## Brain Tumor Detection using Deep Learning

## Introduction

A brain tumour is a disease that develops as a result of the proliferation of abnormal cells in the brain. There are two types of brain tumours: non-cancerous (benign) brain tumours and cancerous (malignant) brain tumours. Non-cancerous (benign) brain tumours are the most common kind of brain tumour. Brain tumours of all kinds may cause different signs based on the region of the brain where they are found. Patients suffering from brain tumours may have migraines, seizures, difficulties with eyesight, nausea, and altered mental states. The typical example is that the headache is worse in the day times and gets better after vomiting. Additional symptoms include a decreased ability to move, talk, and experience pain. The latter stages of the illness will produce unconsciousness.

Tumours arise from a malignant change in the cells that comprise the brain. Generally, cells in the human body replace the old cells with the new cells being generated in a regulated way. This does not apply to brain tumours, however; the tumour cells in these cases may continue to grow at an exponential rate. The American Brain Tumour Society estimates that there are 70,000 primary brain tumours diagnosed in the United States. In India, brain tumours are the 10th most frequent kind of tumour. Tumour is detected by MRI scanning when it is present. Medication based on the findings of the MRI scan must be given by the doctor. Depending on the complexity of the equation, this process may take some time. This project uses an automated method to classify whether or not the patient has a brain tumour. By using this method, the physician may use early choices to begin treatments sooner, ultimately leading to better results.

In this project, I will be using the Computer Vision technique to classify malignant brain tumours. I will be using different types of CNN structures and compare them against the transfer learning model to evaluate the results between these two types of CNN models.

## 1.1 Brain Anatomy

The brain tumour is one of the most frequent and, as a result, one of the most deadly brain disorders that has touched and ruined many lives throughout the world. Cancer is a brain illness in which tumor cells spread throughout the brain tissues. Conferring with a new person According to a cancer study, over a lakh people diagnosed with brain tumours each year. all throughout the world Despite consistent efforts to alleviate brain problems, Figures for tumour patients reveal unfavorable outcomes. Scholars are working to counter this. working upon computer vision to have a deeper understanding of cancers in their early stages how to use sophisticated treatment alternatives to overcome Computed tomography (CT) and magnetic resonance imaging (MR) images of the brain The two most common tests to look for the presence of a tumour and recognize it are the brain and the MRI. position for making treatment decisions in the future These two scan are still widely utilized today.

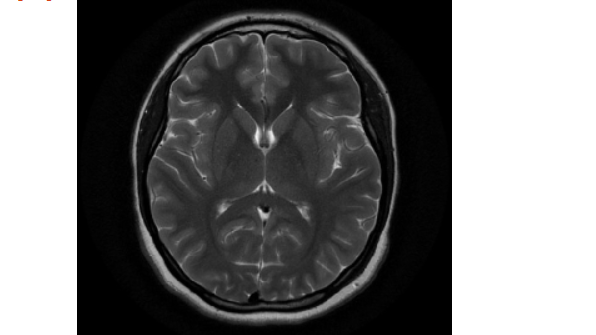
for their portability and capacity to produce high-definition images of diseased conditions There are additional tissues. There are currently various additional approaches available for malignancies. Surgical procedures, as well as medicines like as radiation therapy and chemotherapy, are all options. The choice was made. Treatment depends on a variety of circumstances, including the tumor's size, type, and grade. visible in a MR image It's also in control of whether or not cancer. Other parts of the body were affected. A precise diagnosis of the type of brain dysfunction is critical. With the goal of reducing diagnostic errors, therapeutic operations are carried out. The accuracy is excellent. Computer-aided diagnostic (CAD) technologies are frequently used in a haphazard manner. The most important strategy The goal of computer visual is to create a dependable output, that is an estimate of something else. to aid medical physicians in image comprehension and reduce picture reading time These Medical diagnosis is becoming more consistent and accurate as technology advances; however, It's a difficult job to segment an MR picture of a tumour and its surrounding area. The tumours appearing in specific locations within a brain picture without being distinguished An additional difficulty that makes a computerized brain detection difficult is picture intensities. It's a difficult job to remove a tumour and segment it.

