

# BRAIN TUMOR CLASSIFICATION AND PREDICTION USING CONVOLUTIONAL NEURAL NETWORKS

CAPSTONE THREE: PROJECT REPORT  
DATA SCIENCE CAREER TRACK  
SPRINGBOARD

SUBMITTED BY: SHAHJAHAN AHMED  
DATE: DECEMBER 31, 2021

## Introduction:

A brain tumor is a collection, or mass of abnormal cells in our brain. Our skull, which encloses our brain, is very rigid. Any growth inside such a restricted space can cause problems. Brain tumors can be cancerous (malignant) or noncancerous (benign). When benign or malignant tumors grow, they can cause the pressure inside our skull to increase. This can cause brain damage, and it can be life-threatening. Early detection and classification of brain tumors is an important research domain in the field of medical imaging and accordingly helps in selecting the most convenient treatment method to save patients life.

## 1. Problem Identification

### 1.1 Problem statement formation

Brain Tumor Classification and Prediction using Convolutional Neural Networks to help automate the diagnostic process which will ensure proper treatment and will save lives and resources.

### 1.2 Context

Brain Tumors are complex. There are a lot of abnormalities in the sizes and location of the brain tumor(s). This makes it difficult for complete understanding of the nature of the tumor. Also, a professional Neurosurgeon is required for MRI analysis. Often, in developing countries the lack of skillful doctors and lack of knowledge about tumors makes it really challenging and time-consuming to generate reports from MRI'. So, an automated system on Cloud can solve this problem.

### 1.3 Criteria for success

The objective of this project is to classify whether a patient has a brain tumor or not by using deep neural networks. The main criteria will be to have a high recall value so that there are minimum false negatives and maximum true positive rates. Accurately identifying the tumor in the initial stage will have a chance to get proper treatment, and finally will save lives.

### 1.4 Scope of solution space

The main aim of this project is to build a classifier model by using Convolutional Neural Network which will be used by Doctors, Hospitals, and Diagnostics Centers for their MRI examination report to identify tumors more accurately.

### 1.5 Constraints

The dataset contains only seven thousand images of four different tumor types. For deep learning projects the volume of data is the main key for model performance because the neural network will fetch more features as the size of the dataset increases. Also, the dataset contains RGB high resolution images. So, more computational resources will be required to properly tune the hyperparameter of the models.

## 1.6 Stakeholders

Doctors, Hospitals, Medical Centers, and Patients.

## 1.7 Data sources

I have collected the data from Kaggle and from other websites, and it contains the MRI images of patient's medical records. The dataset contains four different tumor classes including Glioma Tumor, Meningioma Tumor, Pituitary Tumor, and No-tumor.

# 2. Data Wrangling and Exploratory Data Analysis

## 2.1 Dataset Summary:

The dataset contains four separate classes of brain MRI images: Glioma Tumor, Meningioma Tumor, Pituitary Tumor, and Without presence of tumor (No Tumor). The total number of images is more than seven thousand and I have divided the dataset into training set, validation set, and test set with a 70:15:15 ratio. Each of the training, validation, and test dataset contains all four classes. The characteristics of each tumor class is described below:

### 2.1.1 Meningioma Tumor

A meningioma is a tumor that arises from the meninges — the membranes that surround our brain and spinal cord. Although not technically a brain tumor, it is included in this category because it may compress or squeeze the adjacent brain, nerves and vessels. Meningioma is the most common type of tumor that forms in the head. Most meningiomas grow very slowly, often over many years without causing symptoms. But sometimes, their effects on nearby brain tissue, nerves or vessels may cause serious disability.

### 2.1.2 Glioma Tumor

Glioma is a type of tumor that occurs in the brain and spinal cord. Gliomas begin in the gluey supportive cells (glial cells) that surround nerve cells and help them function. Three types of glial cells can produce tumors. Gliomas are classified according to the type of glial cell involved in the tumor, as well as the tumor's genetic features, which can help predict how the tumor will behave over time and the treatments most likely to work. A glioma can affect your brain function and be life-threatening depending on its location and rate of growth.

### 2.1.3 Pituitary Tumor

Pituitary tumors are abnormal growths that develop in your pituitary gland. Some pituitary tumors result in too much of the hormones that regulate important functions of your body. Some pituitary tumors can cause your pituitary gland to produce lower levels of hormones. Most pituitary tumors are noncancerous (benign) growths (adenomas). Adenomas remain in your pituitary gland or surrounding tissues and don't spread to other parts of your body.

### 2.1.4 No Tumor

The MRI image does not contain any kinds of tumor cell.

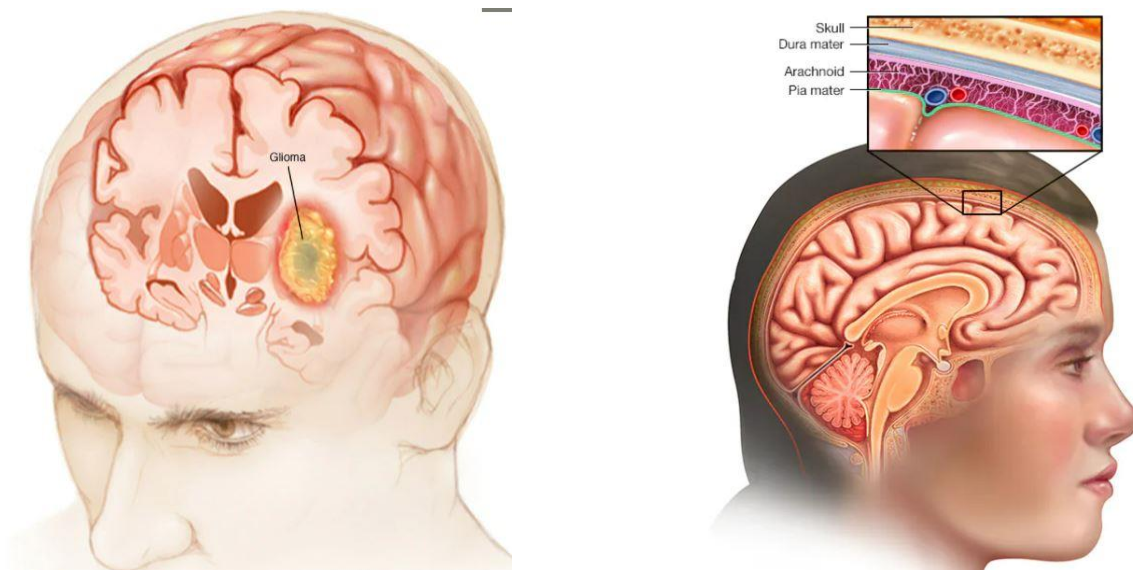


Figure: Images of Glioma (left), Meningioma (right)

