# Final Project Deep Learning:

## Brain Tumors Detection Using Convolutional Neural Network.

**IBM Machine Learning Professional Certificate**

Course 05: Deep Learning & Reinforcement Learning| Brain Tumors Detection

By ARPAN SANKESH

IBM

# Contents

- Dataset Description
- Main objectives of the analysis.
- EDA, Data Cleaning, Feature Engineering
- Training deep learning models.
- ML analysis and findings.
- Models flaws and advanced steps.

# Data Description Section

# ntroduction

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 witha brain tumor. The 5-year survival rote for people witha cancerous brain or CNS tumor is approximately 34 percent for men and 36 percent for women. Brain Tumors ore classified as: , 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 ore examined by the

rodiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties.

Application of automated has consistently

shown higher accuracy than manual classification.

Hence, proposing a system performing detection and classification by using Deep Learning Algorithms using

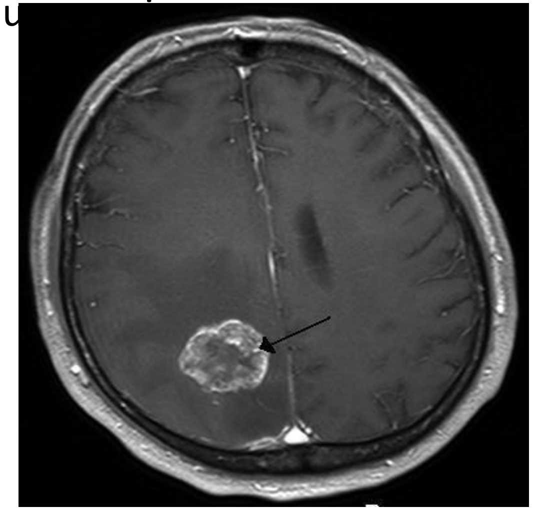
would be helpful to doctors around the world.

# Dataset Description

## What is Brain Tumor?

A brain tumor occurs when abnormal cells form within the brain. There are two main types of tumors: cancerous (malignant) tumors and benign tumors. Cancerous tumors can be divided into primary tumors, which start within the brain, and secondary tumors, which have spread from elsewhere, known as brain metastasis tumors. All types of brain tumors may produce symptoms that vary depending on the part of the brain involved. These symptoms may include headaches, seizures, problems with vision, vomiting and mental changes. The headache is classically worse in the morning and goes away with vomiting. Other symptoms may include difficulty walking, speaking or with sensations. As the disease

progresses, u



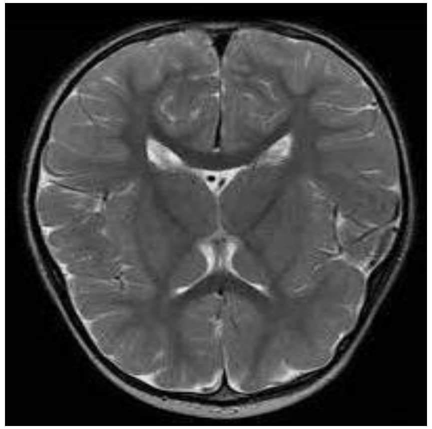
# Dataset Description

*Brain metastasis in the right cerebral hemisphere from lung cancer, shown on magnetic resonance imaging .*

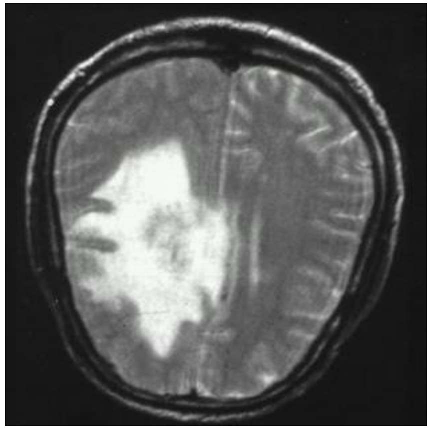
**Quick explanation of the dataset**

# Dataset Description

- YES – tumor, encoded as 1



0



1

an d d ataset for this problem is Brain MRI Images which will be used for Brain Tumor

# Dataset Description

consists of MRI scans of two classes:

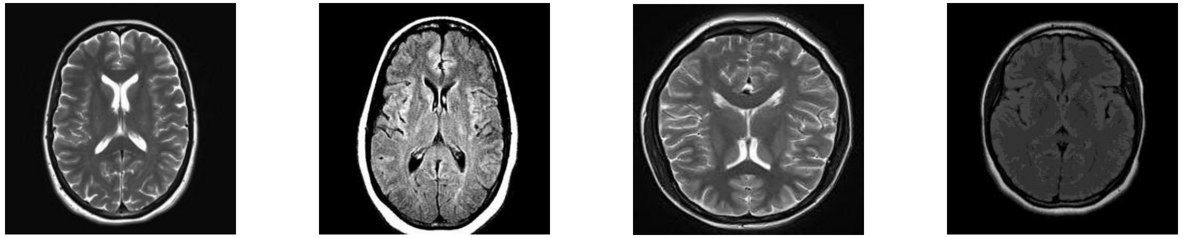
- NO - no tumor, encoded as 0

# Dataset Description

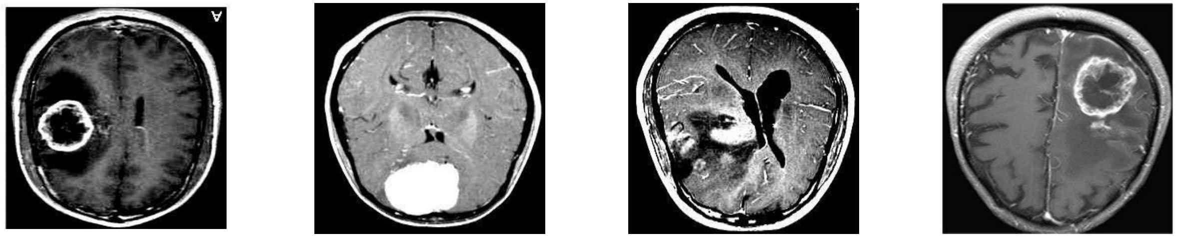
## Quick explanation of the dataset

| Folder | Description |
|---|---|
| No | The folder no contains 1500 Brain MRI Images that are non-tumorous |
| Yes | The folder yes contains 1500 Brain MRI Images that are tumorous |

No Tumor Samples



Tumor Samples



# Main Objective of the analysis:

In thisanalysis we will explore the MRI Brain images

dataset in more details to approach a

robust deep learning model aims to help doctors over the world in brain tumors diagnosis.

The selected model should be robust enough to detect tumors in the brain since there is no room for many errors in this delicate field.

# Exploratory Data Analysis (EDA) + Feature Engineering Section

# Exploratory Data Analysis

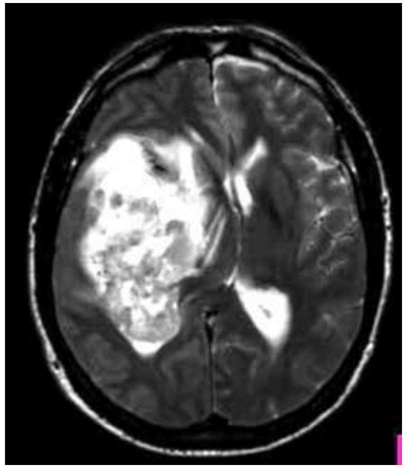
- Identifying images types and dimension:

- Images type : JPG
- Dimensions : (width, height, 3)

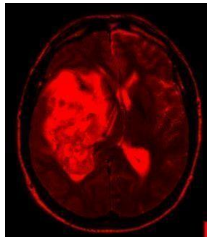
324

- Image sample: “y652.jpg” pixels
- (324, 278, 3)

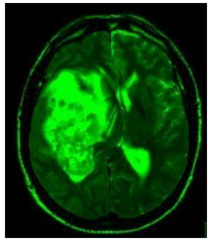
y652.jpg



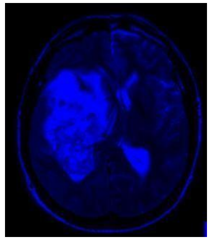
278 pixels



R – Channel



G – Channel



B – Channel

# Exploratory Data Analysis

Here we will compare between the classes

| | image_label | image_width | image_height |
|---|---|---|---|
| 0 | no0 | 630 | 630 |
| 1 | no1 | 198 | 150 |
| 2 | no10 | 225 | 225 |
| 3 | no100 | 217 | 232 |
| 4 | no1000 | 194 | 259 |
| ... | ... | ... | ... |
| 1495 | no995 | 221 | 228 |
| 1496 | no996 | 225 | 225 |
| 1497 | no997 | 225 | 225 |
| 1498 | no998 | 225 | 225 |
| 1499 | no999 | 168 | 300 |