## 1.2 Data Set

The data that I will be using in this project is secondary data that is readily available in Kaggle. The dataset consists of two files one file includes the images that belong to malignant brain tumour MRI images and the other file consists of the images that belong to benign brain tumour MRI images. The malignant is harmful to humans, on the other side, the benign brain tumour is non-cancerous cells in the brain. There are a total of 253 scanned images in the dataset, which is quite a smaller number of images for training a deep learning model. Hence, we can use data augmentation to generate new images from the existing data and increase the number of images in the dataset. I have downloaded the dataset from Kaggle <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection> . This dataset contains a balanced dataset of images.



1. Yes (Having Brain tumor)



2 . No (No Brain tumor)

## 1.3 Project Aim

In this project our aim is to detect the brain tumor using the convolutional neural networks and transfer learning by training these networks with different architectures and evaluate the networks based on the accuracy they have achieved. Our primary aim remains to build a classification model that could classify between the Brain tumor.

## 1.4. Objectives

1. Study about the deep learning techniques that are used in image processing and gain deep understanding of these techniques.
2. Also learn about the transfer learning and understand the methods to implement the transfer learning.
3. Get the domain knowledge in the subject area to have a better understanding about the project that we are working on.
4. Classify the benign and malignant brain tumours using Deep learning techniques.

## 1.5 Research Questions

Research questions are the important part of a study that allows the researchers to learn in-depth regarding a subject. The researchers aim to answer the research questions related to their work. The following research questions were identified in the project that will be analyzed during this project.

1. Is transfer learning is beneficial over simple CNN models?
2. What is the future scope for CNN in the medical imaging in predicting the brain tumours?

The research questions that we are focusing on in this project will help us learn the advantages of using the pre-trained models over the regular CNN models and lets us learn the importance of the CNN’s in the medical imaging.

## 1.6. Tools

There are several tools employed as phase of this project to achieve its main objective. a few essential tools utilized during this Endeavour include:

Tensorflow : Tensorflow is a machine learning and ai - based software library that is free and open-source. It can be used for a variety of applications, but it focuses on deep neural network training and inference. Tensorflow is a dataflow and differentiable programming-based symbolic math toolkit.

NumPy: NumPy is a Library in python designed to help with array management. It has features for linear algebra, Fourier transforms, and matrices, as well.

Pandas: Pandas is a Python-based library that works with data analysis and manipulation. The focus is on the operations and data structures required to manipulate tables and time - series data.

Scikit-learn: Scikit-learn is a library for Python that helps with machine learning. It is open source and available to anyone. It includes several algorithms, such as support vector machines, among others.

Matplotlib: Matplotlib is a tremendous 2D plotting library in Python, perfect for visualizing array data. Matplotlib is a library built on the concept of NumPy arrays, and it is made to work with the other components of the Scipy stack. Matplotlib has numerous plots, including lines, bars, scatter plots, histograms, and more.

Seaborn: Seaborn is an example of a Python library that works with the matplotlib data visualization framework and integrates with pandas data structures. Seaborn is Seaborn's central visualization system, which is crucial in helping the exploration of data. See how the distribution is univariate and bivariate.

# 2. Ethical, Legal, and Social Issues

|  |  |
| --- | --- |
| **Issues** | **Reasons** |
| Ethical issue | **Risk Zone 4 Machine Ethics & Algorithmic Biases**  How reliable an AI algorithm can be in the matter of identifying dangerous diseases and saving human lives. There are many failures of AI systems in recent times. Any misinterpretation of the results will affect the life of an individual.  **Risk Zone 6 Data Control & Monetization**The individuals whose data is being collected should know how the data is being used and to whom the data is being shared. There are multiple such instances where organizations share the data of the individuals to the other companies to make profits. |
| Legal issue | The results of the outcomes from the algorithm should be understandable by everyone. The misinterpretation of the results from the doctors who lack technical aspects could result in a serious threat to the patients. Hence it is important that the results of the outcomes should be readable by everyone. Any miscommunications in the readings will lead to legal charges against the person responsible for building the algorithm and the doctors. |
| Social issue | The social issue that is possible in this project is the effect of mental health on patients due to unexpected results that might show up because of misclassifications that might occur in the model in the case of outliers. In such situations, the patients who are being examined for the brain tumour can get into mental depression and can harm themselves in that mental state. |

