Final Project Deep Learning

Detecting Brain Tumors using CNN

By: Mohd Amzad

Contents:

- Dataset Description
- Objectives of the Analysis
- Exploratory Data Analysis
- Feature Engineering
- Training Deep Learning models
- Model Analysis And Findings
- Strength And Flows of Model
- Conclusion

1. Dataset Description

Introduction

What is a brain tumor?

A brain tumor is a mass of abnormal cells in the brain. Your skull, which contains your brain, is extremely inflexible. Any expansion within such a small space can pose complications. Brain tumors can be cancerous (malignant) or benign.

When benign or malignant tumors develop, the pressure inside your skull increases. This has the potential to cause brain damage and perhaps death.

The importance of the subject

Early detection and classification of brain tumors is an important study subject in the field of medical imaging, and thus aids in selecting the most convenient treatment strategy to preserve patients' lives.

Methods

The use of deep learning algorithms in context to improve health diagnosis is yielding significant results. According to the World Health Organization (WHO), effective brain tumor diagnosis include detecting the tumor, determining its location, and classifying it based on malignancy, grade, and type.

The Convolutional Neural Network (CNN)-based multi-task classification is capable of classifying and detecting cancers. A CNN-based model is also used to segment the brain tumor and identify its location.

Quick explanation of the dataset:

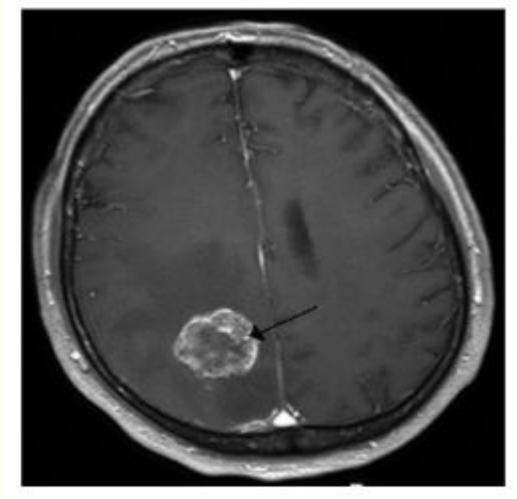


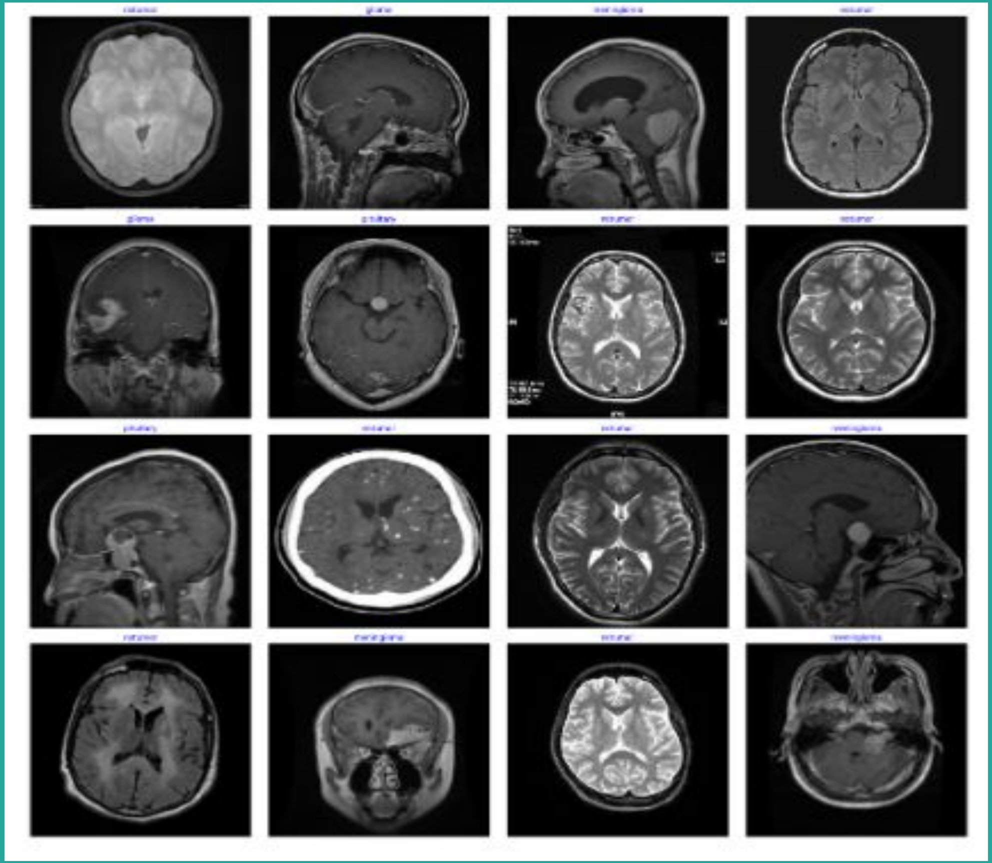
Fig: Brain metastasis in the right cerebral hemisphere from lung cancer, shown on magnetic resonance imaging.