1500 rows × 3 columns

No-Tumor images information

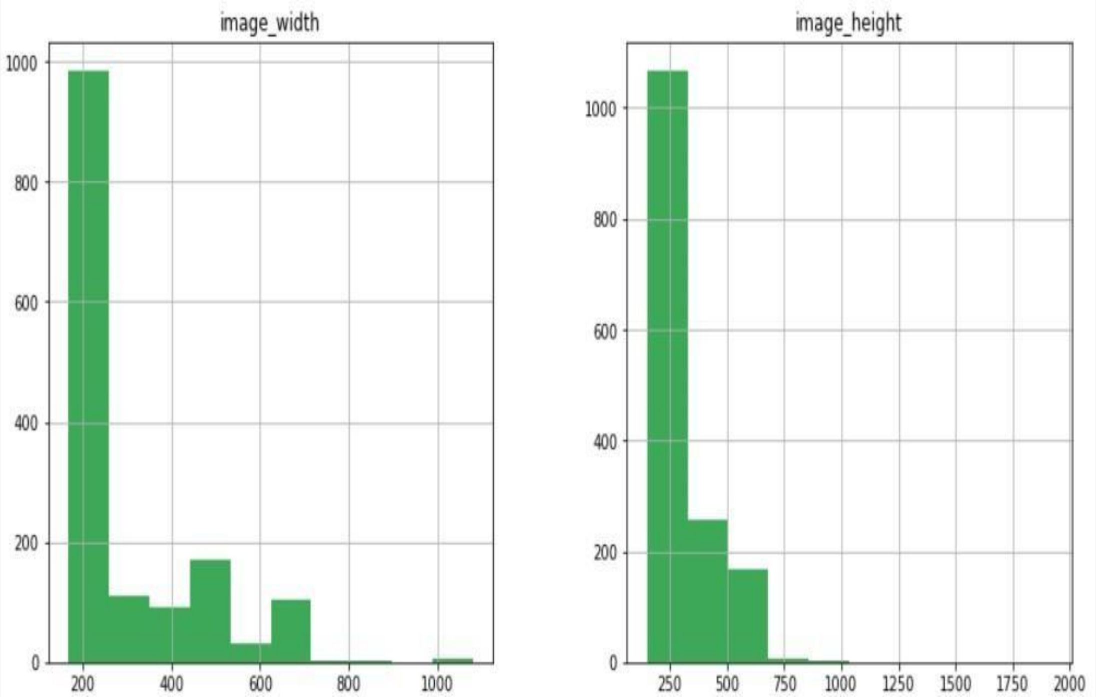
| | image_label | image_width | image_height |
|---|---|---|---|
| 0 | y0 | 348 | 287 |
| 1 | y1 | 630 | 587 |
| 2 | y10 | 879 | 766 |
| 3 | y100 | 630 | 630 |
| 4 | y1000 | 336 | 264 |
| ... | ... | ... | ... |
| 1495 | y995 | 334 | 283 |
| 1496 | y996 | 354 | 303 |
| 1497 | y997 | 348 | 297 |
| 1498 | y998 | 1200 | 1059 |
| 1499 | y999 | 316 | 270 |

