12]. Our proposed model improves on these architectures by including the "no tumor" class to classify brain images as not having cancers as well. Using the preliminary convolutional neural networks, we re-implement the neural network to account for the complexity introduced when including the no-tumor class as well. When constructing and training the neural network from scratch we encountered problems with debugging. After moving on from those problems, we proceeded to add different features and complexity to our model to experiment with the data. We first had a baseline model and used the accuracy it achieved as a reference point moving forward. Moving on to subsequent trials we first added batchnormalization to our model. As batchnormalization had performed well in the models we referenced, we hypothesized that it would help speed up our own model as well. In later trials, we add a convolutional layer or Adam optimizer to experiment with two different manners of improving the model. We theorize that the additional convolutional layer not increasing performance was due to over-fitting of the model. Utilizing the Adam optimizer, we observed a marked improvement in the accuracy of the model and a consistent loss graph. The Adam optimizer mitigated the oscillating of the loss that was occurring in the trials that utilized a SGD optimizer and therefore we utilized the Adam optimizer in subsequent trials. Throughout the trials, our method consisted of finding additions to the neural network that increased accuracy and then taking out features of the neural network that may be contributing to over-fitting or any other forms of accuracy decrease. Throughout the subsequent trials, we notice subtle trends in accuracy that we discuss further in the Results section of this paper. Our trials converged on a model that achieved a high accuracy while being efficient in both design and complexity. By removing the extra layers that did not promote improvement of accuracy whilst also including features that drastically improved it, we were able to reach a high accuracy similar to the accuracy of the reference networks.

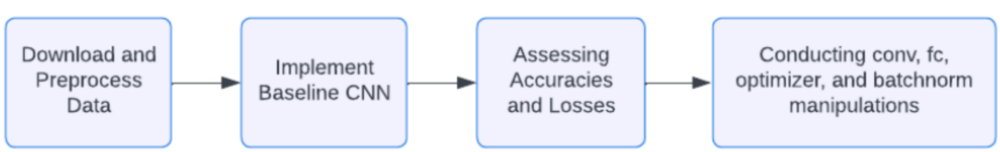


Figure 3: Schematic of proposed methods

### 3.2 Proposed Techniques

Our model builds upon the foundation set by the referenced neural network architectures. The model uses both an additive and subtractive approach to the architecture to maximize efficiency while minimizing complexity. By adding to the architectures and highlighting the main features that contributed to a high accuracy, we were able to maximize the accuracy achieved in the final architecture. As we had removed multiple features of the reference networks that were found inefficient, we were able to reduce the complexity of the neural network and at the same time preserve a majority of the accuracy that was displayed in these referenced neural networks. Moving forward, we propose to make more improvements to the architecture that reduce complexity whilst allowing the model to be generalized to account for the different classes of brain tumor. These techniques offer improvements to the current classification of brain MRI, benefitting radiologists and their patients.

## 4 Experiments

### 4.1 Dataset

We used the Brain Tumor MRI Dataset from Kaggle, which contains a collection of 7,022 grayscale images of MRI scans of patient brains with one of four conditions: non-tumorous, glioma, meningioma, or pituitary tumor. The images are organized in the following distribution of the conditions:

- Non-tumorous; healthy → 2,000 images
- Glioma → 1,621 images
- Meningioma → 1,645 images
- Pituitary → 1,757 images

The images of the data set were not all the same, therefore, during pre-processing, the inputs were normalized to be 64 x 64 pixels and included random vertical/horizontal flips. We also incorporated color jitter to better highlight contrasts in the grayscale images. This data set also enables us to create a multi-class classifier as it has 4 different labels rather than the binary classification networks mentioned in previous literature. Although this would warrant more training, the model would be more practical given the variety of tumor-types it would be able to detect.