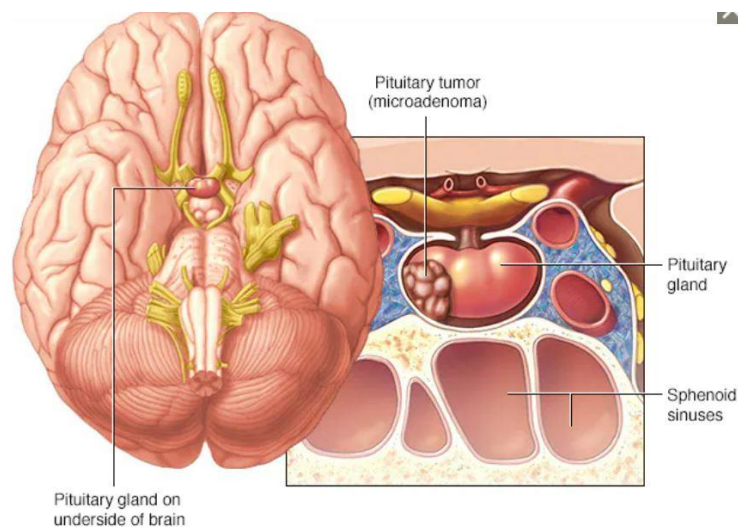


Figure: Pituitary Tumor

## 2.2 Data Wrangling

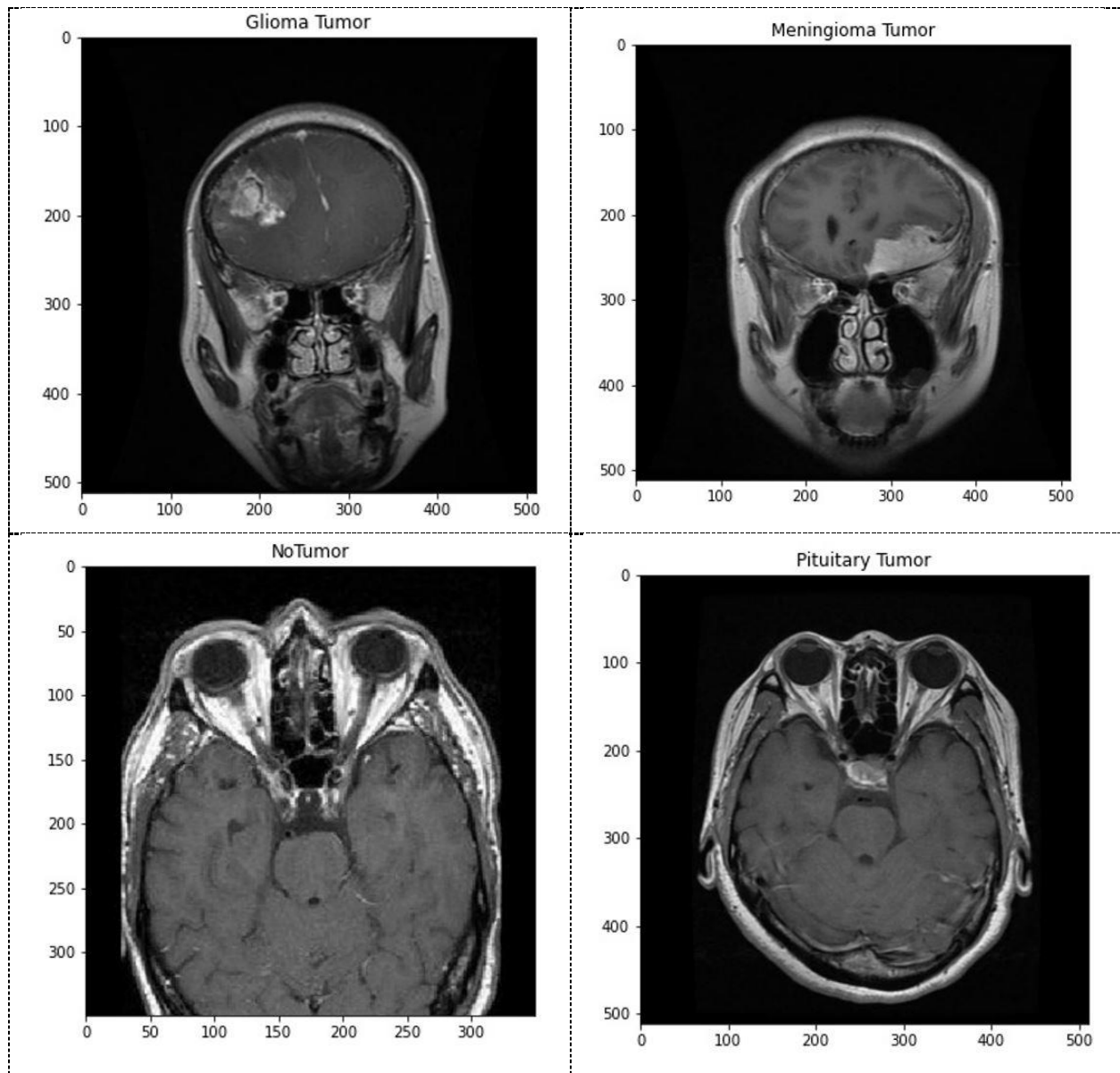
The steps I have taken to wrangle the dataset is described below:

- I. I have Imported all necessary modules and libraries
- II. Observed all the MRI images in training, validation, and test set with their respective file format.
- III. Observed whether the dataset contain RGB or grayscale images

- IV. Counted the number of images of each set
- V. Applied different filtering techniques to improve the image quality
- VI. Observed the shape of each image and found all the images of RGB with different shape
- VII. I have converted all the images into gray scale image of size (128, 128, 1).