- YES- tumor, encoded as 1

The dataset for this topic involves brain MRI images, which will be utilized to detect brain tumours. It includes two types of MRI scans.

- NO- no tumor is encoded as 0.

Show Samples of the dataset:



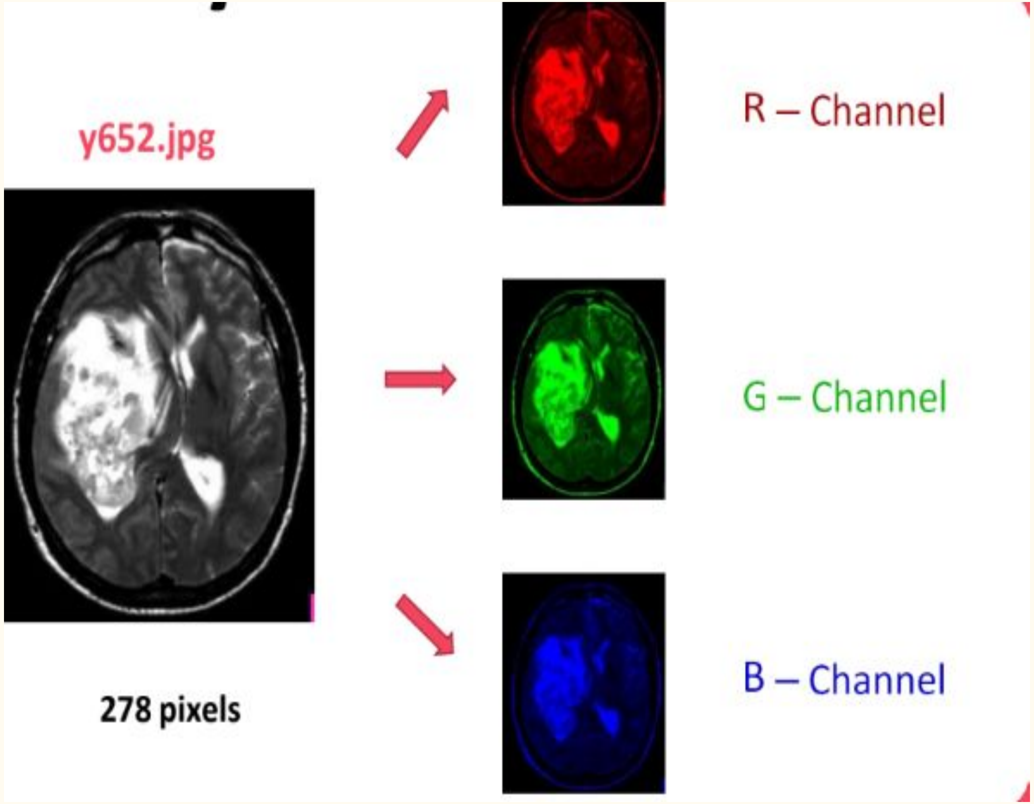
2. Objectives of the Analysis

- The present research will delve deeper into the MRI Brain Images dataset to develop a viable deep learning model for diagnosing brain cancers worldwide.
- The chosen model must be reliable enough to detect brain tumors, as there is no space for errors in this sensitive field.
- To hypertune the model with the simplified hyperparameters to again the better accuracy.

