In [1]:

```
"""
Tumor Classsification using VGG-19
Author: Amruth Karun M V
Date: 07-Nov-2021
"""

import os
import cv2
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Model
from tensorflow.keras.layers import (
    Dense, Input, Dropout, Flatten,
    Conv2D, MaxPooling2D, BatchNormalization
)

from sklearn import metrics
import matplotlib.pyplot as plt
%matplotlib inline

TRAIN_PATH = "../input/brain-tumor-mri-dataset/Training"
TEST_PATH = "../input/brain-tumor-mri-dataset/Testing"
CLASS_NAMES = ['Glioma', 'Meningioma', 'No-tumor', 'Pituitary']
EPOCHS = 100
BATCH_SIZE = 128
LEARNING_RATE = 0.001
```

```
def plot_sample_images():
    """
    Plots sample images for each class
    Arguments: None
    Returns: Plots sample data
    """

    plt.figure(figsize=(10, 10))
    sample_image_path = ['/glioma/Tr-gl_0010.jpg', '/meningioma/Tr-me_0010.jpg',
                         '/notumor/Tr-no_0010.jpg', '/pituitary/Tr-pi_0010.jpg']
    for i in range(len(CLASS_NAMES)):
        ax = plt.subplot(2, 2, i + 1)
        img = cv2.imread(TRAIN_PATH + sample_image_path[i])
        img = cv2.resize(img, (128, 128))
        plt.imshow(img)
        plt.title(CLASS_NAMES[i])


def load_data(input_path, shuffle=False):
    """
    Loads input data fro directory
    Arguments:
        input_path -- input data path
        shuffle    -- whether data needs to be shuffled or not
    Returns: Data generator
    """

    data_generator = keras.preprocessing.image.ImageDataGenerator()
    data_generator = data_generator.flow_from_directory(directory=input_path, target_size=(224,224),
                                                        shuffle=shuffle, class_mode= "categorical")
```

```
    return data_generator


def load_model():
    """
    Creates a keras VGG-19 model
    Arguments: None
    Returns: VGG-19 Model
    """

    img_input = Input(shape=(224, 224, 3))

    # Block 1
    x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv1')(img_input)
    x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv2')(x)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block1_pool')(x)

    # Block 2
    x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv1')(x)
    x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv2')(x)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block2_pool')(x)

    # Block 3
    x = Conv2D(256, (3, 3), activation='relu', padding='same', name='block3_conv1')(x)
    x = Conv2D(256, (3, 3), activation='relu', padding='same', name='block3_conv2')(x)
    x = Conv2D(256, (3, 3), activation='relu', padding='same', name='block3_conv3')(x)
    x = Conv2D(256, (3, 3), activation='relu', padding='same', name='block3_conv4')(x)
    x = BatchNormalization()(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block3_pool')(x)
```

```