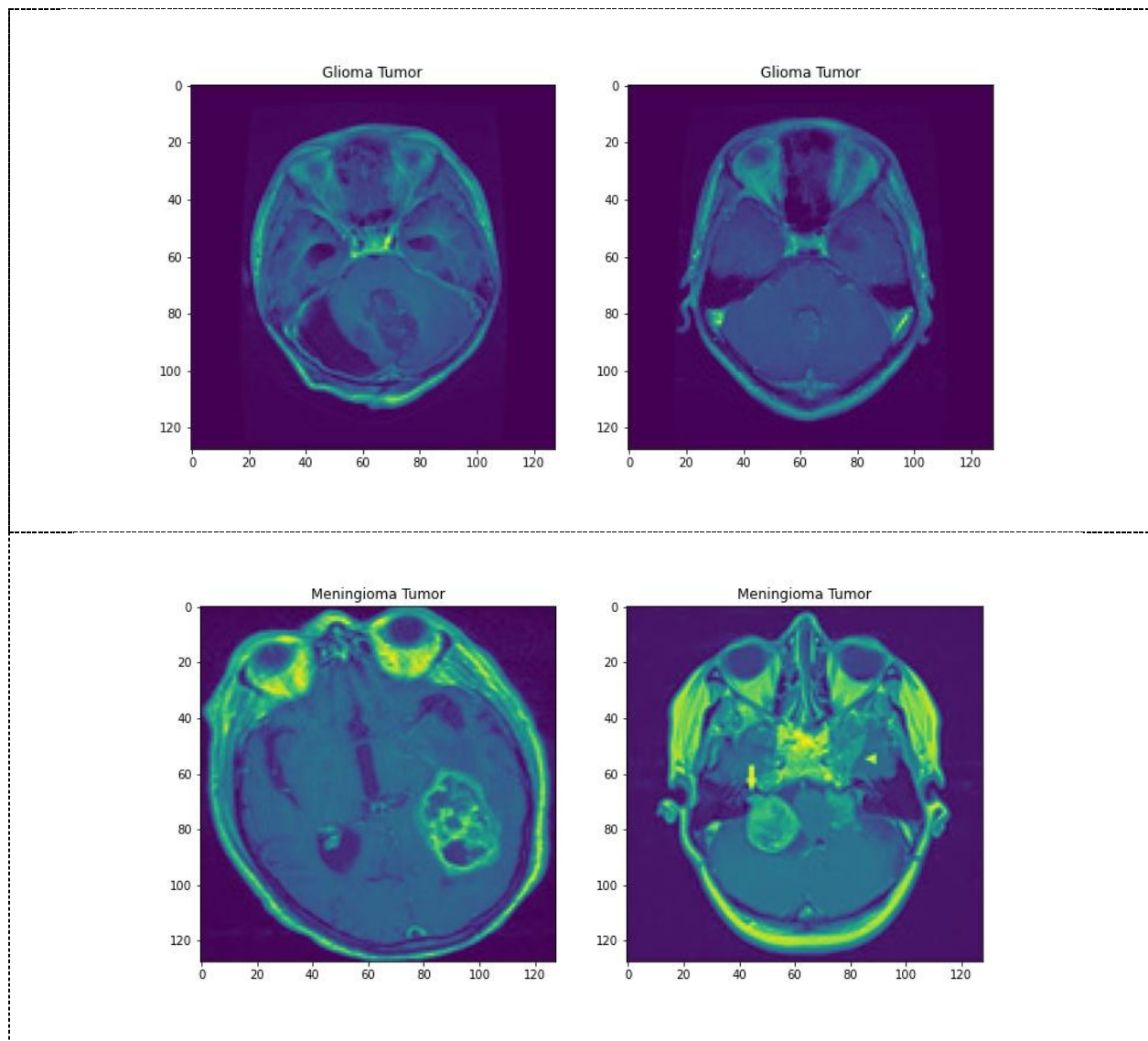
### 2.3 Exploratory Data Analysis

The initial MRI images were RGB image with variable shapes. The figure below depicted the original MRI images of four different tumor classes.



Figures: The original images

I have converted all the images from RGB to gray and resized them into (128,128) shape. The figure below shows the converted and resized images.



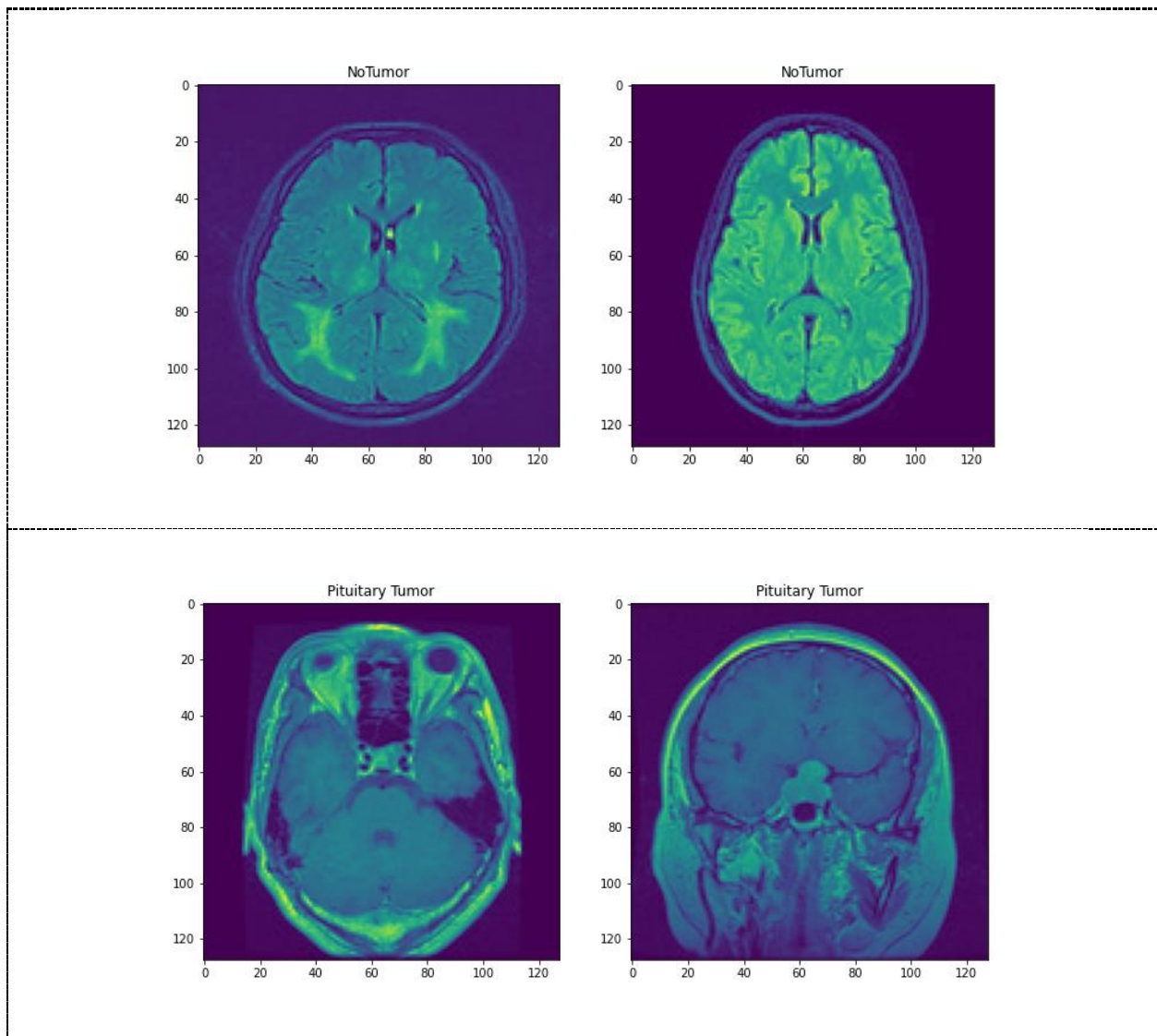
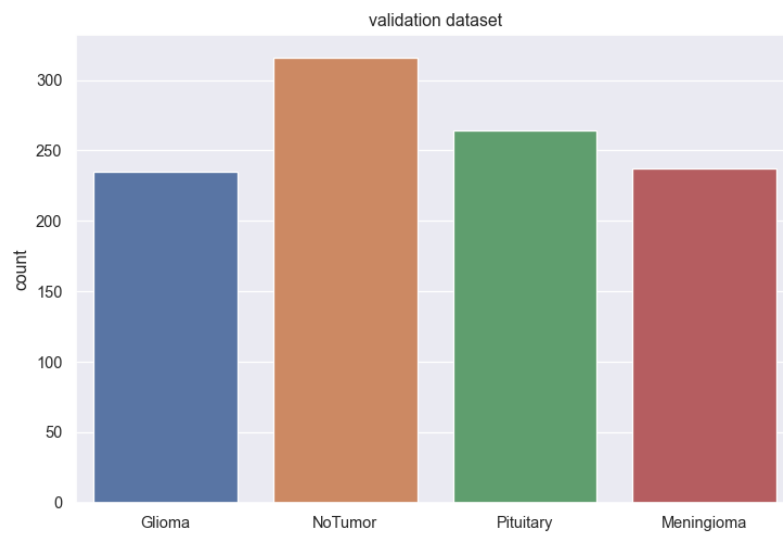
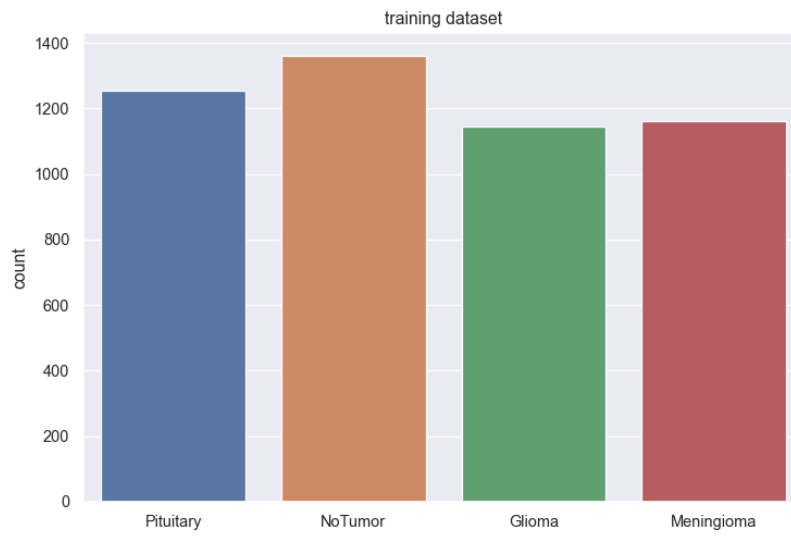