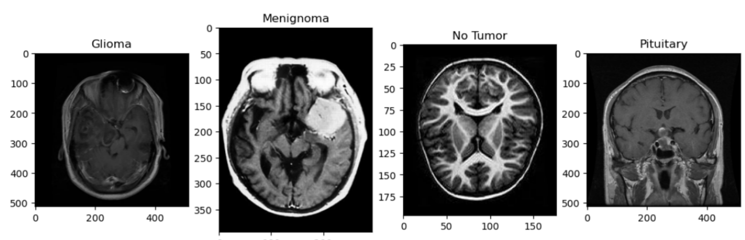


Figure 4: Samples of pre-processed images from each of the 4 classes

### 4.2 Results

When implementing the Kibirya et. al (2022) 13-layer CNN, the model was outputting accuracies around 85 - 90%. We thus aimed to improve on these results by going through the following series of trials that built upon our baseline convolutional neural network while also maintaining lesser complexity and increasing efficiency. A summary of the resulting accuracies are captured in Figure 5 below.

**Trial 1:** Consisted of a base 5-layer CNN with an SGD optimizer, trained over 10 epochs. The CNN was as follows: Conv → ReLU → Conv → ReLU → FC. After training and testing this architecture, we generated a basis upon which we would aim to improve through later adjustments. The average accuracy was 80.5 percent for this 5-layer CNN.

| Trial | Accuracy |
|---|---|
| 1 | 80.50% |
| 2 | 91.43% |
| 3 | 92.06% |
| 4 | 92.11% |
| 5 | 91.06% |
| 6 | 92.93% |

Figure 5: Summary of average accuracies across trials

**Trial 2:** Batchnorm layers were introduced after every convolutional layer. Using Batchnorm is a typical preliminary step toward increasing efficiency of the model as it stabilizes the model with normalizations after every input layer. Additionally, this stabilization speeds up processing time for the model by increasing the rate of convergence. We saw a jump in accuracy to 91.43%.

**Trial 3:** In trial 3 we added a second fully-connected layer to the architecture's classification layers in an attempt to increase the accuracy akin to the manner in which the AlexNet utilizes stacked fully-connected layers. This achieved 92.06% accuracy, a slight improvement from Trial 2.

**Trial 4:** While fine-tuning the model, we decided to remove the extra fully-connected layer in exchange for the incorporation of an Adam optimizer. The Adaptive Moment Estimator (Adam) utilizes momentum to compute the loss while tuning the learning rate to follow suit. Compared to Stochastic Gradient Descent (SGD) which experienced an oscillating loss decay (Figure 6). This more adaptive method of Adam's loss minimization also results in a more steady loss decay.

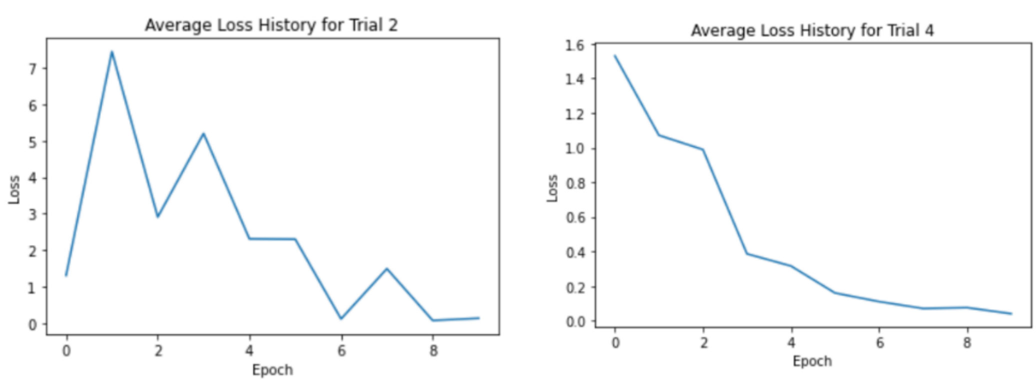


Figure 6: Comparison of SGD and Adam loss decay

**Trial 5:** By adding a 3rd convolutional layer to the model, we hoped to forgo a slight increase in complexity for a greater increase in accuracy. While training this model, we did not see marked improvement in the results, achieving an accuracy of only 91.06%.