## 3. Methodology

## 3.1. Installing set-up

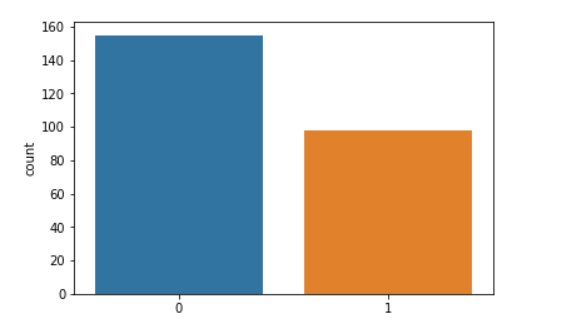
I have used Python 3.7, which I downloaded from the official Python website, for my project. I installed Python on my machine by setting it up on my hard drive and adding Python to my path. Additionally, I've installed Anaconda on my computer to get everything ready for running Python via Jupyter. After that, I was able to execute this project by using command prompt to install a few libraries.

Example : Pandas, Numpy, Scikit-learn, Tensorflow etc.

## 3.2 Importing Dataset

We are working on Jupyter environment since it has a very interactive interface, and it is suitable for production. This IDE is the most used by Data Scientists across the globe. The initial step was to import all the libraries in the notebook. Firstly, we have to import all the data that was collected from the Kaggle in order to proceed with our research. All the data will be imported to the Jupyter IDE.

I have used the OS library to read the directory of the data file location and I have imported the dataset and assigned them to train and test variables according to the way they were organized. Since the whole data was divided into two classes, I have plotted a bar plot to see how the data is distributed among the classes in the train data.



Here 0 denotes no category and 1 denotes yes class.

## 3.3. Image Pre-processing

After importing the data into the IDE, I have separated the labels and features and stored them in different variables for all the sets, i.e., train and test. We further perform the image pre-processing on the data to normalize and re-size the images. This step is intended to enhance the picture quality so where we can conduct a more thorough analysis of the image. The use of pre-processing allows us to eliminate unwanted distortions and improve certain characteristics that are essential for the specific application for which we are developing. Those characteristics may vary depending on the application.

There are two pre-processing stems that were performed as discussed. They are as follows:

### 

## 3.3.1. Normalization:

Data normalization is a crucial step in the image processing process since it guarantees that every input parameter (in this instance, pixel) has a comparable data distribution. As a result, the network's convergence time is reduced while it is being trained. Data normalization is accomplished by removing the average out of each pixel and dividing the resulting value by a standard deviation of the data set. In this case, the distribution of the data may resemble a Gaussian distribution with the origin at zero. We need positive pixel values for picture inputs, therefore we could choose to scale the normalized data in the range [0, 1] or [0, 255], depending on the application.

For the purpose of normalizing our data, I have transformed the features into a NumPy array and then divided the array by 255. In the train, validation, and test sets, this process was repeated for each feature set.

## 3.3.2. Resizing images

In computer vision, resizing pictures is a crucial step in the pre-processing process. Machine learning techniques learn to recognize patterns in smaller pictures, which is important since they train quicker on smaller images. The learning process for a twice-larger input images needs the network to learn over 4 times as much of pixels, which increases the amount of time it takes. A further point to consider is that many neural network model designs need input images to be of the same size, while our raw gathered images may be of different sizes.

It was necessary to scale the whole data set to fall inside the range of [-1, 1] across all of the feature sets. This ensures that the image's size falls within the boundaries of the specified range.

## 3.4. Data Augmentation

The quantity and variety of data provided during training have a significant impact on the accuracy of predictions made by Supervised Deep Learning models. When it comes to deep learning models and the quantity of training data needed, the relationship is similar to the relationship among deep learning models and the massive quantities of data required for the deep learning model to be successful. Computer vision tasks including such images categorization, object recognition, and segmentation have shown to be very effective among deep learning applications that are already in use today. In these types of applications, data augmentation may be utilized to successfully train deep learning models. Simple changes that may be done to an image include transformations like Flipping, Rotating, Translations, Cropping, and Scaling, as well as pixel transformations such as colour casting, varying contrast, and noise injection are a few examples.