Figure: Resized gray scale images

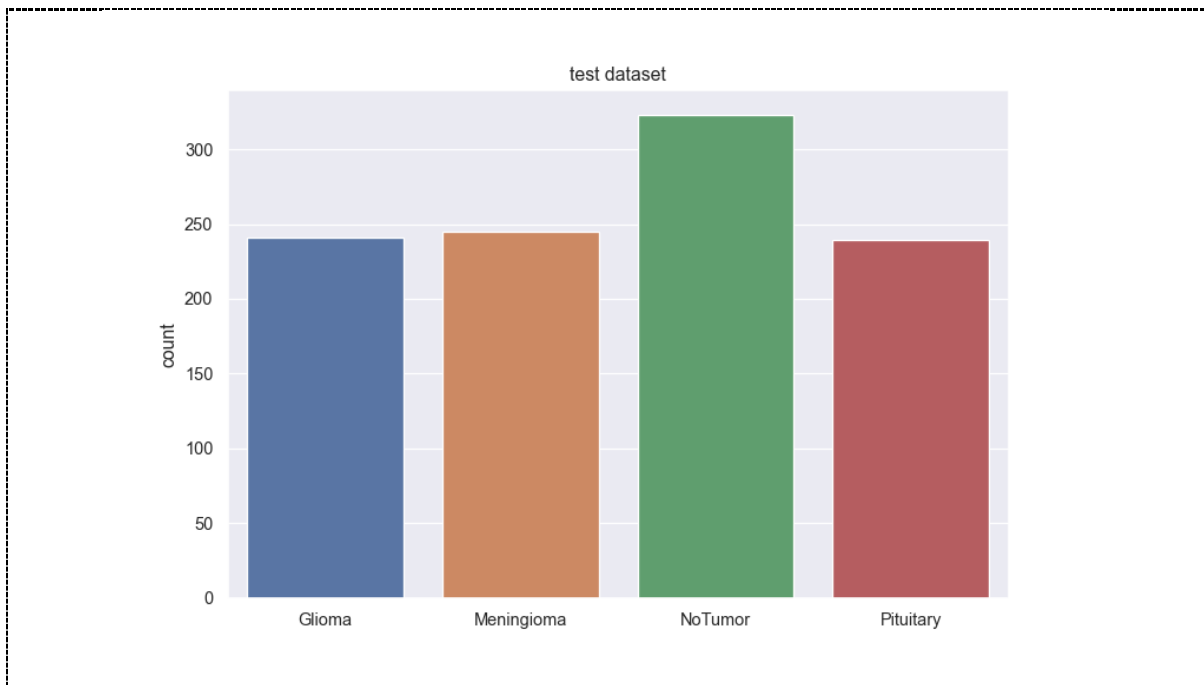
### 3. Data Preprocessing

The steps I have taken to preprocess all the training, validation and test dataset is given below:

- I. I have Created image arrays and corresponding labels for training, validation, and test dataset
- II. The image arrays for feature vector are respectively  $X_{train}$ ,  $X_{valid}$ ,  $X_{test}$ , and the corresponding labels are  $y_{train}$ ,  $y_{valid}$ , and  $y_{test}$
- III. Then, I have Created one-hot vectors for  $y_{train}$ ,  $y_{test}$ , and  $y_{valid}$
- IV. Figure below shows the count plot of the final training, validation, and test dataset.







Figures: Count plot of each dataset

#### 4. Modeling

To classify and detect brain tumor, I have implemented and trained several models and evaluated them by using test data to measure models performance.

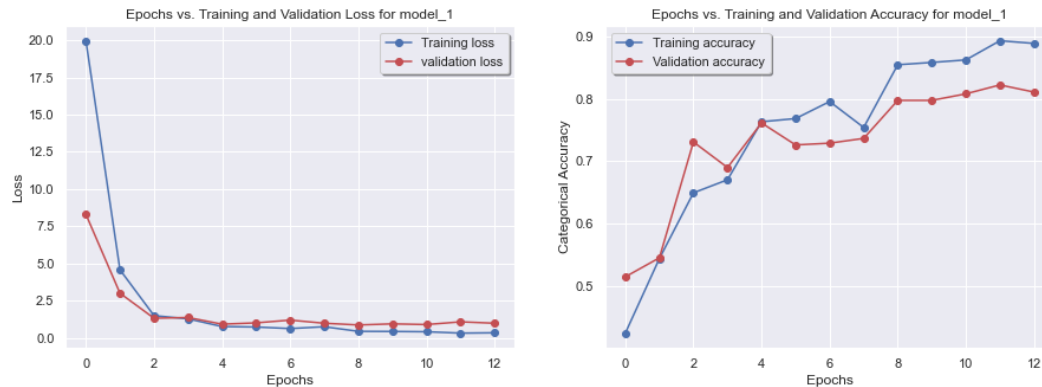
##### **Model 1: Simple neural network with only dense layers**

Architecture of Model 1:

The model 1 contains an input layer, six Dense layers, and one output layer. I have used ReLU as an activation function for the Dense layers and Softmax as an activation function for the output layer for multiclass classification and Adam as an optimizer.