**Trial 6:** Our last trial achieved the highest accuracy at 92.93 percent. This was achieved by removing both batchnormalization and the third convolutional layer. We decided that batchnormalization, while speeding up the model, hindered accuracy. Additionally, this would decrease complexity while maximizing accuracy. We believed the third convolutional layer had a decreased accuracy due to over-fitting to the data while contributing to the overall complexity as well. In order to maximize accuracy while also minimizing complexity and minimizing the amount of layers, we opted for taking these aforementioned layers out.

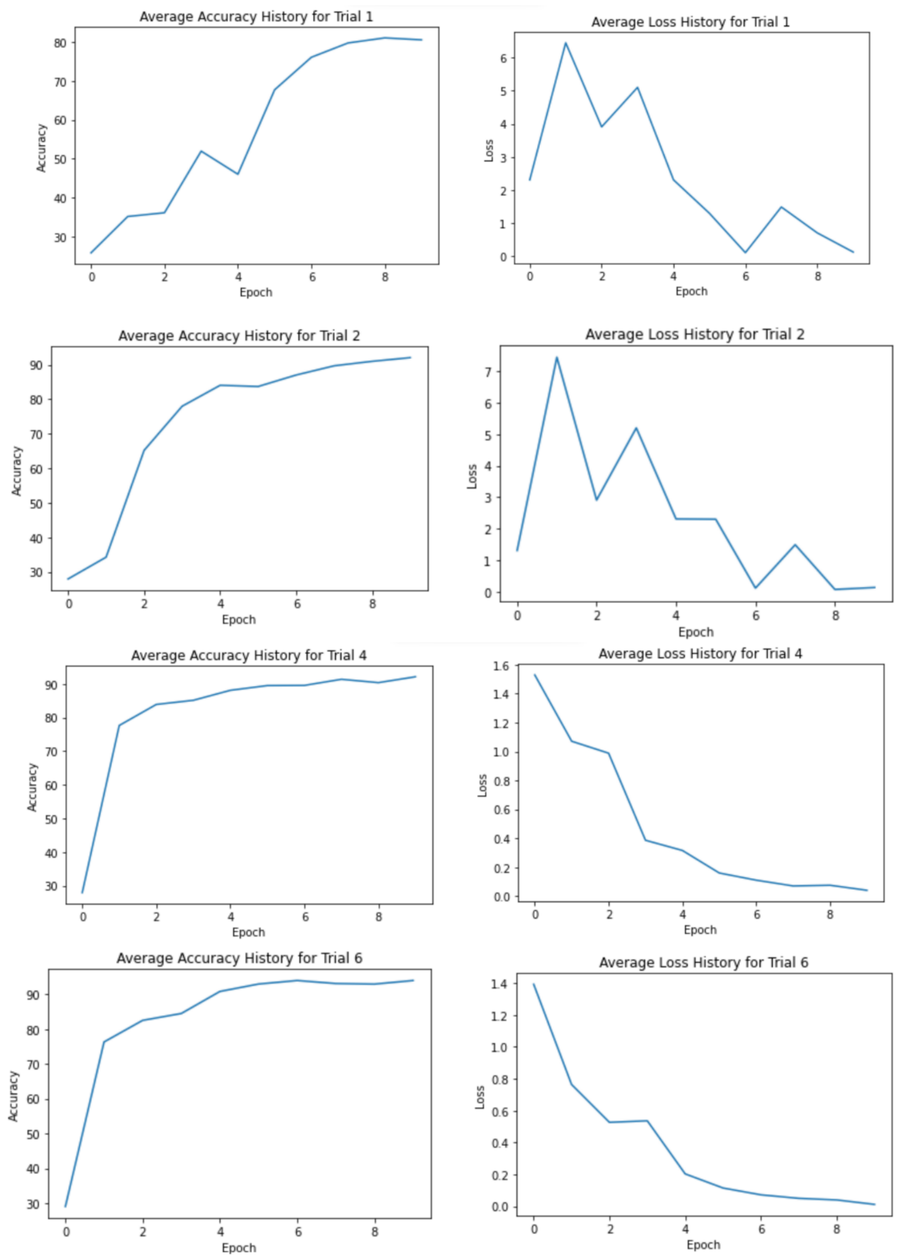


Figure 7: Graphs of select trials

### 4.3 Analysis and Conclusions

One of the most notable observations during our analysis of the results was that Batchnorm, while effective at decreasing runtime, did not benefit the model in the context of accuracy. Given this, we removed it from the model and opted for the increase in accuracy over increase in runtime. We also experienced diminishing returns when adding the extra convolutional