# Block 4
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block4_conv1')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block4_conv2')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block4_conv3')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block4_conv4')(x)
x = BatchNormalization()(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block4_pool')(x)

# Block 5
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block5_conv1')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block5_conv2')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block5_conv3')(x)
x = Conv2D(512, (3, 3), activation='relu', padding='same', name='block5_conv4')(x)
x = BatchNormalization()(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block5_pool')(x)

# Classification block
x = Flatten(name='flatten')(x)
x = Dense(4096, activation='relu', name='fc1')(x)
x = Dropout(0.4)(x)
x = Dense(4096, activation='relu', name='fc2')(x)
x = Dropout(0.4)(x)
x = Dense(4, activation='softmax', name='predictions')(x)

model = Model(img_input, x, name='vgg-19')

model.summary()

opt = Adam(learning_rate=LEARNING_RATE)
model.compile(loss = keras.losses.categorical_crossentropy, optimizer=opt, metrics=['accuracy'])
```

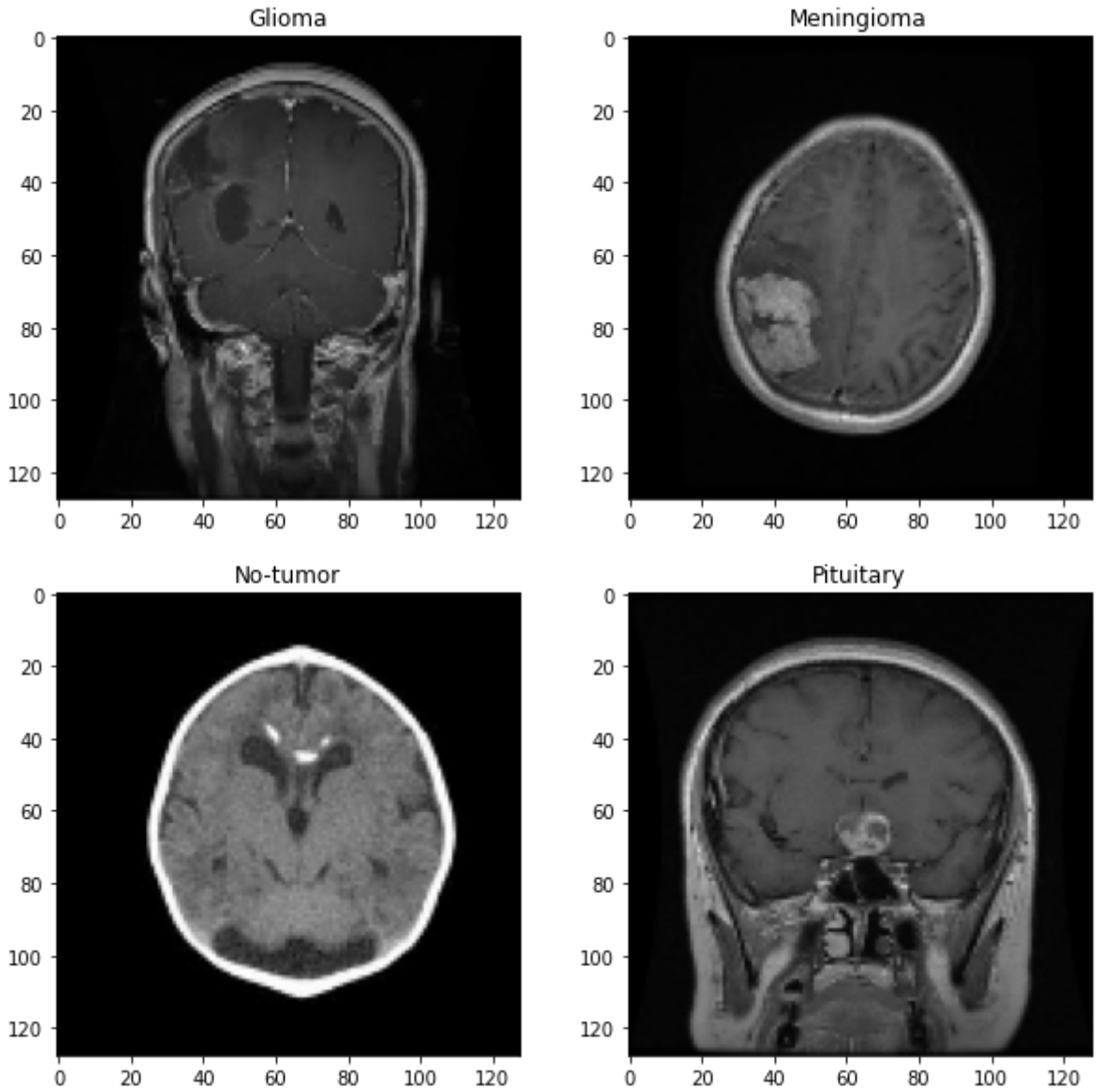
```
    return model


def plot_curves(history):
    """
    Plots loss and accuracy and loss plots for
    training and validation datasets
    Arguments:
        history -- training history
    Returns: None
    """

    plt.plot(history.history['loss'], label="Training loss")
    plt.plot(history.history['val_loss'], label="Validation loss")
    plt.legend()
    plt.title('Training Loss VS Validation Loss')
    plt.show()

    plt.plot(history.history['accuracy'], label="Training accuracy")
    plt.plot(history.history['val_accuracy'], label="Validation accuracy")
    plt.title('Training Accuracy VS Validation Accuracy')
    plt.legend()
    plt.show()


def evaluate_model(model, input_path):
    """
    Evaluates the model and displays
    the confusion matrix
    Arguments:
```

```
        model       -- trained model
        input_path  -- input data path
    Returns: Model score and confusion matrix
    """

    data_generator = load_data(input_path)
    predictions = model.predict(data_generator, BATCH_SIZE)
    y_pred = np.argmax(predictions, axis=1)
    y_true = data_generator.classes

    print("Score = ", model.evaluate(data_generator))
    print("Accuracy = ", metrics.accuracy_score(y_true, y_pred))
    cm = metrics.confusion_matrix(y_true, y_pred)
    metrics.ConfusionMatrixDisplay(cm, display_labels=CLASS_NAMES).plot(cmap=plt.cm.Blues,
                                                                        xticks_rotation='vertical')
    plt.show()


def train_model():
    """
    Trains VGG-19 model and saves the
    trained weights to an H5 file.
