**About Dataset**

**Abstract**

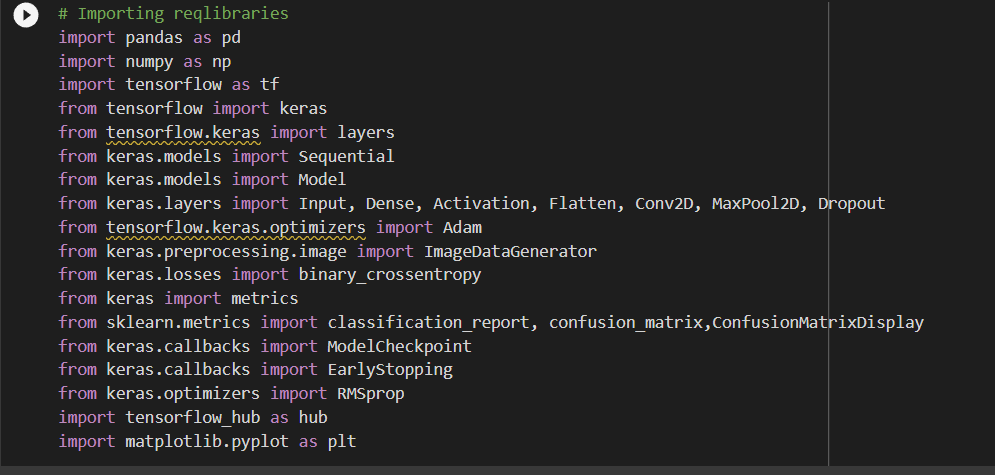
A Brain tumor is considered as one of the aggressive diseases, among children and adults. Brain tumors account for 85 to 90 percent of all primary Central Nervous System(CNS) tumors. Every year, around 11,700 people are diagnosed with a brain tumor. The 5-year survival rate for people with a cancerous brain or CNS tumor is approximately 34 percent for men and36 percent for women. Brain Tumors are classified as: Benign Tumor, Malignant Tumor, Pituitary Tumor, etc. Proper treatment, planning, and accurate diagnostics should be implemented to improve the life expectancy of the patients. The best technique to detect brain tumors is Magnetic Resonance Imaging (MRI). A huge amount of image data is generated through the scans. These images are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties.  
Application of automated classification techniques using Machine Learning(ML) and Artificial Intelligence(AI)has consistently shown higher accuracy than manual classification. Hence, proposing a system performing detection and classification by using Deep Learning Algorithms using ConvolutionNeural Network (CNN), Artificial Neural Network (ANN), and TransferLearning (TL) would be helpful to doctors all around the world.

**Context**

Brain Tumors are complex. There are a lot of abnormalities in the sizes and location of the brain tumor(s). This makes it really difficult for complete understanding of the nature of the tumor. Also, a professional Neurosurgeon is required for MRI analysis. Often times 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.

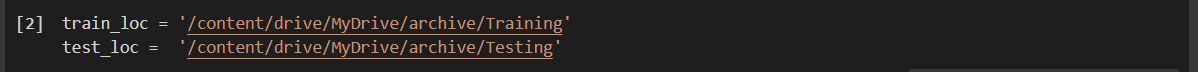
**Definition**

To Detect and Classify Brain Tumor using, CNN and TL; as an asset of Deep Learning and to examine the tumor position(segmentation).



Step-1 Importing the dependencies , as we are using Transfer learning and neural networks we need to import tensorflow