1500 rows × 3 columns

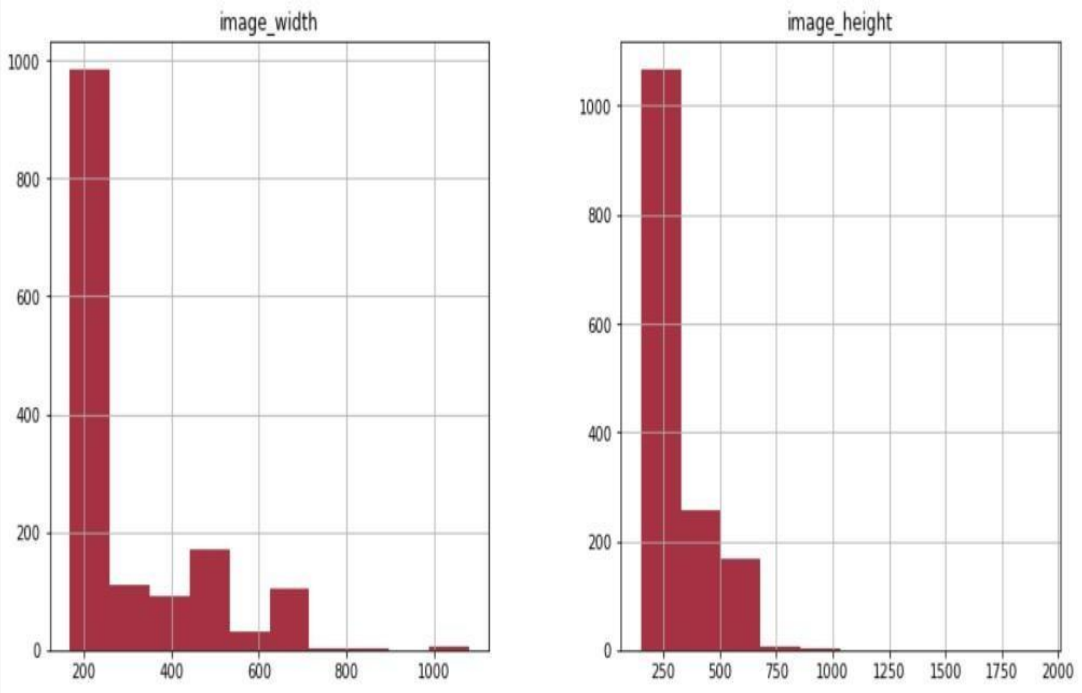
Tumor images information

# Exploratory Data Analysis

**Here we compare between images widths and heights in both classes**




No–Tumor images [width, height] distribution




Tumor images [width, height] distribution

# Feature Engineering

## Images Resizing

Since we have images with different dimensions, we must uniform all the dimensions due to the architecture of deep learning models :

- Input shape = 64 pixels
- After reshaping every image, we store them in a NumPy array

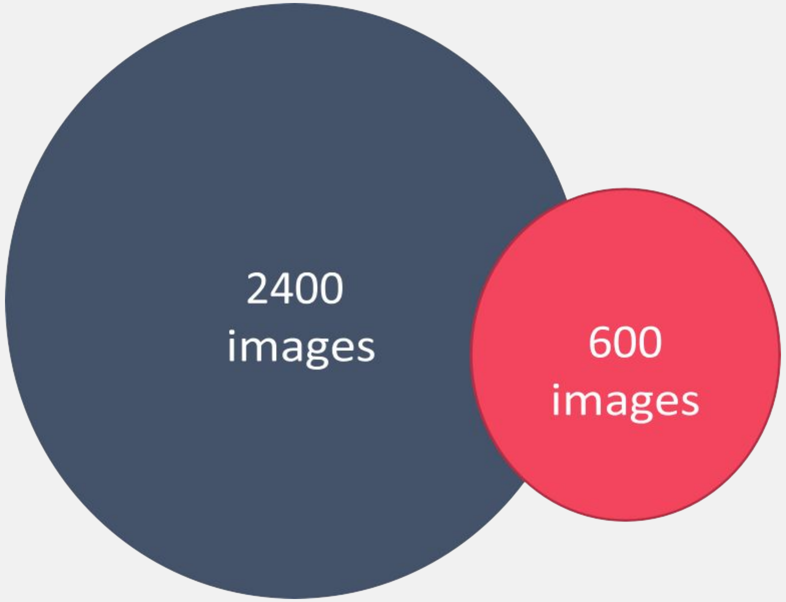
64 x 64 x 3

# Feature Engineering

## Dataset Splitting

As we mentioned before in this presentation, we have in total 3000 images from this point we are going to split these images into two sets 80% for training set and 20% testing set