    Arguments: None
    Returns: None
    """

    train_generator = load_data(TRAIN_PATH, True)
    val_generator = load_data(TEST_PATH, True)

    # Loads VGG-19 model
    model = load_model()
```

```
    earlystop = keras.callbacks.EarlyStopping(monitor='loss', min_delta=1e-11, patience=10)
    reduce_lr = keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2,
                                                  patience=6, verbose=1)
    model_callbacks = [earlystop, reduce_lr]

    history = model.fit(
        train_generator,
        batch_size=BATCH_SIZE,
        epochs=EPOCHS,
        validation_data=val_generator,
        validation_steps=val_generator.samples//BATCH_SIZE,
        steps_per_epoch=train_generator.samples//BATCH_SIZE,
        callbacks=model_callbacks)

    plot_curves(history)
    model.save_weights("model_vgg19.h5")
    print("Model saved successfully!")

    return model
```

In [2]:

```
plot_sample_images()
```



In [3]:

```
# Train the model
model = train_model()
print("Confusion matrix for train data: ")
evaluate_model(model, TRAIN_PATH)
print("Confusion matrix for val data: ")
evaluate_model(model, TEST_PATH)
```

```
Found 5712 images belonging to 4 classes.
Found 1311 images belonging to 4 classes.
```

```
Model: "vgg-19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 224, 224, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
batch_normalization (BatchNo (None, 224, 224, 64)      256
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
batch_normalization_1 (Batch (None, 112, 112, 128)     512
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv4 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
```

```
batch_normalization_2 (Batch (None, 56, 56, 256)       1024
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv4 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
batch_normalization_3 (Batch (None, 28, 28, 512)       2048
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv4 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
batch_normalization_4 (Batch (None, 14, 14, 512)       2048
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
```

```
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0
_________________________________________________________________
predictions (Dense)          (None, 4)                 16388
=================================================================
Total params: 139,592,516
Trainable params: 139,589,572
Non-trainable params: 2,944
_________________________________________________________________

2021-11-07 08:13:08.730421: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the ML
IR Optimization Passes are enabled (registered 2)

Epoch 1/100

2021-11-07 08:13:11.151784: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005
```

```
44/44 [==============================] - 27s 347ms/step - loss: 18.9244 - accuracy: 0.3956 - val_loss: 107
28.3447 - val_accuracy: 0.3406
Epoch 2/100
44/44 [==============================] - 14s 323ms/step - loss: 1.8303 - accuracy: 0.4780 - val_loss: 67.6
216 - val_accuracy: 0.3156
Epoch 3/100
44/44 [==============================] - 14s 322ms/step - loss: 1.6955 - accuracy: 0.4993 - val_loss: 67.5
509 - val_accuracy: 0.3344
Epoch 4/100
44/44 [==============================] - 16s 356ms/step - loss: 1.0784 - accuracy: 0.5402 - val_loss: 4.29
85 - val_accuracy: 0.4313
Epoch 5/100
44/44 [==============================] - 14s 309ms/step - loss: 0.9957 - accuracy: 0.5888 - val_loss: 11.3
010 - val_accuracy: 0.3781
Epoch 6/100