layers. We theorize this was due to our model over-fitting to the training data. Convolutional layers, while effective at highlighting important image features, can end up mapping their feature maps too closely to training data. Therefore, we settled with two convolutional layers as three or beyond ended up decreasing accuracy. After the many adjustments made over the course of the six trials, we were able to increase accuracy to 93% while maintaining an accuracy comparable to that of Kibriya et al. with half of the number of layers. As seen in the graphs in Figure 8, we were able to relatively mimic the accuracy progressions shown in the 13-layer CNN of Kibriya et al. with less layers involved. Our loss decay also is very similar to the graph of the loss for the 13-layer CNN. We conclude that using our tuning methods and setup we were able to maximize accuracy while minimizing complexity.

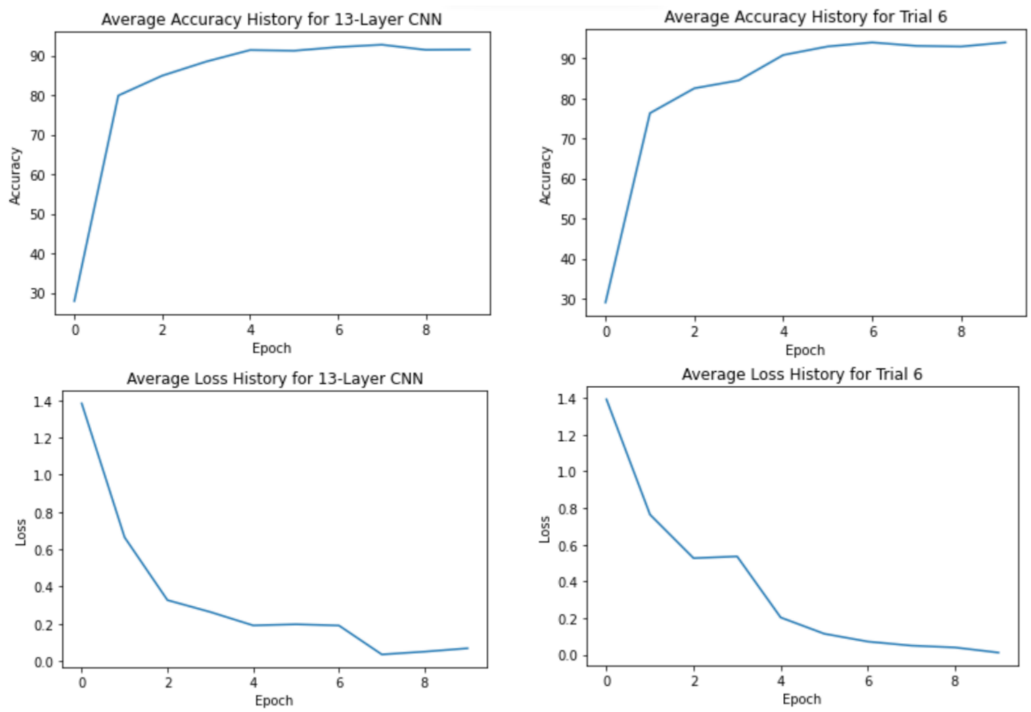


Figure 8: Cross-comparison of loss and accuracy progression between Kibriya et al. 2022 13-layer CNN performance and our best-performing 5-layer CNN