- Training set: 2400 images
- Testing set: 600 images



# Machine Learning Analysis & Findings

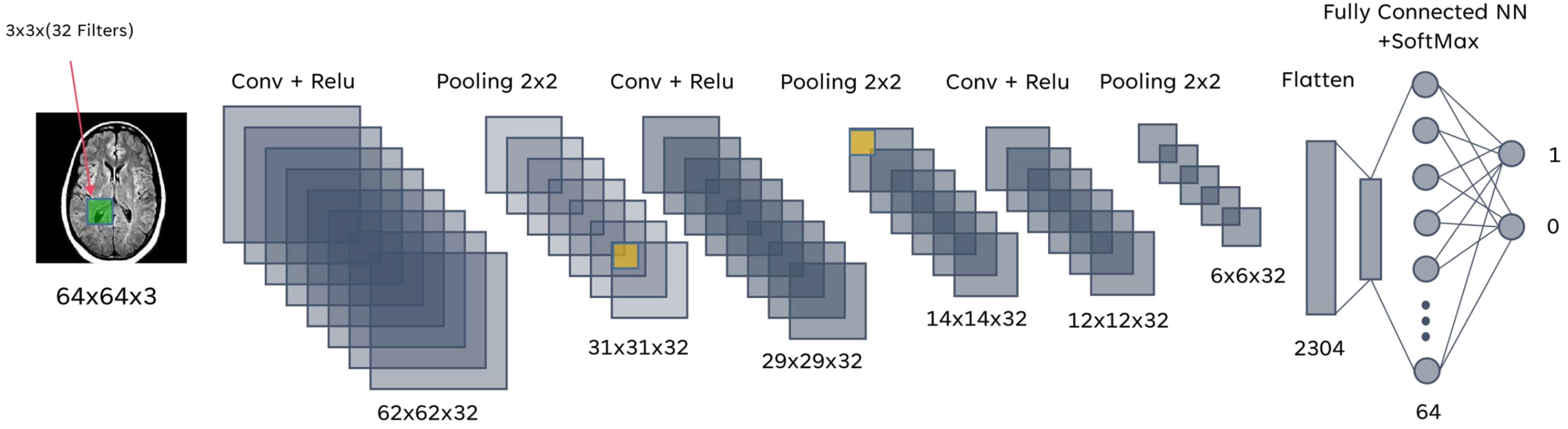
# Machine Learning Analysis & Findings

In the following slides we will compare between 2 different convolutional neural networks (CNN) one is based on Binary Cross Entropy and sigmoid function and the second one is based on Categorical Cross Entropy and SoftMax function for the final prediction layer.

These two models aim to classify MRI brain images to distinguish between the images that contain tumors  and those that don't , for the sake of helping doctors in the diagnostic processes in the healthcare sector.

# Machine Learning Analysis

## Model 01 : CNN Categorical Cross Entropy Based & SoftMax Function.



# Machine Learning Analysis

## Model 01 : CNN Binary Cross Entropy Based & SoftMax Function.

## Model 01 architecture & total number of parameters

```
dense_7 (Dense)              (None, 2)              130
_________________________________________________________________
activation_19 (Activation)   (None, 2)              0
=================================================================
Total params: 176,290
Trainable params: 176,290
Non-trainable params: 0
```

