**Brain Tumor Detection Classification:**

**INTRODUCTION:**

Cancer is one of the most deadly disease that we have faced today. Detecting the tumour can serve as potential indicators of cancer, which is crucial in our fight against this formidable advisory. As AI tools continue to evolve and find applications in medicine and image diagnostics, they become valuable allies in tumour detection alongside healthcare professionals. However, it is essential to emphasize that AI should not replace professional diagnosis. Instead, it serves as a supportive tool, assisting clinicians in their assessments.

In this project we have used CNN (**Convolutional Neural Networks**) for detecting whether the brain images has tumour in it or not.

**OBJECTIVE:**

The main motto of this project is to develop a Convolutional Neural Networks (**CNN**) that will help in detecting whether the brain images provided has tumour in it or not.

**PROBLEM STATEMENT:**

Cancer is one of the most deadly disease that we have faced today. Detecting the tumour can serve as potential indicators of cancer, which is crucial in our fight against this formidable advisory. Therefore, to detect the cancer, we aim to develop the CNN (**Convolutional Neural Networks**) for detecting whether the brain images has tumour in it or not.

**DATASET:**

The brain tumour has two sets of folder ‘No’ and ‘Yes’.

1. **No Folder** = It contains the list of all the images that are not detected with the brain tumour. There are total 98 brain images that are not detected with the brain tumour.
2. **Yes Folder =** It contains the list of all the brain images that are detected with the brain tumour. There are total 155 brain images that are detected with the brain tumour.

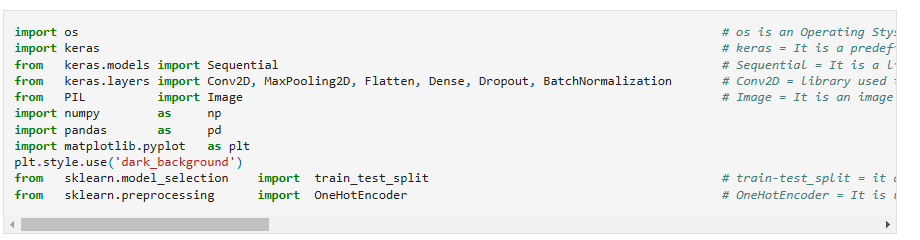
**ALGORITHM:**

1. Creating three Important Lists for storing the data.
   1. **Data** - It contains all the data in the numpy array form.
   2. **Paths** - It contains the paths where the images are to be stored.
   3. **Result** - It contains the result of whether the brain image has a tumour or not.

1. The dataset is splitted into training and testing data where the training data has 80% data and the testing data has 20% data.
2. Then a Convolutional Neural Network is developed where Batch Normalization is used. This Batch Normalization is used to stabilize the learning process by splitting the inputs into layers. It is also used to reduce the number of epochs for every learning processes.
3. The hyperparameters used in this project is of below:
   1. Input : 128 \* 128 \* 3
   2. Number of epochs : 30
   3. Batch size : 40
   4. verbose : 1
4. Project Results
   1. The model used on MRI images test set provides the following result:
   2. Accuracy for Non Tumor Dataset : (**100% Accuracy**)
   3. Accuracy for Tumor Dataset : (**61.17897629737854% Accuracy**)