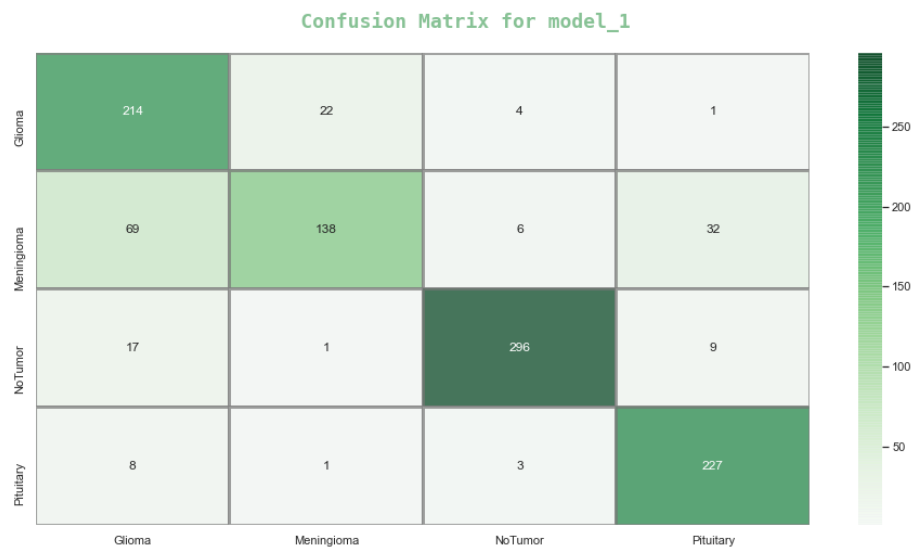


## Training Progress for Model 1:



## Evaluation Results of Model 1:

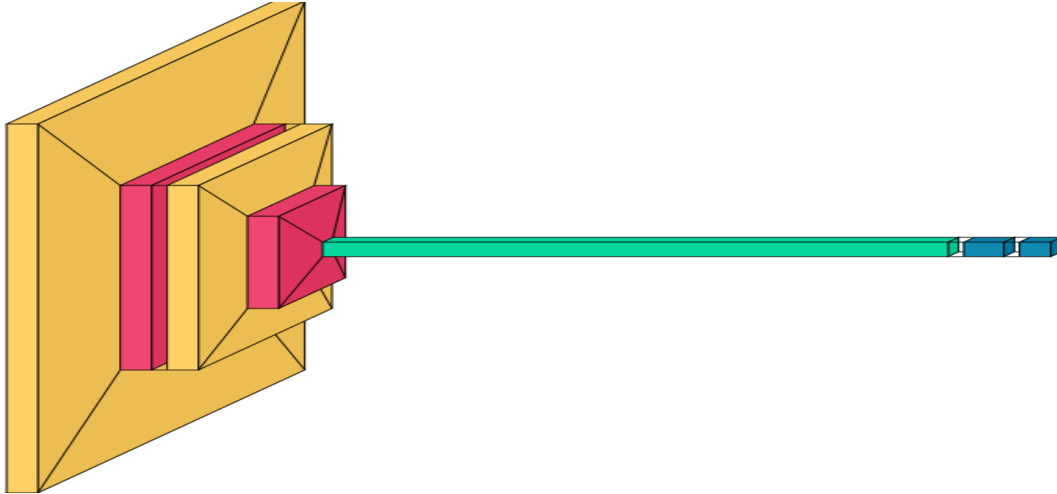
	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 1	0.89	0.822	0.834	0.85	0.83	0.83



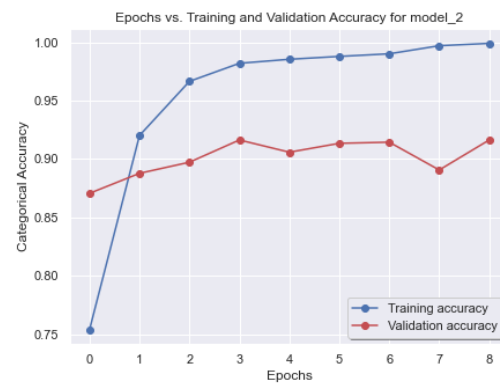
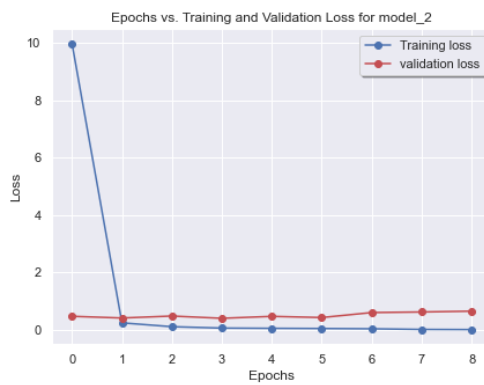
## Model 2: Convolutional neural network with two Conv2D layers

### Architecture of Model 2:

Model 2 contains an input layer, two Conv2D layers with MaxPolling2D as a pooling layer, one Dense layer, and an output layer. Similar to model 1, I have used ReLU as an activation function for the hidden layers, Softmax as an activation function for the output layer, and Adam as an optimizer again. I have not used any regularization in model 2.

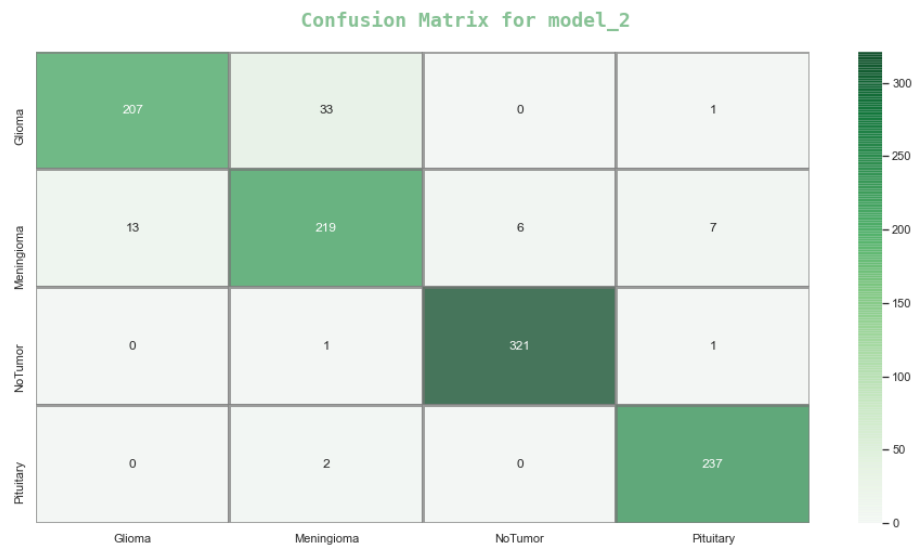


## Training Progress of Model 2:



## Evaluation Results of Model 2:

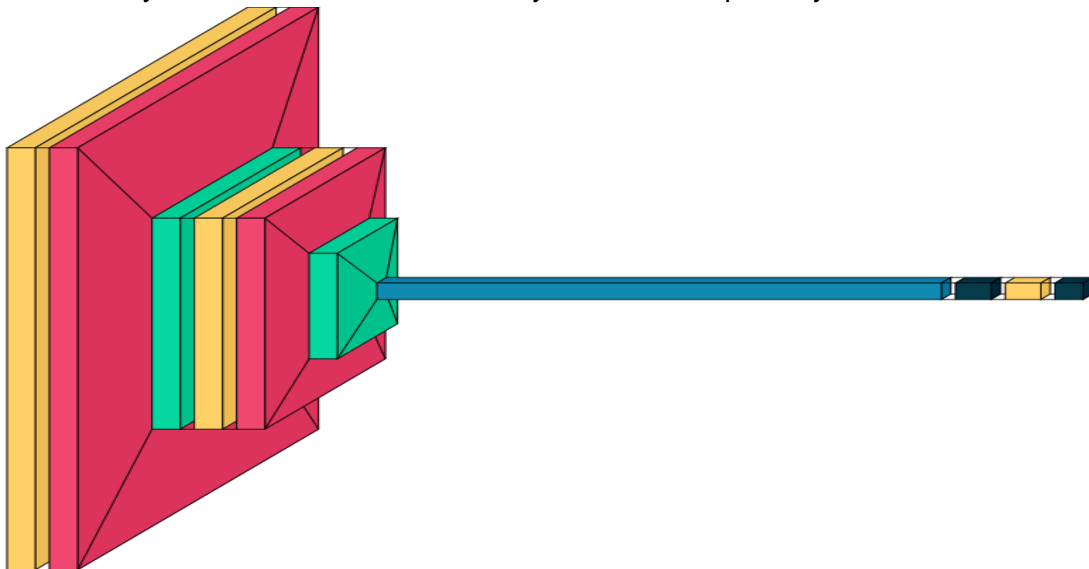
	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 2	0.99	0.916	0.938	0.94	0.94	0.94