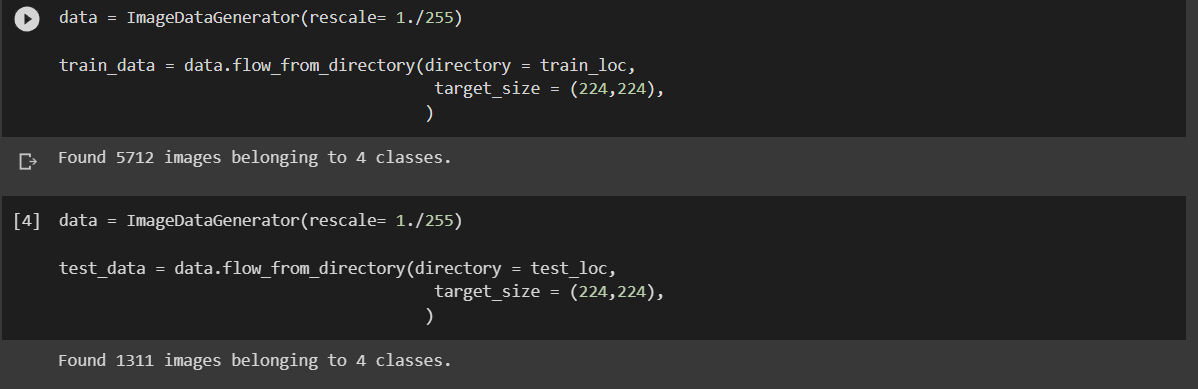
Step-2



Defining the training and testing directories

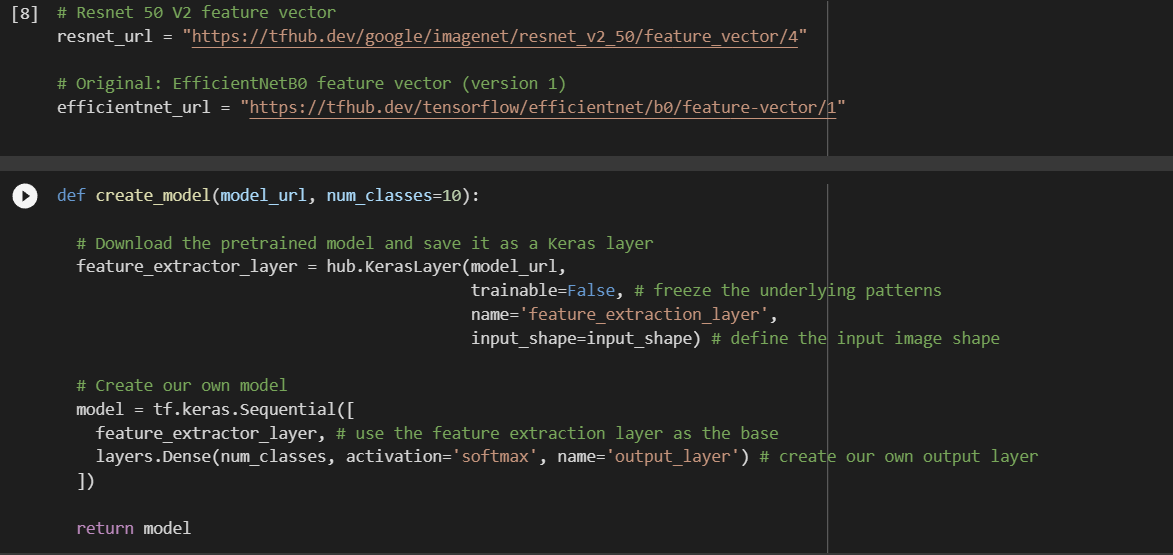
(We have uploaded the dataset available from Kaggle on google drive and further used google colab for the same)

Step-3



Generating the image dataset using ImageDataGenerator from tensorflow which creates the train data and test data with their labels for the images according to the file format specified for ImageDataGenerator

Step-4



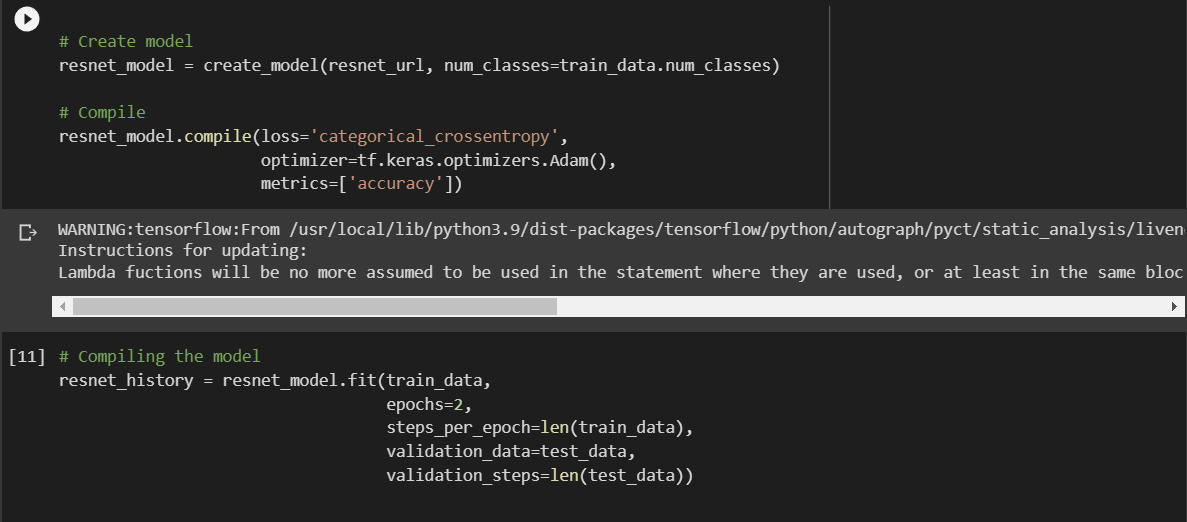
We use transfer learning so we need to import tensorflow hub to get access of transfer learning models available for use like efficientNet Resnet,Mobilenet,Vgg16