3. Exploratory Data Analysis

Images Types And Dimensions:

- Image type: JPG
- Dimensions:(width=324, height=278, 3)
- Image Sample: “y652.jpg”



Here we compare between the two 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

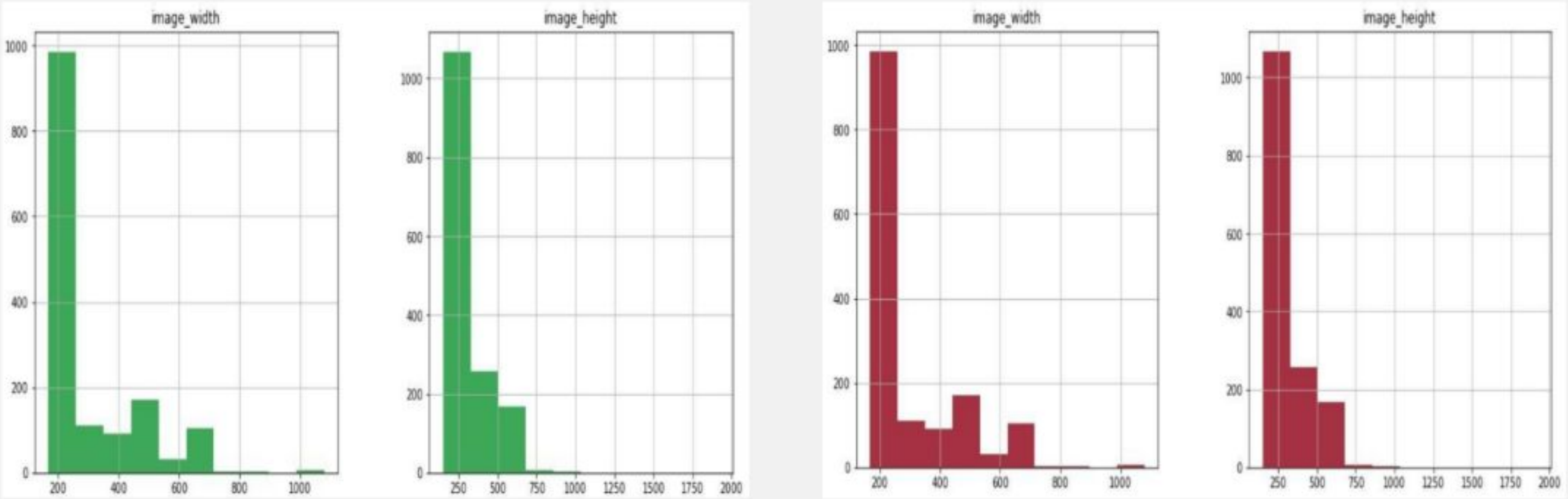
No-Tumor images information

	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

Comparison between the images width and height:



No-Tumor images [width, height] distribution

Tumor images [width, height] distribution

Image Statistics:

	image_width	image_height
count	1500.000000	1500.000000
mean	306.702667	299.980667
std	141.918226	148.211275
min	168.000000	150.000000
25%	225.000000	214.000000
50%	238.000000	227.000000
75%	400.000000	368.000000
max	1080.000000	1920.000000

Tumor images Statistics

	image_width	image_height
count	1500.000000	1500.000000
mean	398.853333	350.455333
std	206.095229	193.916053
min	167.000000	175.000000
25%	294.000000	247.500000
50%	342.000000	283.000000
75%	380.000000	353.000000
max	1427.000000	1275.000000

No-Tumor Images Statistics

4. Feature Engineering

Resizing Image:

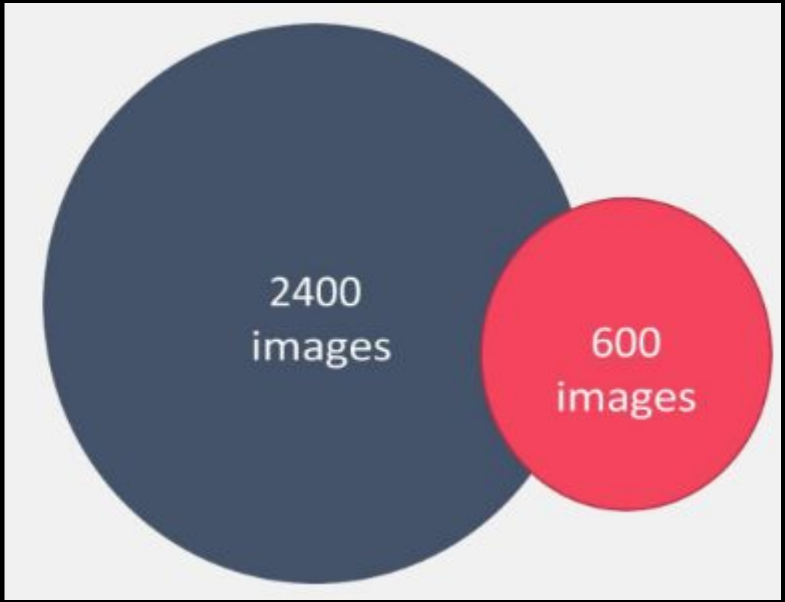
Because the architecture of deep learning models requires us to uniform all dimensions, we must use images with varying dimensions.