The data augmentation helps the models to be robust since the model’s train on images that are transformed. The operations that were performed on the training data are as follows:

1. **Rotation range:** This operation ensures that the images are rotated at certain angles. The user specifies the angle for rotation.
2. **Zoom range:** This is an operation that transforms the images by performing the zoom in and zoom out.
3. **Width shift range:** It really causes the picture to be shifted horizontally towards left or right. If the value is variable and the range is more than one, the percentage of the entire width will be used as the range. Assume that the picture width is 100px. If width shift range = 1.0, the range will be from -100 percent to +100 percent, which is equivalent to -100px to +100px. It will move the picture in a random manner between these two ranges. A randomly chosen positive number will move the picture to the right side of the screen, while a randomly selected negative value would move the image towards the left of the screen. This may also be accomplished by choosing pixels. It will have the same impact if we specify width shift range = 100, which is the default value. More significantly, integer values more than one count pixel are used as range, whereas float values less than one count percentage of entire width are used as range.
4. **Height shift range:** It functions in the same way as width shift range but shifts vertically up or down.
5. **Horizontal flip:** This operation performs the random horizontal flip on the images to generate new images.

These whole operations were performed using the Keras image data generator function that allows to transform the images.

## 3.5. Deep Learning - CNN

Artificial Intelligence has seen tremendous progress in recent years in terms of reducing the gap in between abilities of mankind and those of computers. Researchers are working on a variety of areas of the field in order to bring about amazing results. In this regard, the field of Computer Vision is only one among several. Among the objectives of this field is to enable machines to view and comprehend their environment in the same way that humans do, and to use this expertise for a variety of tasks such as image and video recognition, image processing & categorization , recommendation systems, etc. The advances in Computer Vision in regard with Neural Networks have been built and refined over time, mainly via the use of a single algorithm — the Convolutional Neural Network as the foundation.

When given an images as input, a Convolutional Neural Network (ConvNet) may assign significance to various features or objects in the image, and then distinguish between them. When contrasted to all other classification methods, the amount of pre-processing needed by a ConvNet is much less. While basic techniques need filters to be hand-engineered, ConvNets have the capability of learning these characteristics with sufficient training. In many ways, the design of a ConvNet is similar to the connection patterns of Neurons in the Structure Of the brain, and it was influenced by the structure of the Visual Cortex. Every single neuron in the human visual system responds to signals primarily in a certain area of the field of vision termed as Receptive Field. A group of similar fields may be used to fill the whole visual region by overlapping them. For this project we are experimenting with different combinations of layers and the convnets. The models that were used in our project are listed as follows:

1. 3 layer CNN model with 3x3 convnets
2. 3 layer CNN model with 5x5 convnets

**Activation Function:**

Here in these networks, the activation function used was ReLu activation function. the rectified linear activation function also known as ReLu is a non-linear function that could produce the input directly if the input is positive and will otherwise produce zero if the input is negative. Due to the fact that a model that employs it is simpler to train and produces higher performance in many cases.

Chart, line chart, scatter chart

Description automatically generated

Figure 6 - ReLu activation function (Toprak, 2020)

**Loss Function:**

Binary cross entropy loss function was used in the CNN network since this is a two class classification problem. The binary cross entropy is calculated using the following formula:

Logo, company name

Description automatically generated

Figure 7 - Binary cross entropy loss function

**Optimizer:**

The optimizer used in the CNN network is ADAM optimizer, also known as Adaptive Moment Estimation, is an optimization method for gradient descent that is based on an algorithm. When dealing with a massive problem containing a large number of data points or parameters, this technique is very efficient. As a result, it uses less memory and is more efficient.

**Evaluation Metrics:**

The evaluation metrics used to evaluate the CNN models is Accuracy. On the basis of training data, accuracy is the metric used to evaluate which model is the most effective at detecting patterns and connections among variables in a dataset.

## 

## 3.5.1. Architecture of CNN

Every layer of a basic ConvNet converts one region of activation functions to another via the use of a differentiable function, and each layer of a simple ConvNet is composed of layers. Convolutional Network designs are built using three kinds of layers: the Convolutional Layer, the Pooling Layer, and the Fully-Connected Layer. These layers will be stacked together to create a complete ConvNet architecture.