44/44 [==============================] - 14s 305ms/step - loss: 1.8594 - accuracy: 0.5192 - val_loss: 2.80
99 - val_accuracy: 0.4594
Epoch 7/100
44/44 [==============================] - 13s 304ms/step - loss: 1.1025 - accuracy: 0.5866 - val_loss: 1.05
70 - val_accuracy: 0.5969
Epoch 8/100
44/44 [==============================] - 13s 301ms/step - loss: 0.9950 - accuracy: 0.5724 - val_loss: 0.87
98 - val_accuracy: 0.6875
Epoch 9/100
44/44 [==============================] - 14s 304ms/step - loss: 1.2834 - accuracy: 0.5866 - val_loss: 2.83
24 - val_accuracy: 0.4125
Epoch 10/100
44/44 [==============================] - 13s 298ms/step - loss: 1.2852 - accuracy: 0.6406 - val_loss: 1.12
71 - val_accuracy: 0.5219
Epoch 11/100
```

```
44/44 [==============================] - 13s 299ms/step - loss: 1.0293 - accuracy: 0.6243 - val_loss: 1.27
10 - val_accuracy: 0.4906
Epoch 12/100
44/44 [==============================] - 13s 298ms/step - loss: 0.9163 - accuracy: 0.6406 - val_loss: 1.43
12 - val_accuracy: 0.5875
Epoch 13/100
44/44 [==============================] - 13s 302ms/step - loss: 1.2838 - accuracy: 0.6151 - val_loss: 1.17
62 - val_accuracy: 0.6531
Epoch 14/100
44/44 [==============================] - 13s 298ms/step - loss: 0.8429 - accuracy: 0.6712 - val_loss: 1.49
01 - val_accuracy: 0.4656

Epoch 00014: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
Epoch 15/100
44/44 [==============================] - 13s 298ms/step - loss: 0.7949 - accuracy: 0.6746 - val_loss: 0.81
77 - val_accuracy: 0.6906
Epoch 16/100
44/44 [==============================] - 13s 303ms/step - loss: 0.7275 - accuracy: 0.7230 - val_loss: 0.73
94 - val_accuracy: 0.6781
Epoch 17/100
44/44 [==============================] - 13s 300ms/step - loss: 0.6466 - accuracy: 0.7141 - val_loss: 0.65
46 - val_accuracy: 0.7250
Epoch 18/100
44/44 [==============================] - 13s 300ms/step - loss: 0.6264 - accuracy: 0.7322 - val_loss: 0.57
77 - val_accuracy: 0.7594
Epoch 19/100
44/44 [==============================] - 13s 299ms/step - loss: 0.5872 - accuracy: 0.7450 - val_loss: 0.66
47 - val_accuracy: 0.6938
Epoch 20/100
44/44 [==============================] - 13s 294ms/step - loss: 0.6402 - accuracy: 0.7342 - val_loss: 0.62
28 - val_accuracy: 0.7375
```

```
Epoch 21/100
44/44 [==============================] - 13s 299ms/step - loss: 0.6015 - accuracy: 0.7607 - val_loss: 0.61
61 - val_accuracy: 0.7219
Epoch 22/100
44/44 [==============================] - 13s 303ms/step - loss: 0.5820 - accuracy: 0.7521 - val_loss: 0.60
57 - val_accuracy: 0.7437
Epoch 23/100
44/44 [==============================] - 13s 294ms/step - loss: 0.5488 - accuracy: 0.7708 - val_loss: 0.56
67 - val_accuracy: 0.7250
Epoch 24/100
44/44 [==============================] - 13s 297ms/step - loss: 0.5017 - accuracy: 0.8017 - val_loss: 0.64
81 - val_accuracy: 0.7281
Epoch 25/100
44/44 [==============================] - 13s 297ms/step - loss: 0.5340 - accuracy: 0.7692 - val_loss: 0.52
78 - val_accuracy: 0.7937
Epoch 26/100
44/44 [==============================] - 13s 299ms/step - loss: 0.5419 - accuracy: 0.7756 - val_loss: 0.55
29 - val_accuracy: 0.7594
Epoch 27/100
44/44 [==============================] - 14s 308ms/step - loss: 0.5835 - accuracy: 0.7692 - val_loss: 0.70
31 - val_accuracy: 0.7000
Epoch 28/100
44/44 [==============================] - 13s 298ms/step - loss: 0.5325 - accuracy: 0.7884 - val_loss: 0.57
54 - val_accuracy: 0.7781
Epoch 29/100
44/44 [==============================] - 13s 297ms/step - loss: 0.4807 - accuracy: 0.8118 - val_loss: 0.54