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

Dataset split:

As previously stated in this study, we have 3000 images in total. From this point forward, we will divide the images into two sets: 80% for training and 20% for testing.

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

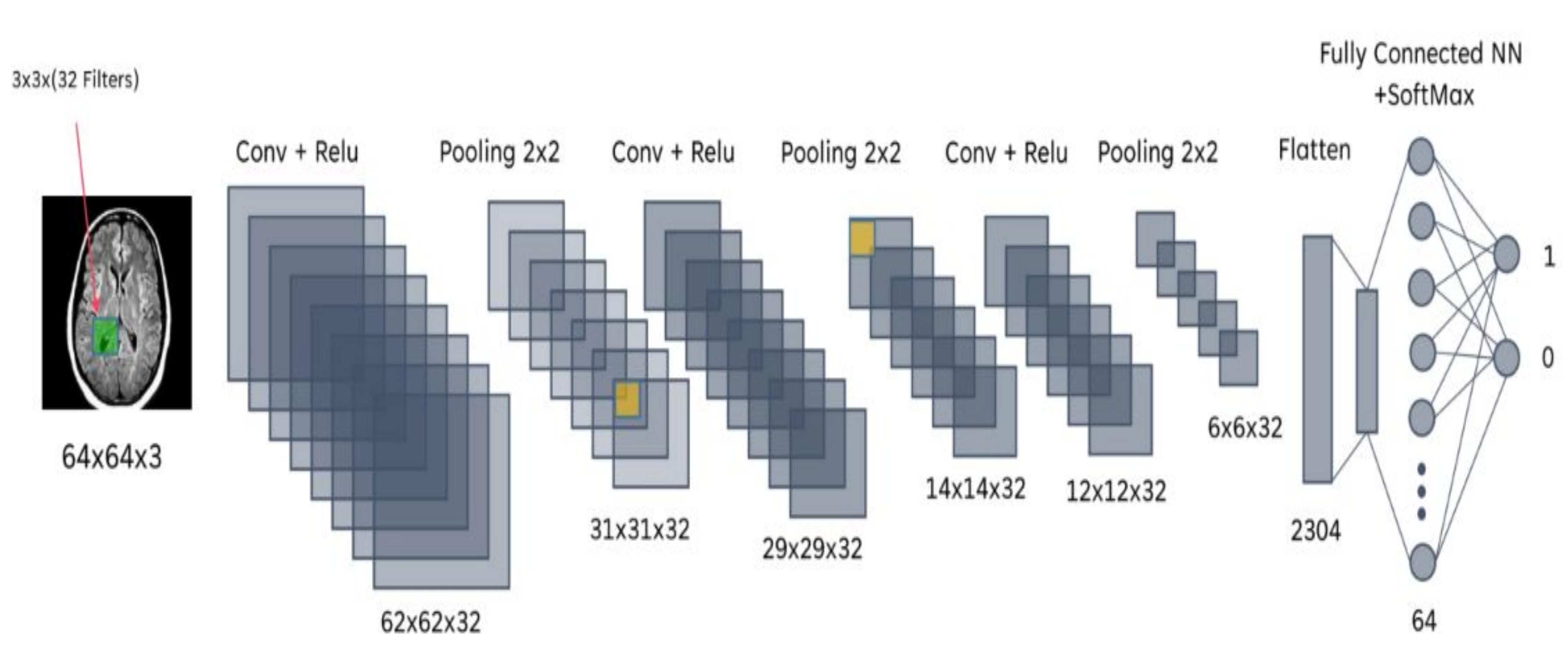


5. Training Deep Learning Models

In the next slides, we will examine two distinct convolutional neural networks (CNN), one based on Binary Cross Entropy and the sigmoid function, and the other based on Categorical Cross Entropy and the 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.