Here we are using 2 models Resnet\_v2\_50

And efficientnetB0

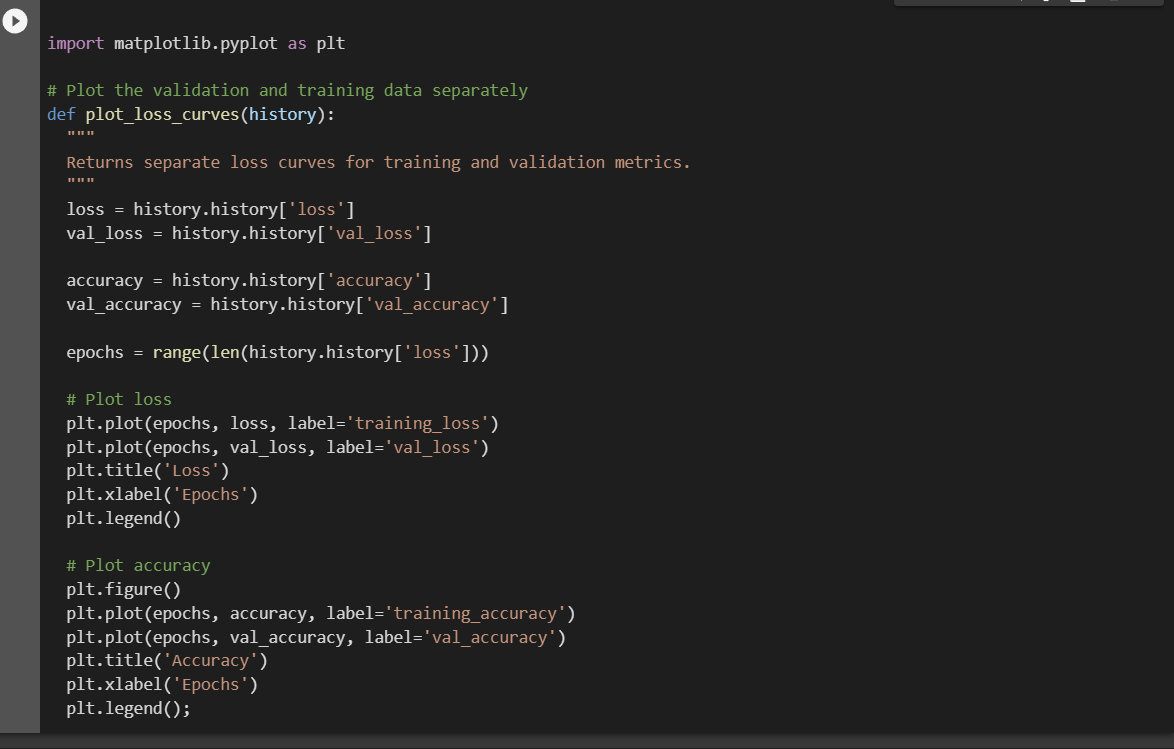
And we define the model for the same as a function