18 - val_accuracy: 0.7688
Epoch 30/100
44/44 [==============================] - 13s 300ms/step - loss: 0.4834 - accuracy: 0.7969 - val_loss: 0.53
49 - val_accuracy: 0.7531
Epoch 31/100
```

```
44/44 [==============================] - 13s 298ms/step - loss: 0.4834 - accuracy: 0.8011 - val_loss: 0.52
32 - val_accuracy: 0.7781
Epoch 32/100
44/44 [==============================] - 13s 301ms/step - loss: 0.4476 - accuracy: 0.8104 - val_loss: 0.51
57 - val_accuracy: 0.7688
Epoch 33/100
44/44 [==============================] - 13s 293ms/step - loss: 0.4554 - accuracy: 0.8089 - val_loss: 0.57
41 - val_accuracy: 0.7656
Epoch 34/100
44/44 [==============================] - 13s 296ms/step - loss: 0.4515 - accuracy: 0.8053 - val_loss: 0.50
34 - val_accuracy: 0.7688
Epoch 35/100
44/44 [==============================] - 13s 300ms/step - loss: 0.4619 - accuracy: 0.7976 - val_loss: 0.54
47 - val_accuracy: 0.7719
Epoch 36/100
44/44 [==============================] - 13s 297ms/step - loss: 0.4471 - accuracy: 0.8203 - val_loss: 0.55
81 - val_accuracy: 0.7625
Epoch 37/100
44/44 [==============================] - 13s 296ms/step - loss: 0.4316 - accuracy: 0.8267 - val_loss: 0.48
50 - val_accuracy: 0.7812
Epoch 38/100
44/44 [==============================] - 13s 297ms/step - loss: 0.4427 - accuracy: 0.8203 - val_loss: 0.51
10 - val_accuracy: 0.7969
Epoch 39/100
44/44 [==============================] - 13s 298ms/step - loss: 0.4593 - accuracy: 0.8189 - val_loss: 0.51
88 - val_accuracy: 0.7750
Epoch 40/100
44/44 [==============================] - 13s 299ms/step - loss: 0.4260 - accuracy: 0.8295 - val_loss: 0.52
90 - val_accuracy: 0.7750
Epoch 41/100
44/44 [==============================] - 13s 300ms/step - loss: 0.4570 - accuracy: 0.8210 - val_loss: 0.45
```

```
01 - val_accuracy: 0.8250
Epoch 42/100
44/44 [==============================] - 13s 296ms/step - loss: 0.4136 - accuracy: 0.8345 - val_loss: 0.44
60 - val_accuracy: 0.8125
Epoch 43/100
44/44 [==============================] - 13s 296ms/step - loss: 0.4211 - accuracy: 0.8317 - val_loss: 0.47
14 - val_accuracy: 0.8062
Epoch 44/100
44/44 [==============================] - 13s 297ms/step - loss: 0.3930 - accuracy: 0.8317 - val_loss: 0.44
40 - val_accuracy: 0.8125
Epoch 45/100
44/44 [==============================] - 13s 297ms/step - loss: 0.4150 - accuracy: 0.8388 - val_loss: 0.50
96 - val_accuracy: 0.7812
Epoch 46/100
44/44 [==============================] - 13s 297ms/step - loss: 0.4074 - accuracy: 0.8303 - val_loss: 0.44
36 - val_accuracy: 0.7906
Epoch 47/100
44/44 [==============================] - 13s 298ms/step - loss: 0.3636 - accuracy: 0.8473 - val_loss: 0.55
71 - val_accuracy: 0.7500
Epoch 48/100
44/44 [==============================] - 13s 295ms/step - loss: 0.3985 - accuracy: 0.8551 - val_loss: 0.47
46 - val_accuracy: 0.8031
Epoch 49/100
44/44 [==============================] - 13s 302ms/step - loss: 0.3733 - accuracy: 0.8423 - val_loss: 0.46
42 - val_accuracy: 0.7656
Epoch 50/100
44/44 [==============================] - 13s 297ms/step - loss: 0.3467 - accuracy: 0.8580 - val_loss: 0.48
12 - val_accuracy: 0.8062
Epoch 51/100
44/44 [==============================] - 13s 296ms/step - loss: 0.4094 - accuracy: 0.8487 - val_loss: 0.53
66 - val_accuracy: 0.7656
```

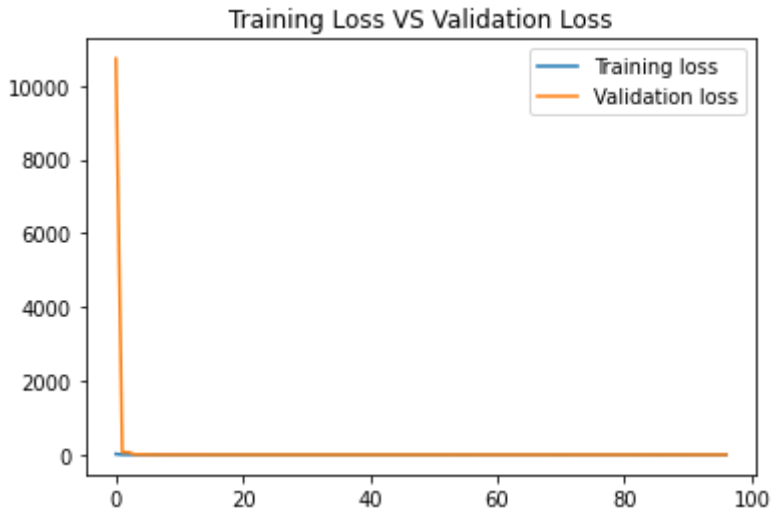
```
Epoch 52/100
44/44 [==============================] - 13s 298ms/step - loss: 0.3562 - accuracy: 0.8565 - val_loss: 0.46
38 - val_accuracy: 0.8281

Epoch 00052: ReduceLROnPlateau reducing learning rate to 4.0000001899898055e-05.