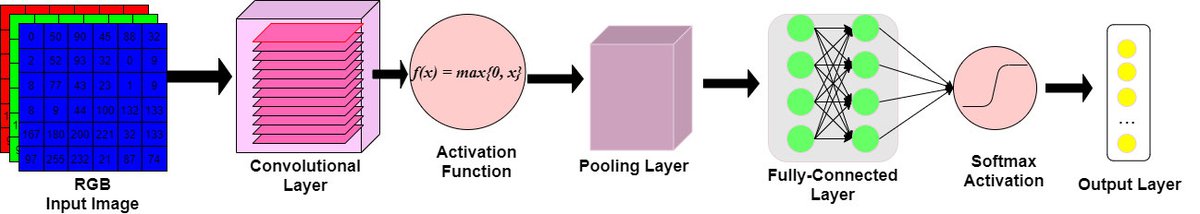


Figure 5 - Architecture of CNN

As mentioned above there are various layers in the architecture of a CNN. The layers can be described as follows:

1. Input Layer: The raw image pixels of the picture will be passed through the input layer.
2. Convolutional Layer: Convolution layer is the most essential building blocks of convolutional neural networks, and they are utilized in many different applications. Convolutional layers operate on the input by applying a convolution operation and transferring the output towards the next layer. A convolution is a mathematical operation that transforms all of the pixels through its receiving area it in to a numeric measure. For instance, if we perform a convolution to a picture, we will be reducing the size of the image while also condensing all of the data in the field into single pixel in the process. A vector is produced as the final outcome of the convolutional layer. Different types of convolutions may be used depending on the kind of issue we are attempting to solve and the type of the attributes we are attempting to learn from the data.
3. Pooling Layer: Pooling layers are a technique for down sample feature maps that summarizes the existence of features in regions of the feature map, allowing for more efficient sampling. Average pooling & max pooling are both often used pooling techniques that summaries the mean existence of a feature as well as the most active presence of a feature, respectively, in a dataset.
4. Fully connected Layer: Feed forward networks are what the Fully Connected Layer is all about. The Fully Connected Levels are the last few layers of the network hierarchy. A flattened version of the outputs from the last Pooling Layer is sent into the fully connected layer, which is then fed back into the fully connected layer.

## 

## 3.5.1.1. layer CNN model with 3x3 convnets

The structure of this layer includes three convolutional layers that has the 3x3 convnets. Assume that the input volume has the proportions [16x16x10].  With an example receptive field size of 3x3, each neuron inside the Conv Layer would then have a maximum of 3\*3\*10 = 90 connection towards the input volume.

The architecture of this CNN constitutes as follows:

Graphical user interface, text, application

Description automatically generated

Figure 8 - 3 layer CNN model with 3x3 convnets

The total parameters from the above model are out of which all the parameters are trainable.

At the end of twelve epochs -

Test accuracy: 82.35%

Test Loss: 1.72

### 

## 3.5.1.2. layer CNN model with 5x5 convnets

The structure of this layer includes three convolutional layers that has the 5x5 convnets. Assume that the input volume has the proportions [16x16x10]. With an example receptive field size of 5x5, each neuron inside the Conv Layer would then have a maximum of 5\*5\*10 = 250 connection towards the input volume.

Graphical user interface, text, application

Description automatically generated

Figure 10 - 3 layer CNN model with 5x5 convnets

The total parameters from the above model are 2,269,809 out of which all the parameters are trainable.