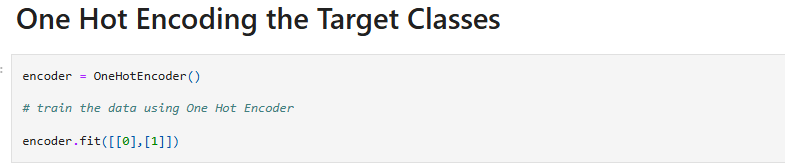
**IMPLEMENTATION:**

1. Import all the necessary libraries:



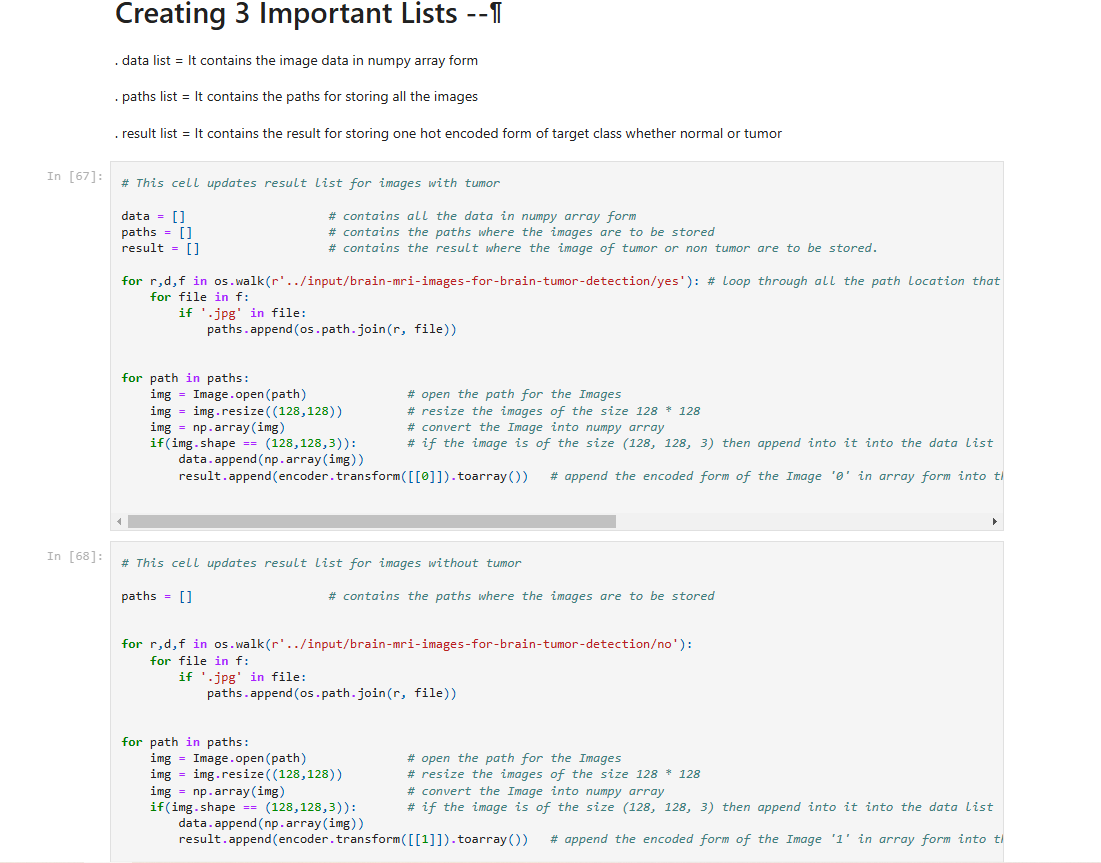
**OBSERVATION:**

1. **os** -> It is an operating system module that interacts with the operating system and helps in retrieving the data from the folder data location.
2. **Keras** -> It is a predefined library used to develop and build the neural networks system.
3. **Sequential** -> It is a library used in keras that is used to divide the input into layers where every layer has one input and one output. It is a sequentially stack of layers connected one after another.
4. **Conv2D ->** It is a library used in keras that is used to apply convolution on 2\*2 image to extract important features from the images useful for feature extraction and image classification etc**.**
5. **MaxPooling2D ->** It is a library used in keras that is used to select the maximum size from the given input size. It helps in dimensionality reduction.
6. **Flatten ->** It is used to convert 2D and 3D images into 1D image form.
7. **Dense ->** It is a library used in keras that is used to create the dense neural networks.
8. **Dropout ->** It is used to reduce the input size such that less data can be used for training the neural network model and it reduces the overfitting.
9. **BatchNormalization ->** It is a library used in keras. It is a sequentially stack of layers connected linearly one after another. Every layer has one input layer and one output. Every layer is linearly connected one after the other.
10. **Image** -> It is a layer that is used in image processing and is used to process the images
11. **train\_test\_split** -> It is a library used in scikit-learn that is used to divide the dataset into the training data and testing data.
12. **One Hot Encoder** -> It is a library used in keras that is used to convert the categorical data into numerical data.
13. Encoding the Target Classes:



**OBSERVATION:**

1. Here we are encoding the target classes ‘**Yes**’ and ‘**No**’ into the numerical form of ‘1’ and ‘0’.
2. Creating three important lists:

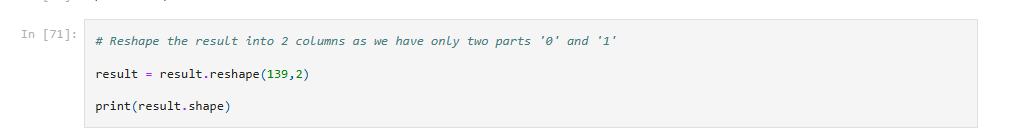


**OBSERVATION:**

1. **Data list ->** It contains the data in numpy array form.
2. **Path list ->** It contains the paths for storing the images.
3. **Result list ->** It contains the result for storing the encoded form of the target images whether it is normal or tumour**.**