Epoch 53/100
44/44 [==============================] - 13s 296ms/step - loss: 0.3808 - accuracy: 0.8556 - val_loss: 0.41
38 - val_accuracy: 0.8188
Epoch 54/100
44/44 [==============================] - 13s 304ms/step - loss: 0.3350 - accuracy: 0.8693 - val_loss: 0.42
98 - val_accuracy: 0.8031
Epoch 55/100
44/44 [==============================] - 13s 298ms/step - loss: 0.3095 - accuracy: 0.8778 - val_loss: 0.40
45 - val_accuracy: 0.8500
Epoch 56/100
44/44 [==============================] - 13s 302ms/step - loss: 0.3204 - accuracy: 0.8750 - val_loss: 0.44
70 - val_accuracy: 0.8125
Epoch 57/100
44/44 [==============================] - 13s 300ms/step - loss: 0.3184 - accuracy: 0.8693 - val_loss: 0.36
98 - val_accuracy: 0.8438
Epoch 58/100
44/44 [==============================] - 13s 298ms/step - loss: 0.3090 - accuracy: 0.8828 - val_loss: 0.43
46 - val_accuracy: 0.8313
Epoch 59/100
44/44 [==============================] - 13s 297ms/step - loss: 0.3140 - accuracy: 0.8800 - val_loss: 0.37
10 - val_accuracy: 0.8531
Epoch 60/100
44/44 [==============================] - 13s 298ms/step - loss: 0.2934 - accuracy: 0.8951 - val_loss: 0.35
54 - val_accuracy: 0.8500
Epoch 61/100
44/44 [==============================] - 13s 298ms/step - loss: 0.2972 - accuracy: 0.8828 - val_loss: 0.34
```

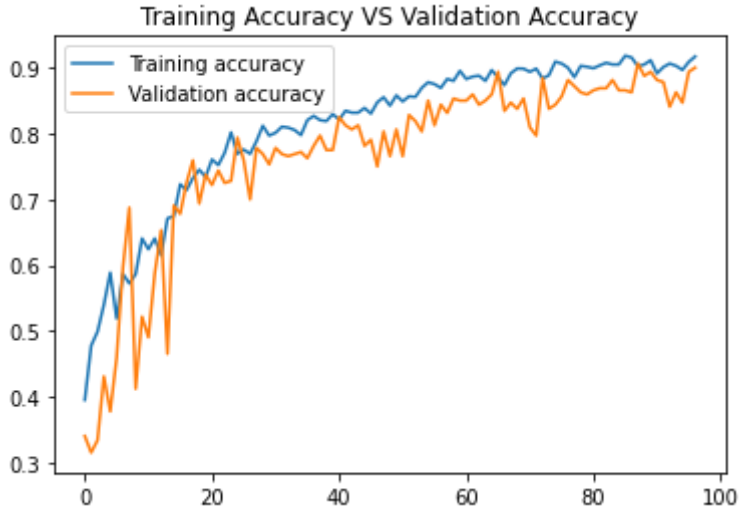
```
40 - val_accuracy: 0.8500
Epoch 62/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2845 - accuracy: 0.8864 - val_loss: 0.34
96 - val_accuracy: 0.8594
Epoch 63/100
44/44 [==============================] - 14s 309ms/step - loss: 0.2934 - accuracy: 0.8878 - val_loss: 0.38
63 - val_accuracy: 0.8438
Epoch 64/100
44/44 [==============================] - 13s 296ms/step - loss: 0.2947 - accuracy: 0.8800 - val_loss: 0.36
78 - val_accuracy: 0.8500
Epoch 65/100
44/44 [==============================] - 13s 299ms/step - loss: 0.2599 - accuracy: 0.8963 - val_loss: 0.31
34 - val_accuracy: 0.8594
Epoch 66/100
44/44 [==============================] - 13s 300ms/step - loss: 0.2691 - accuracy: 0.8849 - val_loss: 0.30
37 - val_accuracy: 0.8938
Epoch 67/100
44/44 [==============================] - 13s 298ms/step - loss: 0.2979 - accuracy: 0.8736 - val_loss: 0.36
40 - val_accuracy: 0.8344
Epoch 68/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2769 - accuracy: 0.8913 - val_loss: 0.36
24 - val_accuracy: 0.8469
Epoch 69/100
44/44 [==============================] - 13s 295ms/step - loss: 0.2567 - accuracy: 0.8987 - val_loss: 0.37
48 - val_accuracy: 0.8375
Epoch 70/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2648 - accuracy: 0.8984 - val_loss: 0.38
37 - val_accuracy: 0.8531
Epoch 71/100
44/44 [==============================] - 13s 300ms/step - loss: 0.2444 - accuracy: 0.8937 - val_loss: 0.49
85 - val_accuracy: 0.8094
```

```
Epoch 72/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2773 - accuracy: 0.8984 - val_loss: 0.42
70 - val_accuracy: 0.7969

Epoch 00072: ReduceLROnPlateau reducing learning rate to 8.000000525498762e-06.