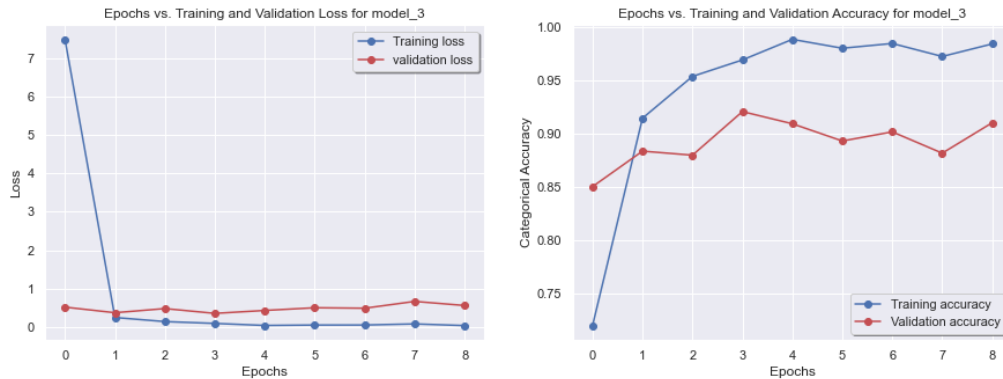
### Model 3: Convolutional neural network with two Conv2D layers with regularization layers

Architecture of Model 3:

Model 3 is a copy of model 2. To overcome the overfitting caused by model 2, I have added BatchNorm2D layer after each convolutional layers and a Dropout layer.

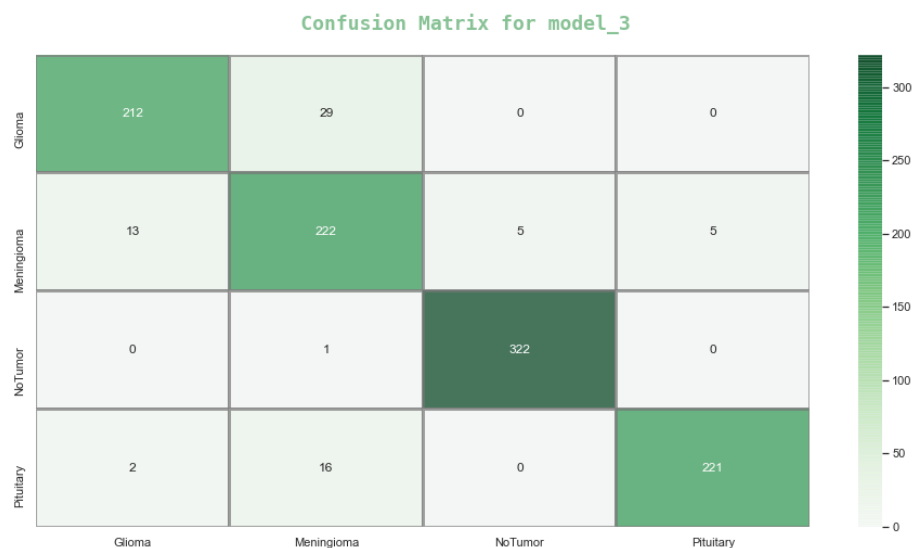


## Training Progress of Model 3:



## Evaluation Results of Model 3:

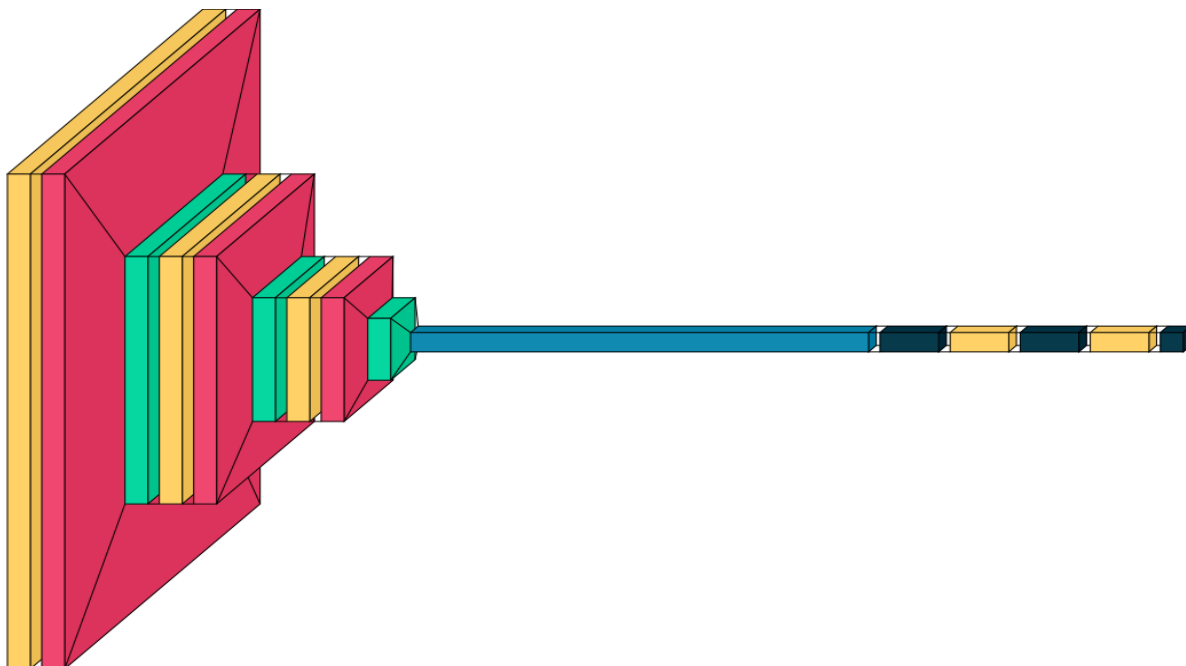
	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 3	0.98	0.921	0.932	0.93	0.93	0.93



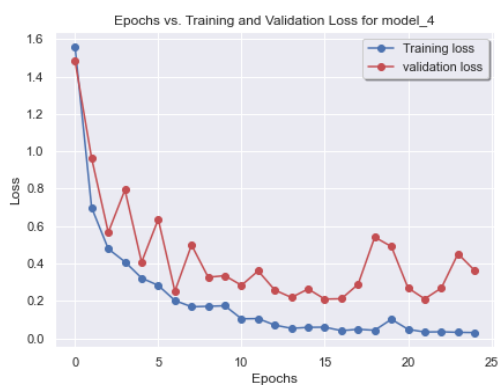
## Model 4: Convolutional neural network with three Conv2D layers and regularization layers

### Architecture of Model 4:

Model 4 contains an input layer, three Conv2D layers with MaxPooling2D and BatchNorm2D, two Dense layers, and an output layer. The activation function, optimizer remains same with previous models. I have introduced learning rate scheduling and EarlyStopping for adaptive training.