Model 1:CNN Categorical cross entropy based and softmax function



Model 1: Architecture and total 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

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 62, 62, 32)	896
activation_15 (Activation)	(None, 62, 62, 32)	0
max_pooling2d_9 (MaxPooling2	(None, 31, 31, 32)	0
conv2d_10 (Conv2D)	(None, 29, 29, 32)	9248
activation_16 (Activation)	(None, 29, 29, 32)	0
max_pooling2d_10 (MaxPooling	(None, 14, 14, 32)	0
conv2d_11 (Conv2D)	(None, 12, 12, 64)	18496
activation_17 (Activation)	(None, 12, 12, 64)	0
max_pooling2d_11 (MaxPooling	(None, 6, 6, 64)	0
flatten_3 (Flatten)	(None, 2304)	0
dense_6 (Dense)	(None, 64)	147520
activation_18 (Activation)	(None, 64)	0
dropout_3 (Dropout)	(None, 64)	0

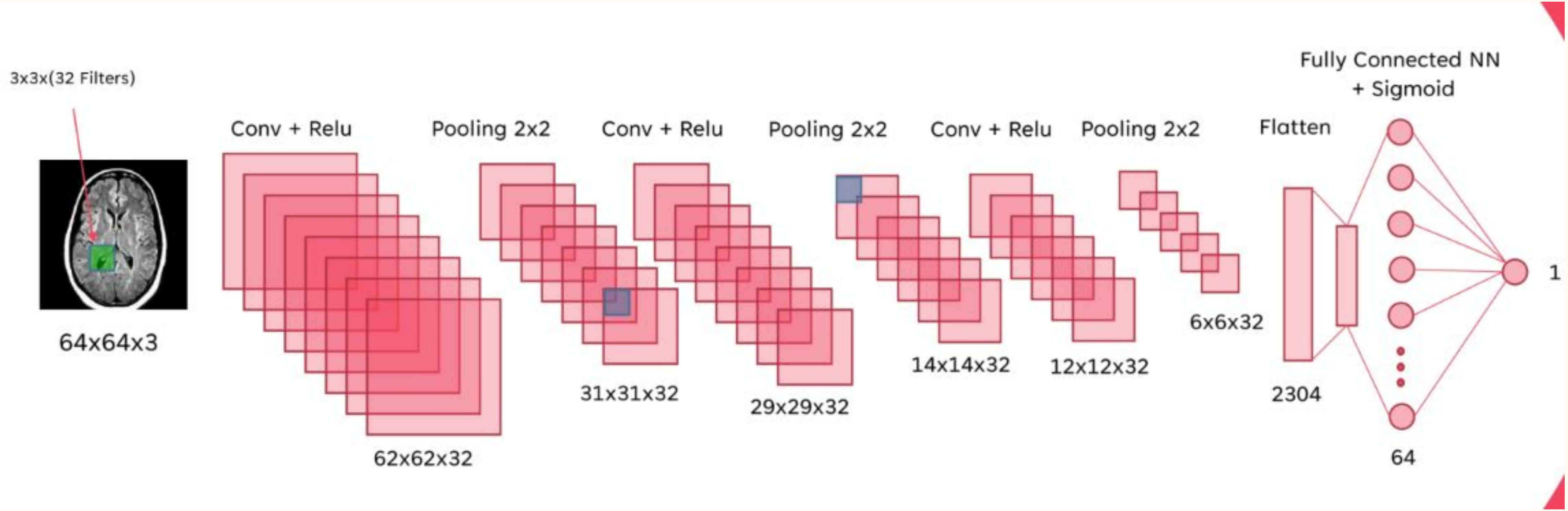
Machine Learning findings Model 1:

```
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: 0.703
Epoch 3/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6263 - accuracy: 0.6338 - val_loss: 0.5744 - val_accuracy: 0.741
Epoch 4/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6339 - accuracy: 0.6375 - val_loss: 0.5849 - val_accuracy: 0.706
Epoch 5/10
75/75 [==============================] - 1s 8ms/step - loss: 0.6251 - accuracy: 0.6279 - val_loss: 0.5065 - val_accuracy: 0.755
Epoch 6/10
75/75 [==============================] - 1s 8ms/step - loss: 0.5902 - accuracy: 0.6637 - val_loss: 0.5243 - val_accuracy: 0.776
Epoch 7/10
75/75 [==============================] - 1s 8ms/step - loss: 0.5813 - accuracy: 0.6804 - val_loss: 0.5312 - val_accuracy: 0.765
Epoch 8/10
75/75 [==============================] - 1s 8ms/step - loss: 0.5651 - accuracy: 0.7054 - val_loss: 0.4740 - val_accuracy: 0.803
Epoch 9/10
75/75 [==============================] - 1s 7ms/step - loss: 0.4967 - accuracy: 0.7354 - val_loss: 0.3874 - val_accuracy: 0.816
Epoch 10/10
75/75 [==============================] - 1s 7ms/step - loss: 0.4699 - accuracy: 0.7533 - val_loss: 0.4243 - val_accuracy: 0.795
```