Epoch 73/100
44/44 [==============================] - 13s 301ms/step - loss: 0.2776 - accuracy: 0.8835 - val_loss: 0.31
30 - val_accuracy: 0.8844
Epoch 74/100
44/44 [==============================] - 13s 305ms/step - loss: 0.2672 - accuracy: 0.8885 - val_loss: 0.39
74 - val_accuracy: 0.8375
Epoch 75/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2516 - accuracy: 0.9091 - val_loss: 0.34
46 - val_accuracy: 0.8438
Epoch 76/100
44/44 [==============================] - 13s 303ms/step - loss: 0.2417 - accuracy: 0.9062 - val_loss: 0.31
97 - val_accuracy: 0.8562
Epoch 77/100
44/44 [==============================] - 13s 296ms/step - loss: 0.2447 - accuracy: 0.8999 - val_loss: 0.29
66 - val_accuracy: 0.8813
Epoch 78/100
44/44 [==============================] - 13s 300ms/step - loss: 0.2802 - accuracy: 0.8857 - val_loss: 0.31
49 - val_accuracy: 0.8719
Epoch 79/100
44/44 [==============================] - 13s 301ms/step - loss: 0.2409 - accuracy: 0.9027 - val_loss: 0.34
12 - val_accuracy: 0.8625
Epoch 80/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2369 - accuracy: 0.9013 - val_loss: 0.38
38 - val_accuracy: 0.8594
Epoch 81/100
44/44 [==============================] - 13s 298ms/step - loss: 0.2518 - accuracy: 0.8991 - val_loss: 0.36
```

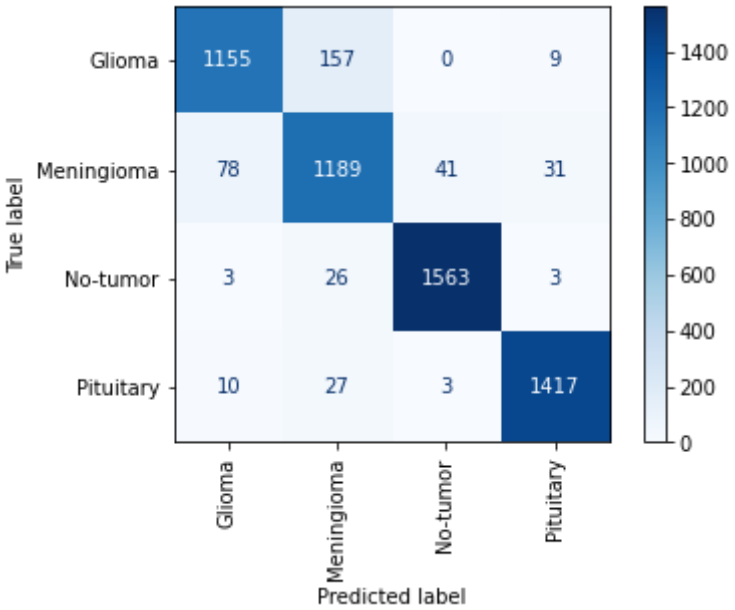
```
63 - val_accuracy: 0.8656
Epoch 82/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2408 - accuracy: 0.9034 - val_loss: 0.30
85 - val_accuracy: 0.8687
Epoch 83/100
44/44 [==============================] - 13s 299ms/step - loss: 0.2283 - accuracy: 0.9073 - val_loss: 0.31
11 - val_accuracy: 0.8687

Epoch 00083: ReduceLROnPlateau reducing learning rate to 1.6000001778593287e-06.