At the end of twelve epochs -

Test accuracy: 76.47%

Test Loss: 2.61

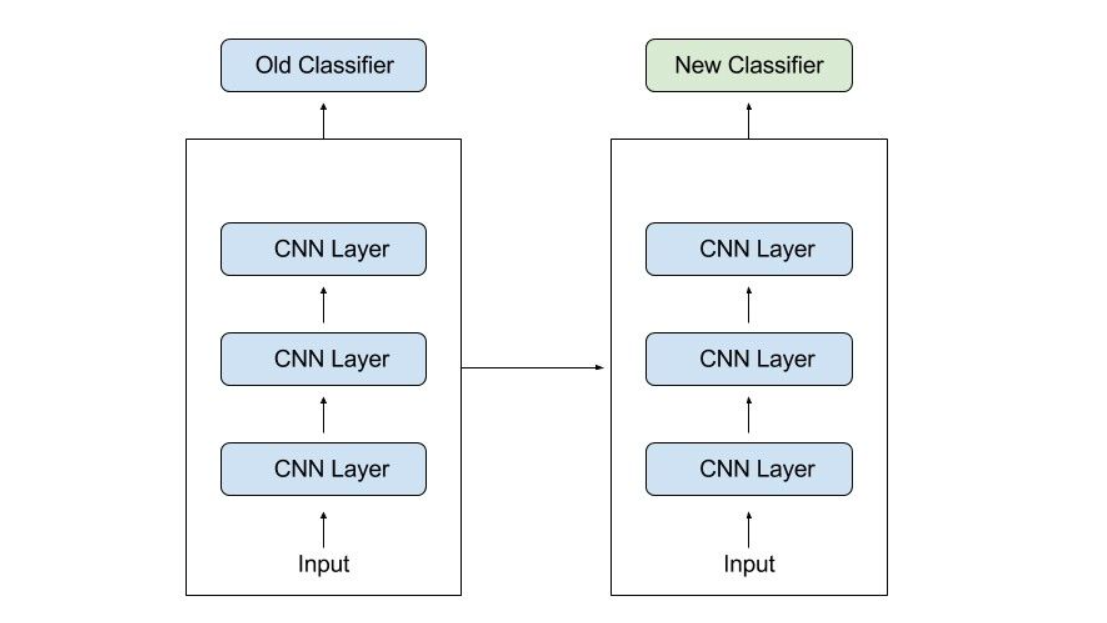
## 3.6 Transfer Learning

Transforming a model is trained on a huge database to another smaller one is at the heart of transfer learning. To use a CNN for object recognition, we freeze the network's early convolutional layers and train only the final few levels that make predictions. A pre-trained model is reused on a new problem through transfer learning (TL). Due to its ability to train convolutional neural networks with relatively little input, it's now quite popular among deep learning researchers. Since most real-world situations don't have millions of labelled data sets to train such complicated models, this is extremely helpful inside the data science field. Our goal is to provide you a better understanding of transfer learning. Transfer learning will be discussed, as well as some resources on pre-trained models will be provided.

## 3.6.1 Working of Transfer Learning

Brain networks are often used in computer vision to recognize edges, forms and task-specific properties in the early levels of the computer's vision system. The early to mid layers are utilized in transfer learning, and only the latter layers are retrained. It makes use of the labelled data from the task this was originally trained on.

Consider the case of a model that was trained to recognize a backpack in a picture and be used to detect sunglasses. For this reason, only the last layers will be retrained to learn how sunglasses differ from other objects. The model has already learned to distinguish objects in the prior levels.



As much information as feasible from the previous work is transferred to the current task when using transfer learning. Information can take a variety of shapes based on the situation and the available data. For example, the way models are put together may help us detect new items more quickly.

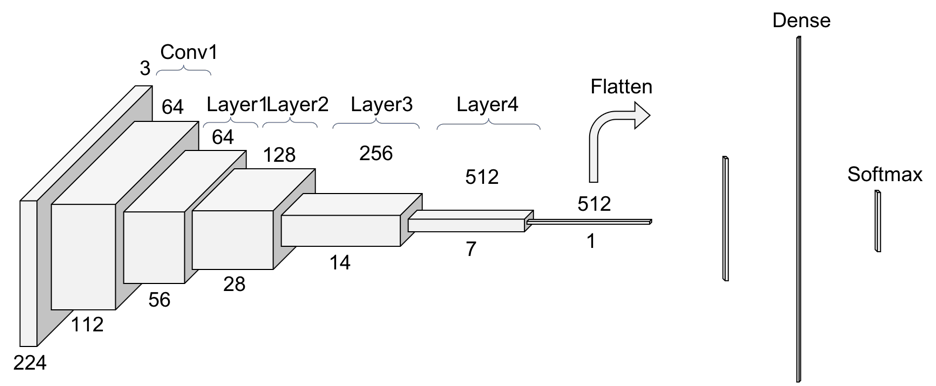
In this project I have implemented two models of transfer learning.