Step-5



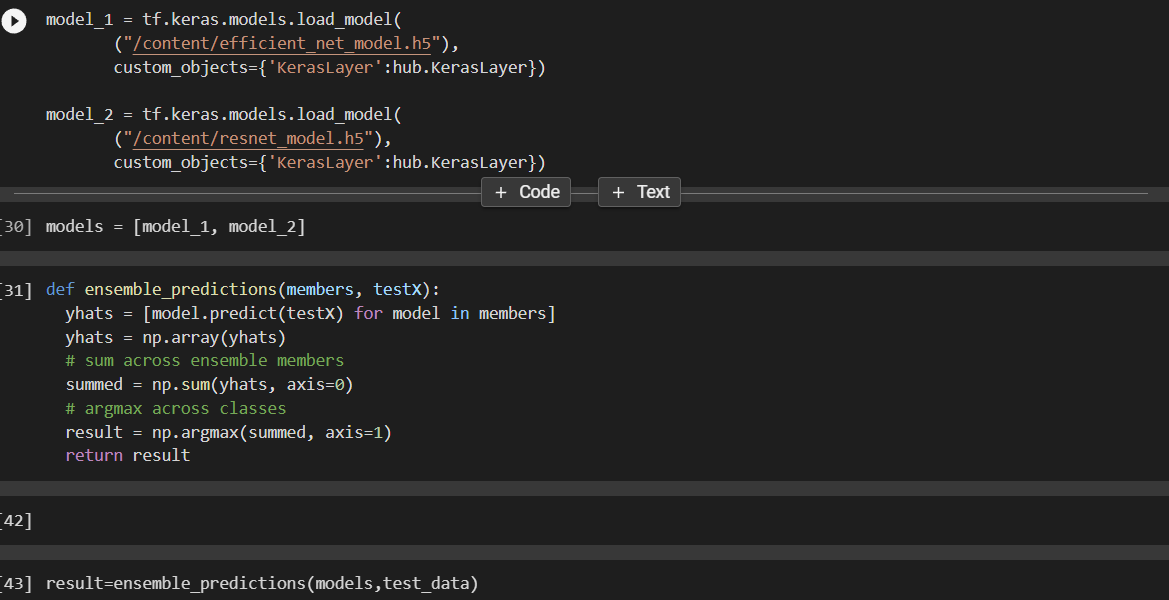
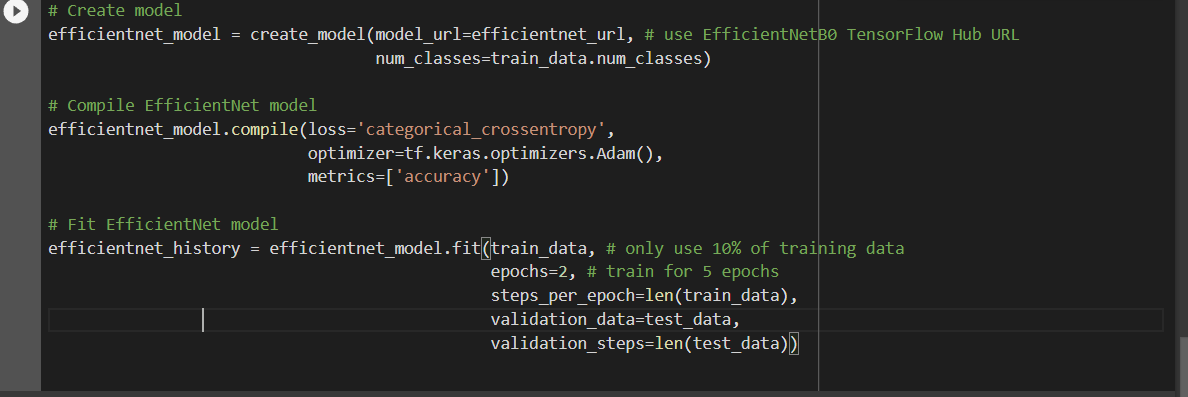
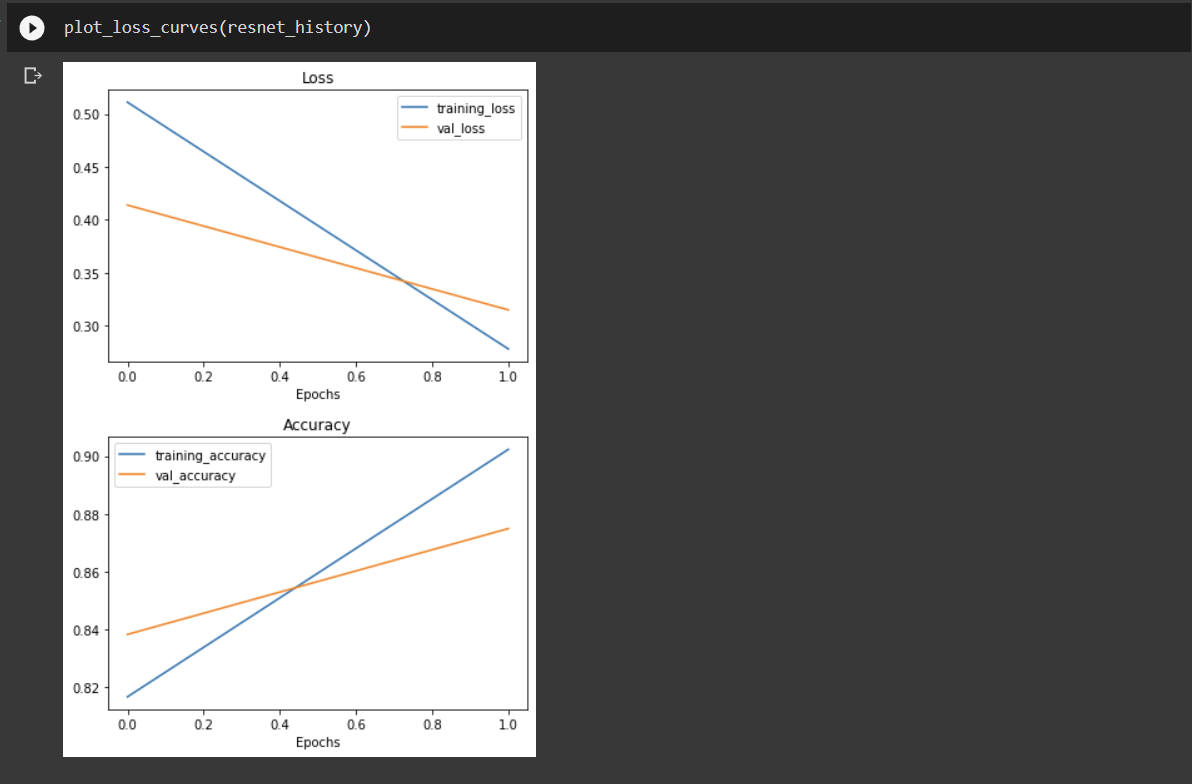
Creating and compiling the model