* First we are looping through the location that has the images of brain tumour in it.
  1. Then we tract every image, resize the image in the form of 128 \* 128 , then we convert it into numpy array form.
  2. Then we append the numpy form of image into the data.
  3. The result list has the encoded form of the target (‘0’) that has brain tumour in it.
* Second, we are looping through the location that has the images of brain tumour in it.
  1. Then we tract every image, resize the image in the form of 128 \* 128 , then we convert it into numpy array form.
  2. Then we append the numpy form of image into the data.
  3. The result list has the encoded form of the target (‘**1**’) that does not have any brain tumour in it.
* The result list has now encoded form of ‘1’ and ‘0’ that represents the target.

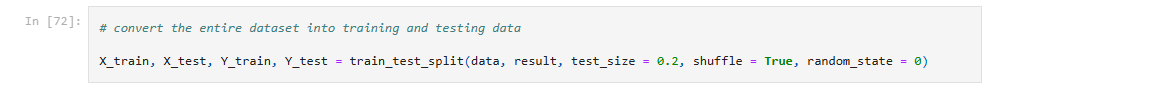
1. Resize the result that represents the target.



**OBSERVATION:**

* 1. The result of the target is resized into two columns that represents ‘0’ and ‘1’.

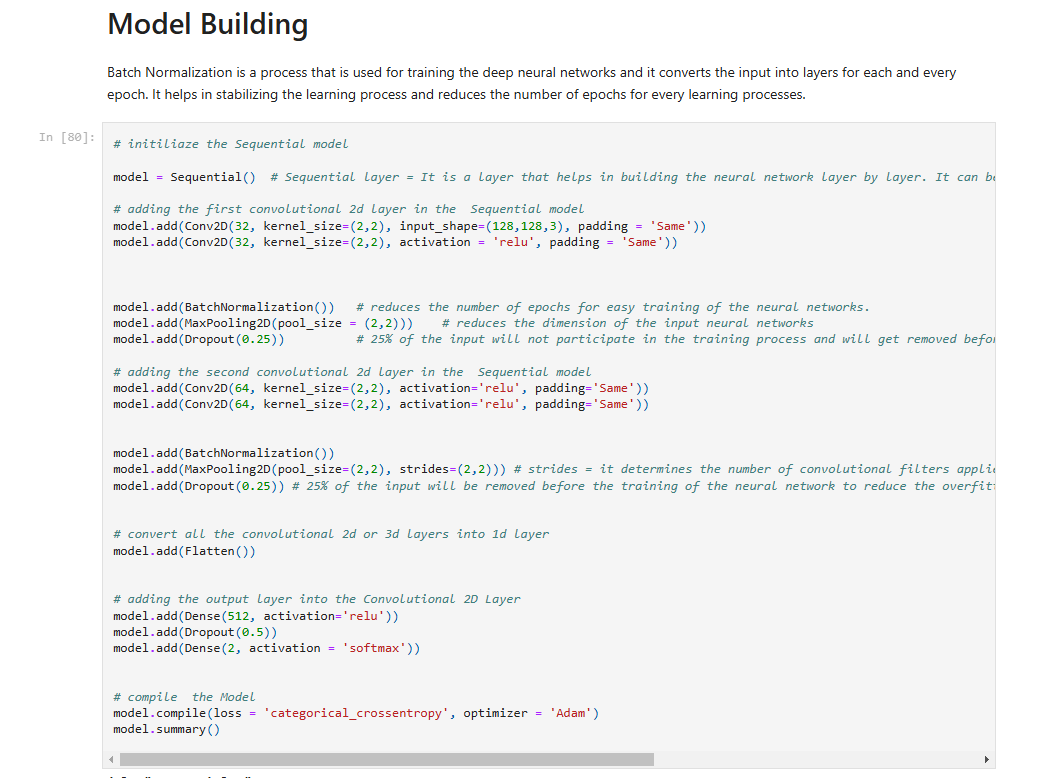
1. Split the dataset into training and testing data.