## 1. Resnet v2

It differs from regular sequential networks in that ResNet uses microarchitecture modules known as a network inside the architecture rather than sequential network such as OverFeat and VGG. New networks are often built from the ground up using microarchitecture, which is a collection of small building blocks. Based on Kaiming He's seminal work that was first published in 2015, this research shows how deep network can be train using regular SGD by modifying the residue module to employ identity mapping as proven in the previous work.

In spite of ResNet's greater depth, the real model weights are smaller because average global pooling instead of the convolution layer is employed, resulting in a smaller model size for ResNet50, which is more commonly used for 100MB's models.

## Architecture :



These are the some important points of Resnet v2.

* It is based on the 'Residual Learning' idea. Residual Learning that is based on stacking residual blocks to improve performance of neural networks.
* Resnet stands for Residual Network, which is a network that facilitates Residual Learning. The number 50 denotes the number of layers. Resnet50 refers to a 50-layer residual network.
* Due to their inability to extract significant information from images, simpler Neural networks do not work well on dataset.
* As a result, the number of hidden layers will automatically grow in order to improve performance indicators like as accuracy and AUROC. However, when we add additional layers to our neural networks, the accuracy begins to saturate and ultimately decline. Residual Training aims to address this issue.
* Instead of attempting to learn some attributes, residual learning focuses on learning some residual. Residual could be simply defined as the removal of a feature learned from a layer's input. Shortcut connection (directly connecting the input of the nth layer to the input of some (n+x)th layer) are used by ResNet to accomplish this. It has been demonstrated that training this type of network is simpler than training simple cnn model, and that the issue of accuracy degradation has been handled.
* Inputs can propagate quicker through back propagation across layers with residual blocks, and Batch Normalization layers are used to speed it up runtime and prevent overfitting.
* The Vanishing Gradients issue is also addressed by ResNets. It prevents the gradients from rapidly decreasing to zero. A "Quick" or a Skip Connections in ResNet allows a gradients to be directly back-propagated to previous layers.

The Implementation of Resnet V2 for this project is :



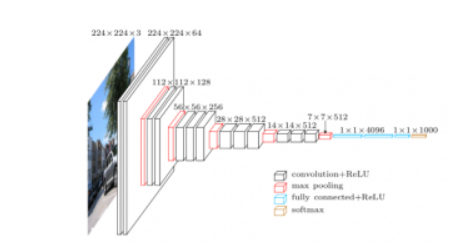
The accuracy on testing data is 50.98% and Testing loss is 7.47.

## 2. VGGNet

Karen Simonyan and Andrew Zisserman of the University of Oxford proposed the VGGNet architecture for a Convolutional Neural Network in 2014. The main focus of this research is on the accuracy of the cnn model as a function of its depth. Very Convolutional Neural Network for Wide Scale Image Processing is the title of the original study on VGGNet.

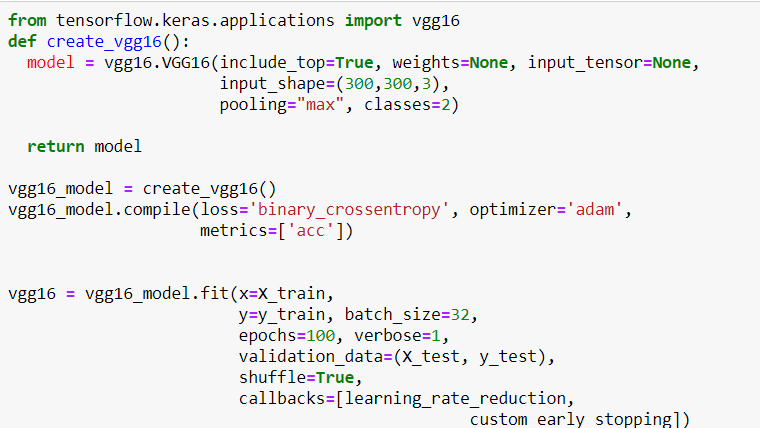
## VGGNet Architecture

When using VGG-based convNet, you'll need a 224\*224 Rgb as your input. The preprocessing layer subtracts the average image values determined for the complete ImageNet training set from the Image with pixels values ranging from 0–255.