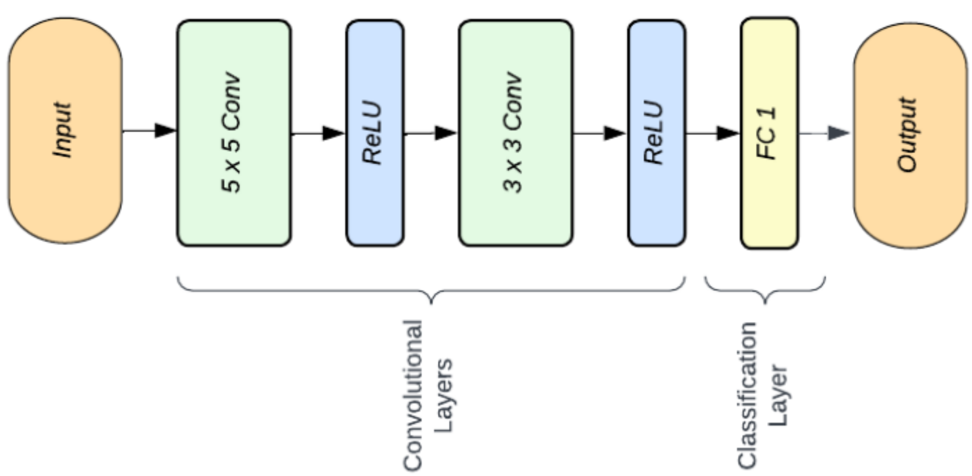


Figure 9: CNN that performed the best

## 5 Supplementary Material

We recorded a video to provide further explanations for the selection of previous literature, experimental methods, and results. We also share some conclusions about the results while discussing some analysis for the techniques utilized in the CNN architectures proposed in the paper.

## 6 References

[1] Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: Proc IEEE International Conference on Engineering and Technology (ICET), Antalya, Turkey, pp 1–6

[2] Brain Tumor - statistics. Cancer.Net. (2023, March 9). Retrieved March 22, 2023, from https://www.cancer.net/cancer-types/brain-tumor/statistics

[3] DeAngelis LM (2001) Brain tumors. N Engl J Med 344(2):114–123

[4] D. V. Gore and V. Deshpande, "Comparative Study of various techniques using Deep Learning for Brain Tumor Detection," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154030.

[5] Kibriya, H., Masood, M., Nawaz, M. et al. Multiclass classification of brain tumors using a novel CNN architecture. Multimed Tools Appl 81, 29847–29863 (2022). https://doi.org/10.1007/s11042-022-12977-y

[6] Khawaldeh, S., Pervaiz, U., Rafiq, A., & Alkhawaldeh, R (2017). Noninvasive Grading of Glioma Tumor Using Magnetic Resonance Imaging with Convolutional Neural Networks. Applied Sciences, 8(1), 27. https://doi.org/10.3390/app8010027

[7] Komninos J et al (2004) Tumors metastatic to the pituitary gland: case report and literature review. J Clin Endocrinol Metab 89(2):574–580

[8] M. Siar and M. Teshnehlab, "Brain Tumor Detection Using Deep Neural Network and Machine Learning Algorithm," 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2019, pp. 363-368, doi: 10.1109/ICCKE48569.2019.8964846.

[9] Msoud Nickparvar. (2021). <i>Brain Tumor MRI Dataset</i> [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/2645886

[10] Mzoughi, H., Njeh, I., Wali, A. et al. Deep Multi-Scale 3D Convolutional Neural Network (CNN) for MRI Gliomas Brain Tumor Classification. J Digit Imaging 33, 903–915 (2020). https://doi.org/10.1007/s10278-020-00347-9

[11] Sultan H.H., Salem NM, Al-Atabany W (2019) Multi-classification of brain tumor images using deep neural network. IEEE Access 7:69215–69225

[12] Toğaçar M, Ergen B, Cömert Z (2020) BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model. Med Hypotheses 134:109531