**OBSERVATION:**

1. The dataset is divided into training and testing data i.e. (training data has 80 % and testing data has 20%)

4. **Model Building:**



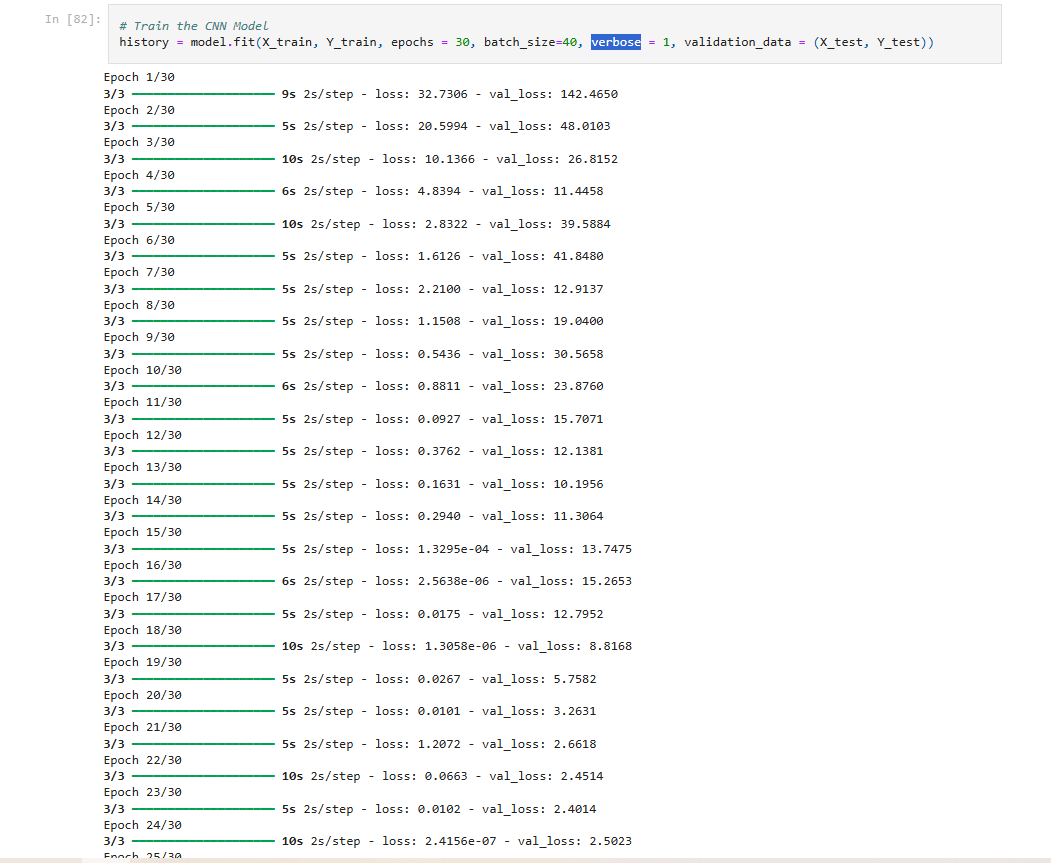
* A deep learning Convolutional Neural Network (CNN) is developed that is used to train the data so that all the images that has brain tumour can be identified.

**Steps for building a Convolutional Neural Network:**

* + A **Sequential Model** is initialized.
    - **Sequential Model** = It is a model that is a linear stack of layers. Here all the layers are linearly connected one after the other. Every layer has one input and one output.

* + **FIRST CONVOLUTIONAL NEURAL NETWORK :** 
    - Then first convolutional layer is added into the Sequential Model with the input\_shape = (128,128,3) and activation = ‘**relu’**
    - **Relu =** It is the activation function used in the Convolutional Neural Network that is used in the image processing to reduce the non- linearity of the images.
    - Before moving to the next layer, some of the inputs are removed, so that less number of inputs can be trained easily and the model training can be done in less time and to prevent over-fitting.
  + **SECOND CONVOLUTIONAL NEURAL NETWORK :** 
    - Then second convolutional layer is added into the Sequential Model with the activation = ‘**relu’**
    - Before moving to the next layer, some of the inputs are removed, so that less number of inputs can be trained easily and the model training can be done in less time and to prevent over-fitting. This is done using Batch Normalization, Max pooling 2D and Dropout activation function.
* Then all the input layers are flattened into one single 1-D layer that is then fed into the output layer.
* The output layer is added into the Convolutional 2D Layer using ‘**softmax**’ activation function and then the model is compiles using ‘**categorical\_crossentropy**’ and ‘**adam**’ optimizer.
  + **Adam Optimizer** = It helps in adjusting the learning rate for better training of the model.
  + **Categorical Cross entropy** = Categorical Cross-Entropy measures the difference between the true label distribution (ground truth) and the predicted label distribution