Epoch 84/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2557 - accuracy: 0.9048 - val_loss: 0.30
51 - val_accuracy: 0.8813
Epoch 85/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2346 - accuracy: 0.9048 - val_loss: 0.31
13 - val_accuracy: 0.8656
Epoch 86/100
44/44 [==============================] - 13s 296ms/step - loss: 0.2279 - accuracy: 0.9181 - val_loss: 0.34
02 - val_accuracy: 0.8656
Epoch 87/100
44/44 [==============================] - 13s 298ms/step - loss: 0.2146 - accuracy: 0.9155 - val_loss: 0.29
61 - val_accuracy: 0.8625
Epoch 88/100
44/44 [==============================] - 13s 295ms/step - loss: 0.2615 - accuracy: 0.9020 - val_loss: 0.25
11 - val_accuracy: 0.9062
Epoch 89/100
44/44 [==============================] - 13s 298ms/step - loss: 0.2523 - accuracy: 0.9048 - val_loss: 0.28
64 - val_accuracy: 0.8875
Epoch 90/100
44/44 [==============================] - 13s 296ms/step - loss: 0.2273 - accuracy: 0.9112 - val_loss: 0.28
06 - val_accuracy: 0.8938
Epoch 91/100
```

```
44/44 [==============================] - 13s 300ms/step - loss: 0.2522 - accuracy: 0.8913 - val_loss: 0.29
90 - val_accuracy: 0.8813
Epoch 92/100
44/44 [==============================] - 13s 303ms/step - loss: 0.2485 - accuracy: 0.9009 - val_loss: 0.30
52 - val_accuracy: 0.8781
Epoch 93/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2476 - accuracy: 0.9062 - val_loss: 0.34
75 - val_accuracy: 0.8406
Epoch 94/100
44/44 [==============================] - 13s 302ms/step - loss: 0.2579 - accuracy: 0.9027 - val_loss: 0.32
39 - val_accuracy: 0.8625

Epoch 00094: ReduceLROnPlateau reducing learning rate to 3.200000264769187e-07.
Epoch 95/100
44/44 [==============================] - 13s 296ms/step - loss: 0.2614 - accuracy: 0.8966 - val_loss: 0.37
22 - val_accuracy: 0.8469
Epoch 96/100
44/44 [==============================] - 13s 294ms/step - loss: 0.2509 - accuracy: 0.9088 - val_loss: 0.29
69 - val_accuracy: 0.8938
Epoch 97/100
44/44 [==============================] - 13s 297ms/step - loss: 0.2205 - accuracy: 0.9169 - val_loss: 0.25
75 - val_accuracy: 0.9000
```

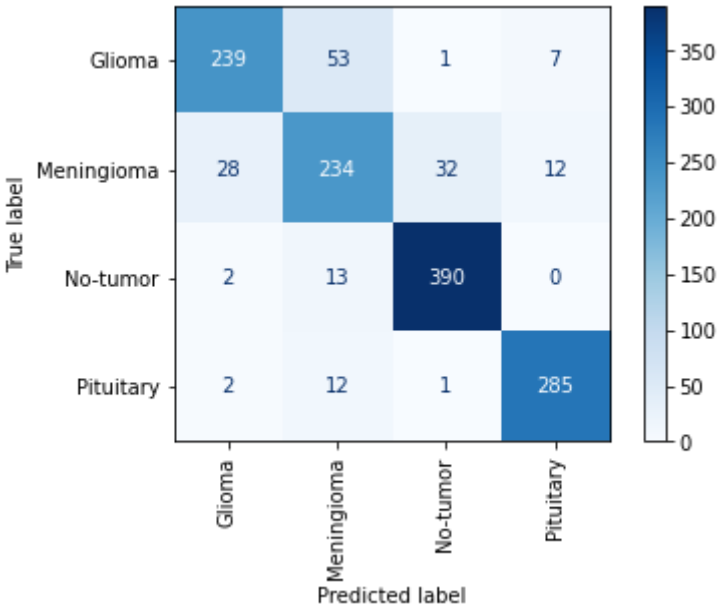

Training Loss VS Validation Loss
Training loss
Validation loss
10000
8000
6000
4000
2000
0
0
20
40
60
80
100


Training Accuracy VS Validation Accuracy
Training accuracy
Validation accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100

```
Model saved successfully!
Confusion matrix for train data:
Found 5712 images belonging to 4 classes.
179/179 [==============================] - 23s 126ms/step - loss: 0.1976 - accuracy: 0.9321
Score =  [0.1975684016942978, 0.9320728182792664]
Accuracy =  0.9320728291316527
```




```
Confusion matrix for val data:
Found 1311 images belonging to 4 classes.
41/41 [==============================] - 5s 111ms/step - loss: 0.3100 - accuracy: 0.8757
Score =  [0.31003186106681824, 0.8756674528121948]
Accuracy =  0.8756674294431731
```


Glioma
Meningioma
No-tumor
Pituitary
True label
239
53
1
7
28
234
32
12
2
13
390
0
2
12
1
285
Glioma
Meningioma
No-tumor
Pituitary
Predicted label
350
300
250
200
150
100
50
0