These weight layers are applied to the preprocessed input photos. A series of convolutional layers is applied to the practice images. In the VGG16 architecture, there are a total of Thirteen convolution layer and three fully connected layers. Instead of using huge filters, VGG uses smaller ones (3\*3) that have more depth in their image processing capabilities. Using this method, the receptive field is identical to that of using only one 7 x 7 convolution layers.

The Implementation of VGG16 for this project is :



The Testing loss for VGG16 Network is 0.6890 and the accuracy on testing data is 54.54%.

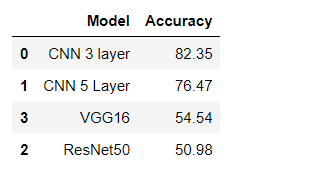
## 4. Evaluation Metrics

The evaluation metrics used to evaluate the CNN models and Transfer learning models is Accuracy. On the basis of training data, accuracy is the metric used to evaluate which model is the most effective at detecting patterns and connections among variables in a dataset. Accurately identifying the ratio of correctly predicted instances to the total amount of instances evaluated is a metric for measuring accuracy. While the accuracy might be adequate as a measurement for model performance, it may not be good enough because it fails to include incorrect predictions. If someone treats a fake post as a real one, it could cause a serious issue. False positives and false negative issues that accommodate for misclassification should be considered because of this.

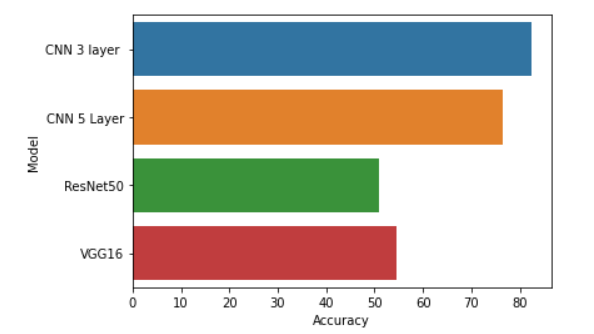
After training the models on the training set using the accuracy as metric, I have further evaluated the models on the test set that was provided along with the dataset.



The accuracy of the models is :



The bar graph of the accuracy is :



## 5. Conclusion

MRI Images are best suitable for brain tumor detection. In this study Digital Image processing Techniques are important for brain tumor prediction by MRI images. The preprocessing techniques include different methods like Augmentation, Data Normalization etc. The preprocessed images are used for post processing operation and for training the models.

In this project we are trying to detect the Brain tumor using the image evidences that were provided in the dataset. Using this dataset, we have performed several pre-processing operations and trained CNN and Transfer learning with different architectures. After applying different architectures we can see that CNN gives higher accuracy with testing data and the testing loss is very less. Upon comparing the performances of all the models that were trained as part of this project it has been observed that the performance of the seven layer CNN with 3x3 kernel size of convnets is more efficient on all the training and test sets. Various neural network strategies for skin brain tumor and classification were discussed in this systematic review research. These methods are all non-invasive. Preprocessing and picture segmentation are two phases in the detection of skin cancer. Following that, feature extraction and categorization are performed. The focus of this review was on CNNs and transfer learning for lesion picture classification. Each algorithm has its own set of benefits and drawbacks. The most important factor in achieving the best results is choosing the right classification technique. When it comes to classifying picture data, however, CNN outperforms other types of neural networks because it is more directly tied to computer vision than other types of neural networks.

Based on the analysis my findings from answering the research questions:

CNN gives higher accuracy as compare to transfer learning models. The fundamental advantage of CNN over its predecessors is that it discovers essential traits without the need for human intervention. Given a large number of photos of cats and dogs, it can learn the key characteristics of each class on its own. With the many CNN-based deep neural networks constructed, a considerable result was reached on the ImageNet Challenger, which is the most significant picture classification and segmentation challenge in the image analyzing area . In medical categorization, the CNN-based deep neural system is commonly utilized. It has been observed that CNN gives better accuracy than Transfer learning models.