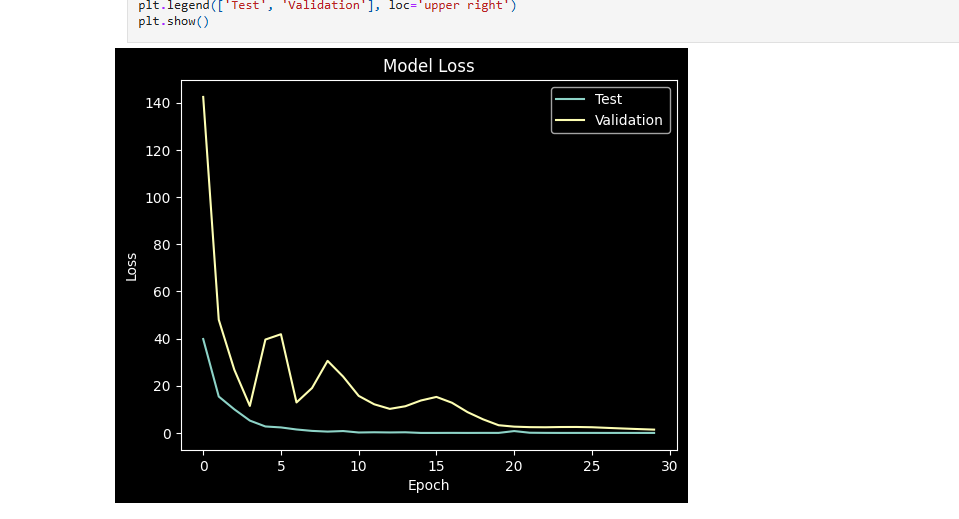
1. **Model Training :**



**OBSERVATION:**

* Here the entire CNN Model is trained andloss and **validation\_loss** is calculated for every epoch.
  + **PARAMETERS:**
    - **Epochs =** no of times used for training the dataset
    - **Batch\_size =** batch of data used for training the data.
    - **Verbose =** amount of input data used for training the data.

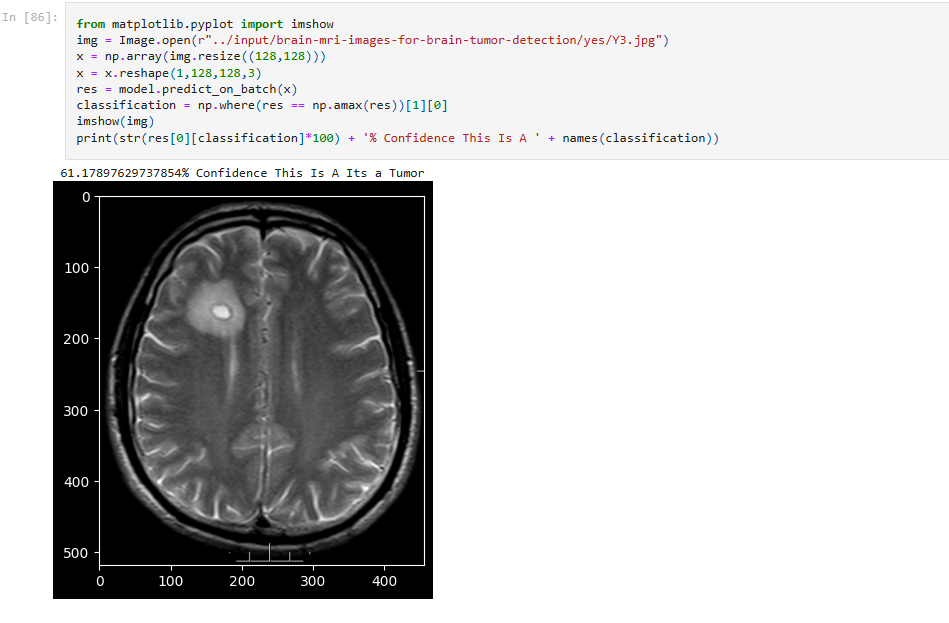
1. **RESULT PLOTTING:**



**OBSERVATION:**

* As the number of epochs increases, the loss value and validation loss keeps on decreasing.

**CHECKING THE MODEL:**



**OBSERVATION:**

* Here we have used the test data as an input brain image. Then we have predicted that this brain image has tumour in it.