In [3]:
```
import tensorflow as tf
import numpy as np
import os
```

In [4]:
```
directory = os.path.join(os.getcwd(), "Training")
test_directory = os.path.join(os.getcwd(), "Testing")
img_size = 224
batch = 64
```

In [18]:
```
train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
    validation_split = 0.1)

test_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale = 1./255,
    validation_split = 0.1)

train_datagen = train_datagen.flow_from_directory(
    directory,
    target_size = (img_size, img_size),
    batch_size = batch,
    subset = "training"
)

test_datagen = test_datagen.flow_from_directory(
    test_directory,
    target_size = (img_size, img_size),
    batch_size = batch,
    subset = "validation"
)
```

```
Found 2585 images belonging to 4 classes.
Found 38 images belonging to 4 classes.
```

In [11]:
```
nueral_net = tf.keras.Sequential()
nueral_net.add(tf.keras.layers.Conv2D(filters=64,padding='same',strides=2,kernel_size=
nueral_net.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
nueral_net.add(tf.keras.layers.Conv2D(filters=32,padding='same',strides=2,kernel_size=
nueral_net.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
nueral_net.add(tf.keras.layers.Conv2D(filters=32,padding='same',strides=2,kernel_size=
nueral_net.add(tf.keras.layers.MaxPool2D(pool_size=2))
nueral_net.add(tf.keras.layers.Flatten())
nueral_net.add(tf.keras.layers.Dense(4,activation='softmax'))
```

In [12]:
```
nueral_net.compile(optimizer=tf.keras.optimizers.Adam(),loss='categorical_crossentropy
```

In [13]:
```
accuracy = 0.85
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('acc') is not None and logs.get('acc') > 0.85):
            print("Reached 85% accuracy")
            self.model.stop_training = True
callback = myCallback()
```

In [19]:
```
nueral_net.fit(train_datagen,epochs=20,validation_data=test_datagen, callbacks = [call
```

```
Epoch 1/20
41/41 [==============================] - 28s 677ms/step - loss: 0.5003 - accuracy: 0.
7992 - val_loss: 3.0208 - val_accuracy: 0.3158
Epoch 2/20
41/41 [==============================] - 28s 675ms/step - loss: 0.4994 - accuracy: 0.
7934 - val_loss: 2.8316 - val_accuracy: 0.4211
Epoch 3/20
41/41 [==============================] - 28s 675ms/step - loss: 0.4712 - accuracy: 0.
8116 - val_loss: 2.8557 - val_accuracy: 0.5000
Epoch 4/20
41/41 [==============================] - 28s 673ms/step - loss: 0.4425 - accuracy: 0.
8248 - val_loss: 3.2026 - val_accuracy: 0.3947
Epoch 5/20
41/41 [==============================] - 28s 674ms/step - loss: 0.4234 - accuracy: 0.
8398 - val_loss: 2.8860 - val_accuracy: 0.4211
Epoch 6/20
41/41 [==============================] - 28s 674ms/step - loss: 0.4018 - accuracy: 0.
8445 - val_loss: 2.3442 - val_accuracy: 0.4737
Epoch 7/20
41/41 [==============================] - 28s 675ms/step - loss: 0.4513 - accuracy: 0.
8244 - val_loss: 3.0531 - val_accuracy: 0.4211
Epoch 8/20
41/41 [==============================] - 28s 675ms/step - loss: 0.4048 - accuracy: 0.
8429 - val_loss: 2.9680 - val_accuracy: 0.4211
Epoch 9/20
41/41 [==============================] - 28s 674ms/step - loss: 0.4102 - accuracy: 0.
8387 - val_loss: 3.0161 - val_accuracy: 0.4474
Epoch 10/20
41/41 [==============================] - 28s 678ms/step - loss: 0.3911 - accuracy: 0.
8460 - val_loss: 3.2977 - val_accuracy: 0.5000
Epoch 11/20
41/41 [==============================] - 28s 673ms/step - loss: 0.4349 - accuracy: 0.
8279 - val_loss: 2.7880 - val_accuracy: 0.4474
Epoch 12/20
41/41 [==============================] - 28s 677ms/step - loss: 0.3844 - accuracy: 0.
8549 - val_loss: 4.4472 - val_accuracy: 0.3947
Epoch 13/20
41/41 [==============================] - 28s 683ms/step - loss: 0.3834 - accuracy: 0.
8584 - val_loss: 2.7253 - val_accuracy: 0.4737
Epoch 14/20
41/41 [==============================] - 28s 680ms/step - loss: 0.3540 - accuracy: 0.
8677 - val_loss: 3.0961 - val_accuracy: 0.4211
Epoch 15/20
41/41 [==============================] - 28s 673ms/step - loss: 0.3392 - accuracy: 0.
8735 - val_loss: 3.5877 - val_accuracy: 0.3158
Epoch 16/20
41/41 [==============================] - 28s 673ms/step - loss: 0.3510 - accuracy: 0.
8673 - val_loss: 3.6116 - val_accuracy: 0.5000
Epoch 17/20
41/41 [==============================] - 28s 678ms/step - loss: 0.3532 - accuracy: 0.
8712 - val_loss: 3.5535 - val_accuracy: 0.4211
Epoch 18/20
41/41 [==============================] - 28s 680ms/step - loss: 0.3296 - accuracy: 0.
8789 - val_loss: 2.8182 - val_accuracy: 0.4474
Epoch 19/20
41/41 [==============================] - 28s 675ms/step - loss: 0.3215 - accuracy: 0.
8739 - val_loss: 3.4626 - val_accuracy: 0.4211
Epoch 20/20
41/41 [==============================] - 28s 678ms/step - loss: 0.3247 - accuracy: 0.
8832 - val_loss: 3.4505 - val_accuracy: 0.4211
```

Out[19]:
```
<keras.callbacks.History at 0x2b64a2e2e00>
```

In [29]:
```
from keras.preprocessing import image
```

In [81]:
```
path = os.path.join(directory, "glioma_tumor")
img_path = os.path.join(path, "gg (198).jpg")
```