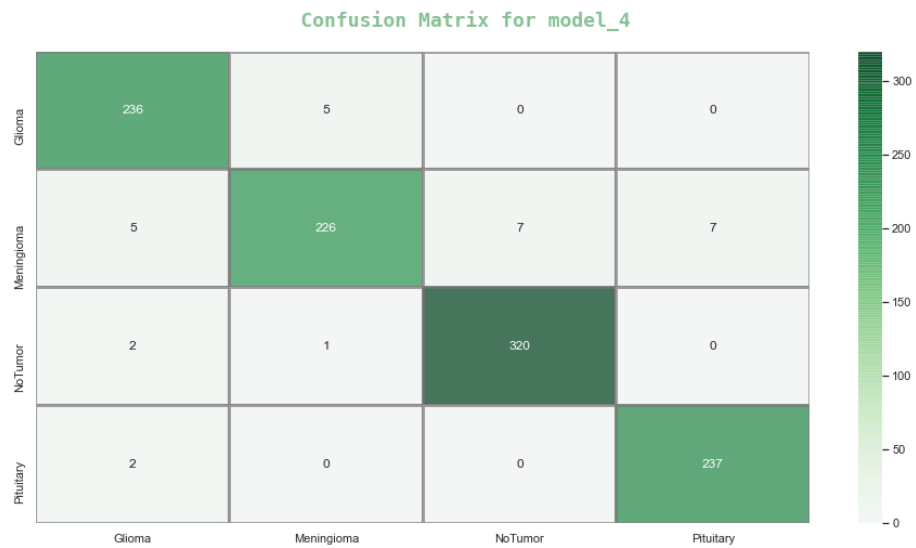


Training Progress of Model 4:



Evaluation Results of Model 4:

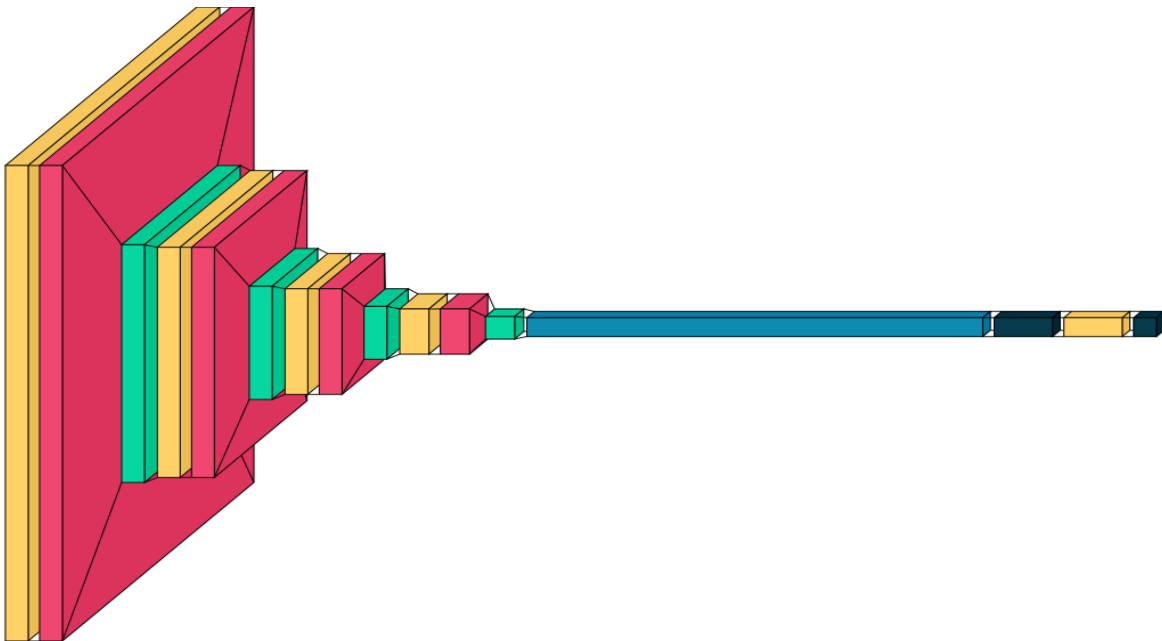
	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 4	0.99	0.955	0.972	0.97	0.97	0.97



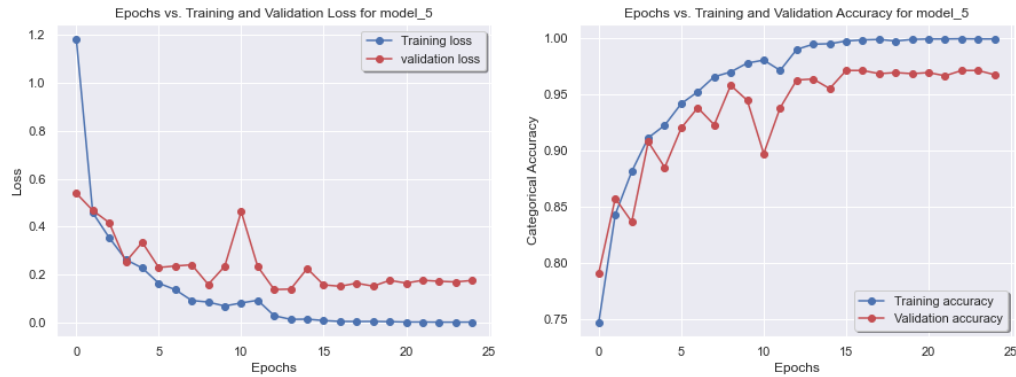
## Model 5: Convolutional neural network with four Conv2D layers and regularization layers

Architecture of Model 5:

Model 5 contains an input layer, four Conv2D layers with MaxPolling2D and BatchNorm2D, one Dense layer, and an output layer. The activation function, optimizer remains same with previous models. I have introduced learning rate scheduling and EarlyStopping for adaptive training.

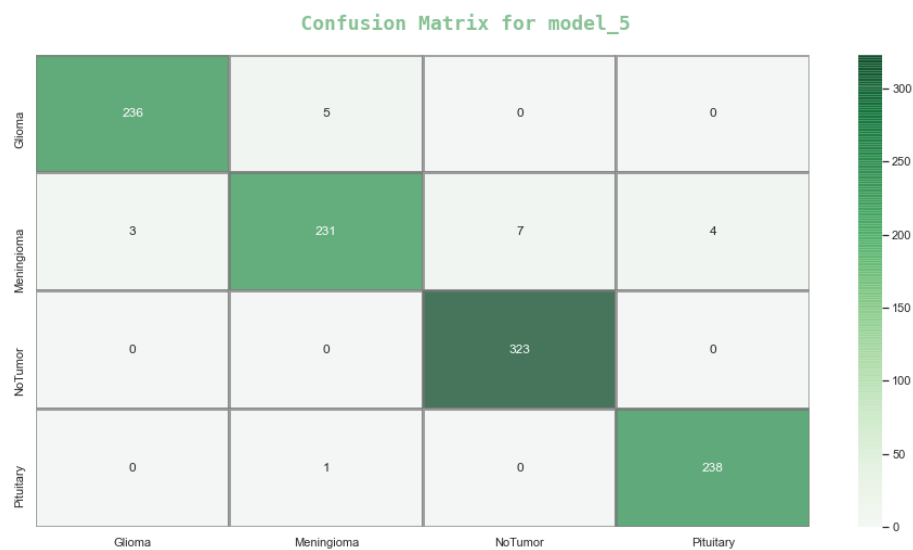


## Training Progress of Model 5:



## Evaluation Results of Model 5:

	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 5	0.99	0.972	0.981	0.98	0.98	0.98