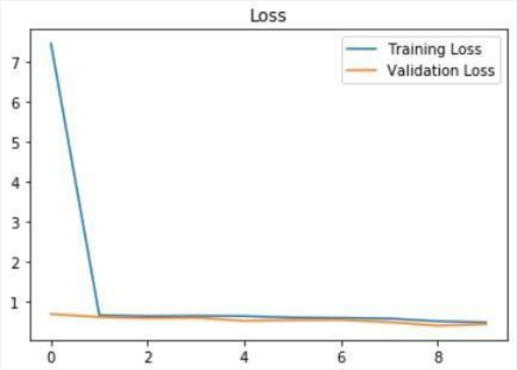
# Machine Learning Findings

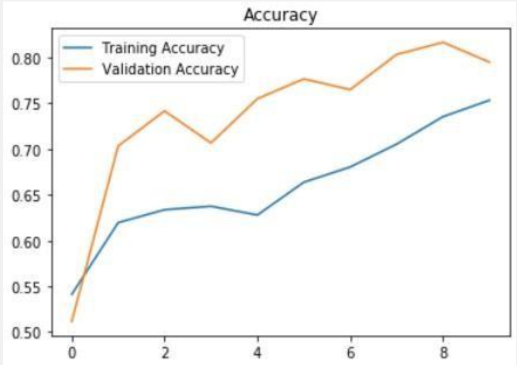
## Model 01 Training

```
model_1.fit(x_train, y_train, batch_size=32, verbose=True, epochs=10,
            validation_data=(x_test, y_test), shuffle=False)

model_1.save("BrainTumorCategorical10Epochs.h5")
```

```
Epoch 1/10
75/75 [==============================] - 1s 15ms/step - loss: 7.4719 - accuracy: 0.5412 - val_loss: 0.6770 - val_accuracy: 0.511
Epoch 2/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6492 - accuracy: 0.6196 - val_loss: 0.6024 - val_accuracy: 0.7033
Epoch 3/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6263 - accuracy: 0.6338 - val_loss: 0.5744 - val_accuracy: 0.7417
Epoch 4/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6339 - accuracy: 0.6375 - val_loss: 0.5849 - val_accuracy: 0.7067
Epoch 5/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6251 - accuracy: 0.6279 - val_loss: 0.5065 - val_accuracy: 0.7550
Epoch 6/10
75/75 [==============================] - 1s 8ms/step - loss: 0.5902 - accuracy: 0.6637 - val_loss: 0.5243 - val_accuracy: 0.7767
Epoch 7/10
75/75 [==============================] - 1s 8ms/step - loss: 0.5813 - accuracy: 0.6804 - val_loss: 0.5312 - val_accuracy: 0.7650
Epoch 8/10
75/75 [==============================] - 1s 8ms/step - loss: 0.5651 - accuracy: 0.7054 - val_loss: 0.4740 - val_accuracy: 0.8033
Epoch 9/10
75/75 [==============================] - 1s 7ms/step - loss: 0.4967 - accuracy: 0.7354 - val_loss: 0.3874 - val_accuracy: 0.8167
Epoch 10/10
75/75 [==============================] - 1s 7ms/step - loss: 0.4699 - accuracy: 0.7533 - val_loss: 0.4243 - val_accuracy: 0.7950
```



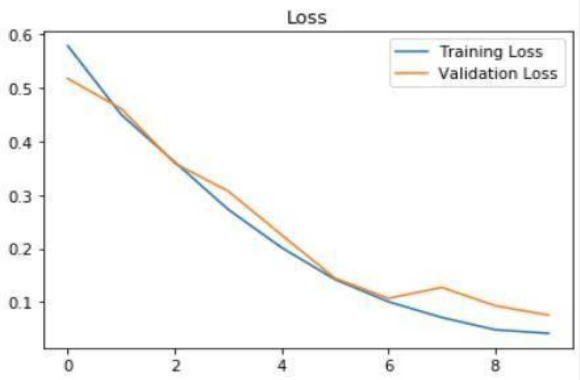

| Set | Accuracy | Losses |
|---|---|---|
| Training | 73.54 % | 0.4967 |
| Validation | 81.67 % | 0.3874 |