Step-6

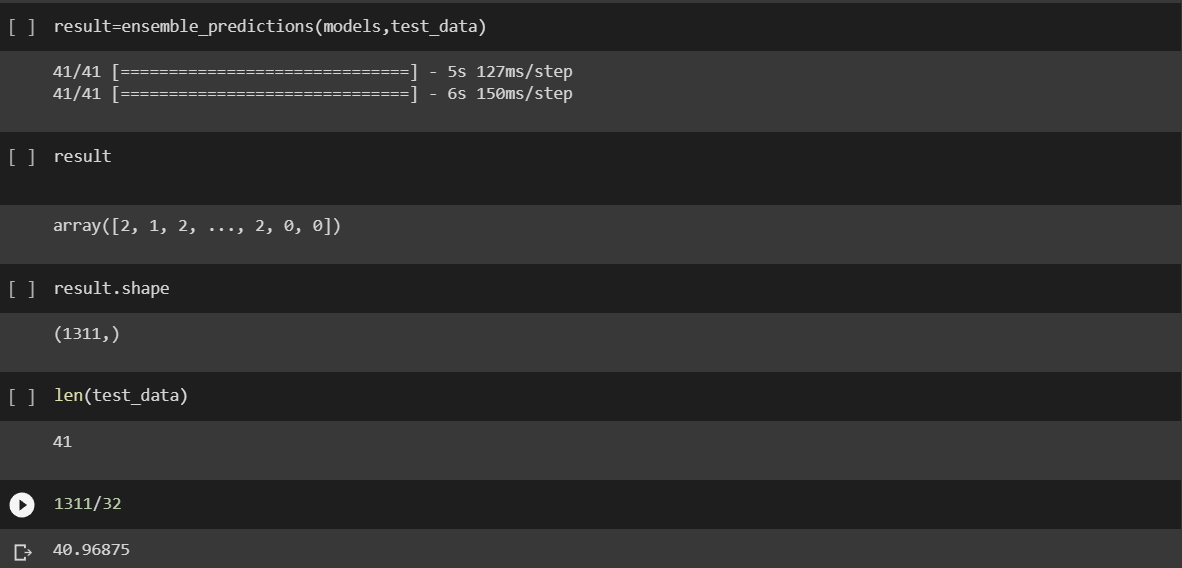


Function to plot the loss curves to understand the trend of the data

As the model consists of a history object it helps us to analyse the model using the accuracy and loss after each epoch



Here we have defined a ensemble functions which compares the predicted features from both the models and results the output classes to compare



As we have 1311 images in the test set which are divided in batches of 32 so we total of 41 batches of 32 images each and the result array consists of the resultant labels