Model 2:CNN Binary cross entropy based and softmax function



Model 2: Architecture and total parameters

dense_4 (Dense)	(None, 1)	65
activation_16 (Activation)	(None, 1)	0

Total params: 176,225

Trainable params: 176,225

Non-trainable params: 0

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 62, 62, 32)	896
activation_12 (Activation)	(None, 62, 62, 32)	0
max_pooling2d_9 (MaxPooling2	(None, 31, 31, 32)	0
conv2d_10 (Conv2D)	(None, 29, 29, 32)	9248
activation_13 (Activation)	(None, 29, 29, 32)	0
max_pooling2d_10 (MaxPooling	(None, 14, 14, 32)	0
conv2d_11 (Conv2D)	(None, 12, 12, 64)	18496
activation_14 (Activation)	(None, 12, 12, 64)	0
max_pooling2d_11 (MaxPooling	(None, 6, 6, 64)	0
flatten_3 (Flatten)	(None, 2304)	0
dense_3 (Dense)	(None, 64)	147520
activation_15 (Activation)	(None, 64)	0
dropout_3 (Dropout)	(None, 64)	0

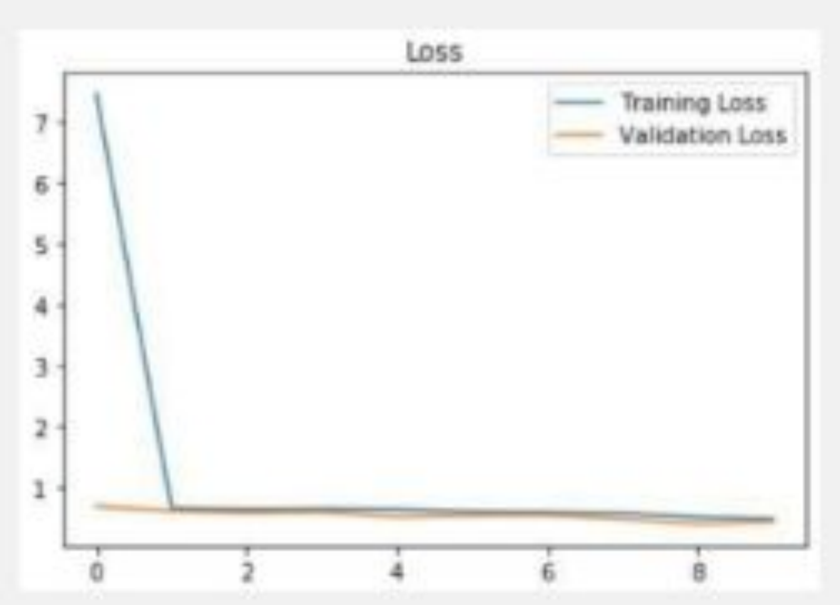
Machine Learning findings Model 2:

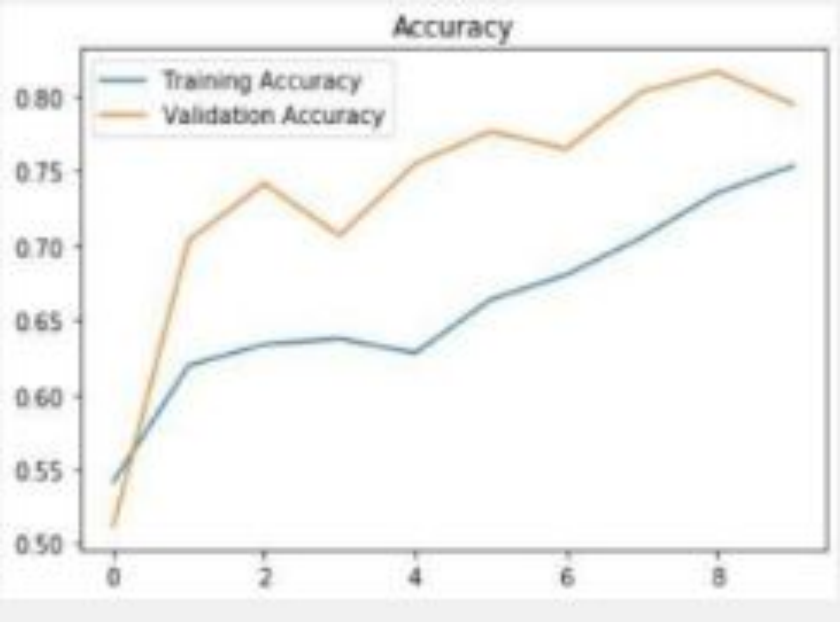
```
model_2.fit(x_train, y_train, batch_size=32, verbose=True, epochs=10,
            validation_data=(x_test, y_test), shuffle=False)
```

```
Epoch 1/10
75/75 [==============================] - 1s 16ms/step - loss: 0.5786 - accuracy: 0.7029 - val_loss: 0.5175 - val_accuracy: 0.726
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
```

6. Model Analysis And findings

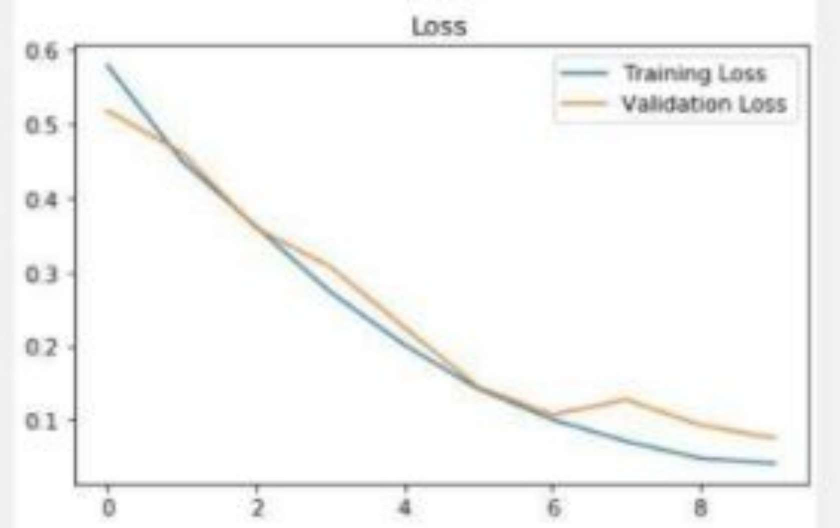
Plotting the loss and accuracy for model 1:





Set	Accuracy	Losses
Training	73.54 %	0.4967
Validation	81.67 %	0.3874

Plotting the loss and accuracy for model 2:



Set	Accuracy	Losses
Training	98.75 %	0.0414
Validation	98.00 %	0.0755

7 .Strength and flows of Model

Since the diagnosis of tumors is a highly sensitive field, we had to improve our model. To do this, we changed the loss function from Categorical Cross Entropy to Binary Cross Entropy and used a sigmoid function for the final prediction layer. This resulted in high accuracy compared to the first model, which achieved 98.75% on the training set and 98.0% on the validation set. The first CNN model was based on SoftMax for the final prediction layer and 73% on the training set and 81% on the validation set.

These models' primary shortcomings include their inability to distinguish between benign and malignant tumors, as well as their inability to determine the tumor's location, which makes them unexplainable. Nevertheless, we can advance our model by incorporating these features in the future.

8.Conclusion

The quantitative, domain-specific data acquired through these studies will improve our understanding of brain toxicity and cognitive decline associated with radiation dosage to non-targeted tissue and can provide the basis for evidence-based cognition-sparing brain radiotherapy.

Treatments and better outcomes for primary brain tumors have long lagged behind those of other tumors. However, a new era in neuro-oncology has emerged,with major advances in both cancer and CNS immunology, and progress in genomics.

Thank you