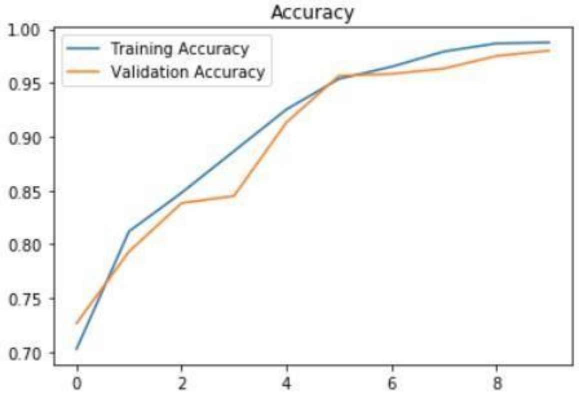
# Machine Learning Findings

## Model 02 Training

```
model_2.fit(x_train, y_train, batch_size=32, verbose=True, epochs=10,
            validation_data=(x_test, y_test), shuffle=False)
```

```
Epoch 1/10
75/75 [==============================] - 1s 16ms/step - loss: 0.5786 - accuracy: 0.7029 - val_loss: 0.5175 - val_accuracy: 0.7267
Epoch 2/10
75/75 [==============================] - 1s 8ms/step - loss: 0.4501 - accuracy: 0.8121 - val_loss: 0.4612 - val_accuracy: 0.7933
Epoch 3/10
75/75 [==============================] - 1s 8ms/step - loss: 0.3616 - accuracy: 0.8479 - val_loss: 0.3592 - val_accuracy: 0.8383
Epoch 4/10
75/75 [==============================] - 1s 8ms/step - loss: 0.2731 - accuracy: 0.8867 - val_loss: 0.3073 - val_accuracy: 0.8450
Epoch 5/10
75/75 [==============================] - 1s 8ms/step - loss: 0.2017 - accuracy: 0.9254 - val_loss: 0.2255 - val_accuracy: 0.9133
Epoch 6/10
75/75 [==============================] - 1s 8ms/step - loss: 0.1421 - accuracy: 0.9538 - val_loss: 0.1444 - val_accuracy: 0.9567
Epoch 7/10
75/75 [==============================] - 1s 8ms/step - loss: 0.1005 - accuracy: 0.9650 - val_loss: 0.1069 - val_accuracy: 0.9583
Epoch 8/10
75/75 [==============================] - 1s 9ms/step - loss: 0.0711 - accuracy: 0.9792 - val_loss: 0.1272 - val_accuracy: 0.9633
Epoch 9/10
75/75 [==============================] - 1s 9ms/step - loss: 0.0481 - accuracy: 0.9867 - val_loss: 0.0930 - val_accuracy: 0.9750
Epoch 10/10
75/75 [==============================] - 1s 8ms/step - loss: 0.0414 - accuracy: 0.9875 - val_loss: 0.0755 - val_accuracy: 0.9800
```




| Set | Accuracy | Losses |
|---|---|---|
| Training | 98.75 % | 0.0414 |
| Validation | 98.00 % | 0.0755 |

# Models flaws and strengths and advanced steps

# Models' strengths and flaws

**Models Strengths and Flaws:**

The first CNN model which is based on Categorical Cross Entropy as loss function and SoftMax for final prediction layer was not that accurate where it is achieved 73% on the training set and 81% on the validation set, since we deal with very sensitive field which is tumors diagnosis, we were required to improve our model, so we change the loss function into Binary Cross Entropy and sigmoid function for final prediction layer which is achieved high accuracy compared to the first one which achieved 98.75 % on the training set and 98.0 % on the validation set.

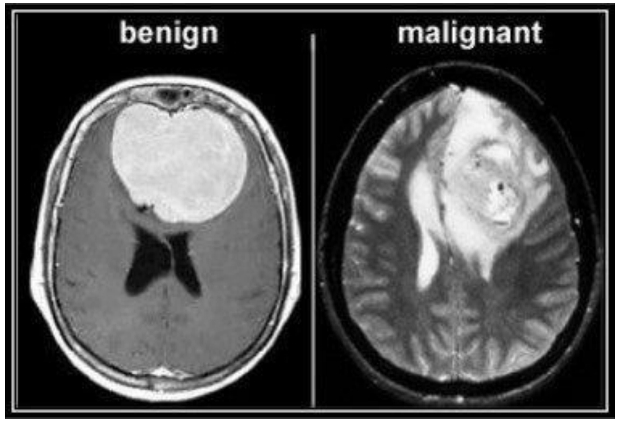
One of main flaws in these models is inability to detect the type of tumor, whether it is Benign Tumor or Malignant Tumor in addition of that it is unable of detecting the location of the tumor which is considered unexplainable model from this point we can improve our model by adding these features in advanced steps in the future.

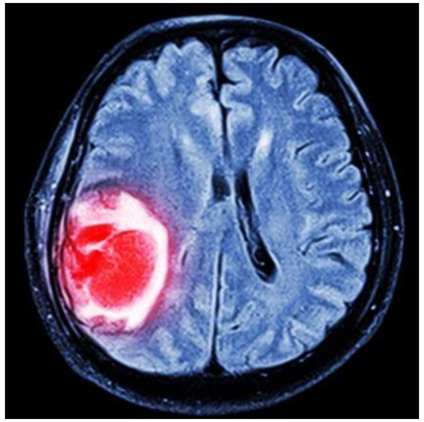
# Advanced steps

further suggestions:

In addition of detecting the tumors in the MRI Images we can provide more additional features :

- Classification of tumor type, whether benign or malignant
- Using advanced object detection algorithm to detect the boundaries of the tumor.
- Deploying the model on smartphone platforms to ease the access of the model to the specialists.



# Thank you

## IBM Machine Learning Professional Certificate

Deep Learning and Reinforcement Learning