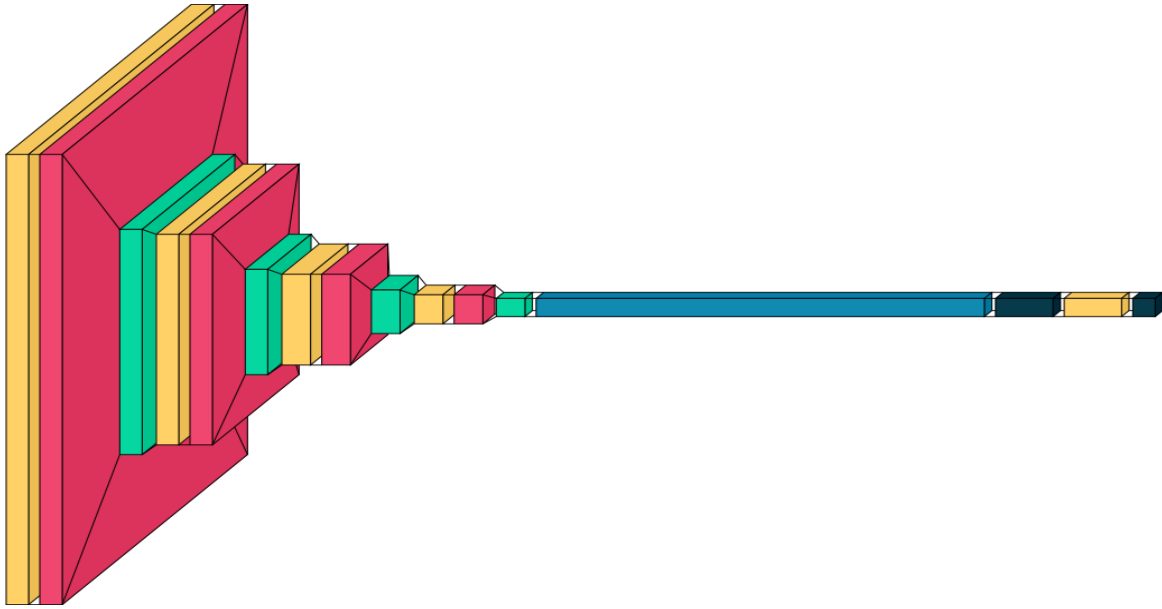


## Model 6: Implementing Hyperparameter Tuning of previous model (Model 5)

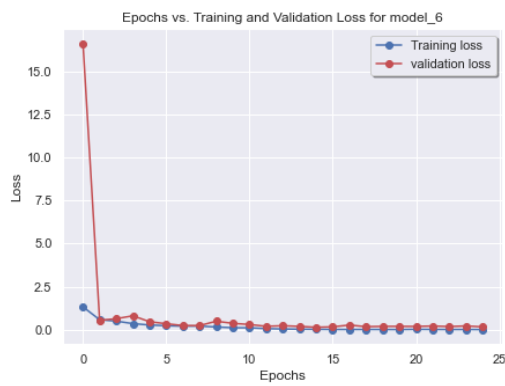
### Architecture of Model 6:

Model 6 is a copy of the model 5. I have changed some hyperparameter (number of filters in each convolutional layer, filter size, the optimizer, value of dropout) and evaluated the model metrics to compare with the previous model (model 5).



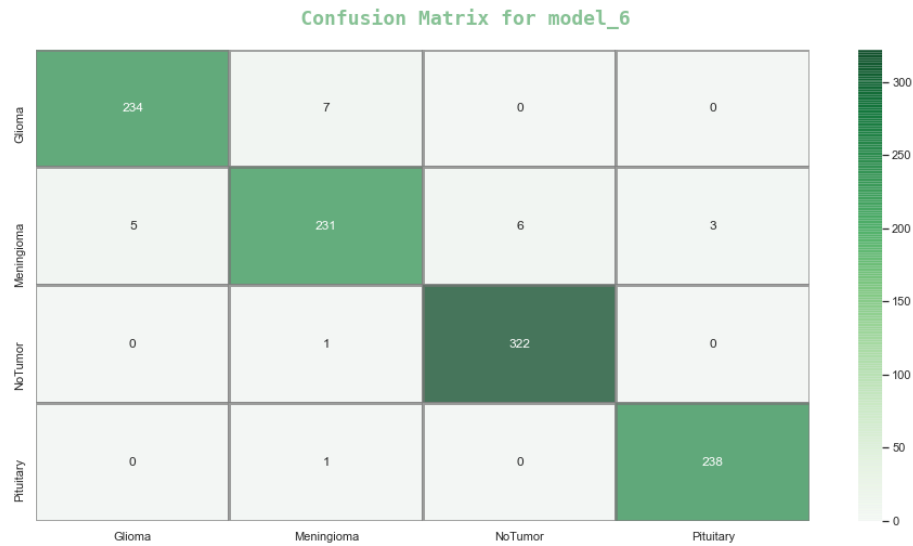


Training Progress of Model 6:



Evaluation Results of Model 6:

	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 6	0.99	0.975	0.978	0.98	0.98	0.98



### Evaluation Metrics:

I have used several metrics namely accuracy, precision, recall, f1-score, and confusion-matrix to evaluate the performance of each model. Since the aim of my model is to predict and classify brain tumor accurately, I have used recall as my main model metrics among all the metrics I have used. More precisely, I have wanted to implement a model which will predict less false negatives.

## 5. Results and Findings

Initially, I have implemented first model (model 1) without using any convolutional neural network and received 83.49% test accuracy and 83% recall value. After that, I have implemented model 2 with two Conv2D layers. Model 2 performed better than first model but had strong overfitting problem since I haven't added any regularization techniques. I have introduced Batch Normalization and dropout layers to the previous model and implemented model 3 to overcome overfitting. Despite addition of regularization techniques, the evaluation metrics was unchanged, and the model's training and test accuracy was not improving. To overcome underfitting, I have implemented complex convolutional neural networks: model 4 with three Conv2D layers and Model 5 with four Conv2D layers. I have received test accuracy and recall value of 97% from model 4 and there was a fluctuation with training accuracy and validation accuracy also. Finally, the model 5 performed great compared with other models and I have received overall training accuracy, validation accuracy, test accuracy, and the weighted value of precision, recall, and F1-score of 98%. Also, by inspecting confusion matrix of model 5, I have observed that out of 1048 test images there is only 20 misclassifications.

	Training accuracy	Validation Accuracy	Test Accuracy	Weighted Average		
				Precision	Recall	F1-Score
Model 1	0.89	0.822	0.834	0.85	0.83	0.83
Model 2	0.99	0.916	0.938	0.94	0.94	0.94
Model 3	0.98	0.921	0.932	0.93	0.93	0.93
Model 4	0.99	0.955	0.972	0.97	0.97	0.97
Model 5	0.99	0.972	0.981	0.98	0.98	0.98
Model 6	0.99	0.975	0.978	0.98	0.98	0.98

Table: Comparison of model's evaluation metrics

## Conclusion

Despite lack of proper computational resources, I have implemented, trained, and tested all my models and obtained good evaluation results. The optimum model (model 5) showed training accuracy, validation accuracy, test accuracy, and the weighted value of precision, recall, and F1-score of 98% and contained only 20 misclassifications out of 1048 test images.