```
In [15]:  from sklearn.ensemble import RandomForestClassifier
          from sklearn.model_selection import train_test_split
          import matplotlib.pyplot as plt
          from sklearn.metrics import accuracy_score
          from sklearn.metrics import classification_report,confusion_matrix
          from sklearn import metrics
          import numpy as np
          import pandas as pd
          import os
          import cv2
```

```
In [16]:  path = os.listdir('brain_tumor/train') # Path of the dataset
          classes = {'glioma_tumor':0, 'meningioma_tumor':1, 'no_tumor':2, 'pituitary_tumor':3} # Class labels
```

```
In [17]:  X = []
          Y = []
          for cls in classes:
              pth = 'brain_tumor/train/' + cls
              for j in os.listdir(pth):
                  img = cv2.imread(pth+'/'+j, 0) # 0 means gray color
                  img = cv2.resize(img, (200,200)) # resize the image
                  X.append(img)
                  Y.append(classes[cls])
```

```
In [18]:  np.unique(Y)
```

```
Out[18]:  array([0, 1, 2, 3])
```

```
In [19]:  X = np.array(X)
          Y = np.array(Y)
```

```
In [20]:  pd.Series(Y).value_counts() # Show each class label instances
```

```
Out[20]:  3    827
          0    826
          1    822
          2    395
          dtype: int64
```

```
In [21]:  X.shape # Show the total number of images and the Dimensions
```

```
Out[21]:  (2870, 200, 200)
```

```
In [22]:  X_updated = X.reshape(len(X), -1)
          X_updated.shape
```

```
Out[22]:  (2870, 40000)
```

```
In [23]:  xtrain, xtest, ytrain, ytest = train_test_split(X_updated, Y, random_state=10, test_size =.20)
```

```
In [24]:  xtrain.shape, xtest.shape
```

```
Out[24]:  ((2296, 40000), (574, 40000))
```

```
In [25]:  print(xtrain.max(), xtrain.min())
          print(xtest.max(), xtest.min())
          xtrain = xtrain/255
          xtest = xtest/255
          print(xtrain.max(), xtrain.min())
          print(xtest.max(), xtest.min())

          255 0
          255 0
          1.0 0.0
          1.0 0.0
```

```
In [26]:  randomforest = RandomForestClassifier(n_jobs=-1)
          model = randomforest.fit(xtrain, ytrain)
```

```
In [27]:  ypred = model.predict(xtest)
          cm = confusion_matrix(ytest, ypred)
          plt.figure(figsize= (8,8))
          plt.imshow(confusion_matrix(ytest, ypred))
          plt.xticks(np.arange(4), classes)
          plt.yticks(np.arange(4), classes)
          plt.colorbar()
          plt.show()
          print(cm)
```
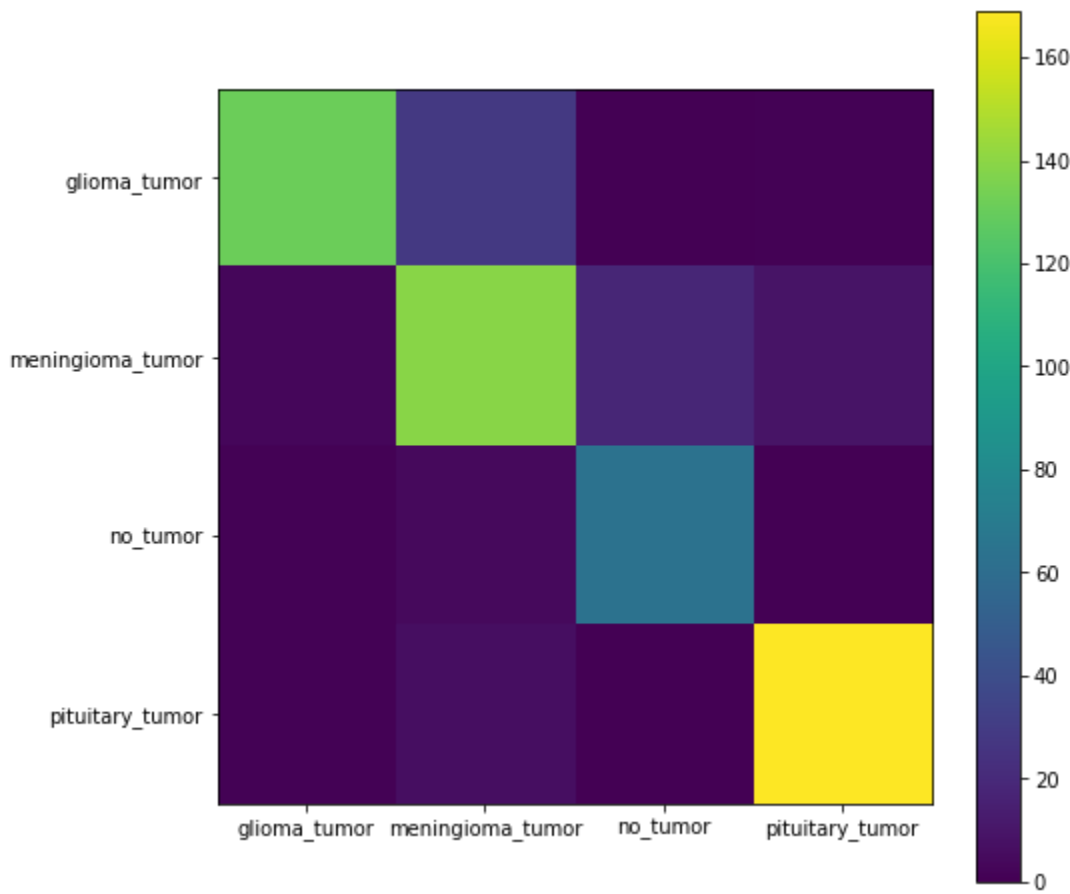


```
          [[131  28   0   1]
           [  3 139  18   9]
           [  1   4  64   0]
           [  1   6   0 169]]
```

```
In [28]:  report = classification_report(ytest, ypred)
          print(report)

                        precision    recall  f1-score   support

                     0       0.96      0.82      0.89       160
                     1       0.79      0.82      0.80       169
                     2       0.78      0.93      0.85        69
                     3       0.94      0.96      0.95       176

              accuracy                           0.88       574
             macro avg       0.87      0.88      0.87       574
          weighted avg       0.88      0.88      0.88       574
```

```
In [29]:  from sklearn.metrics import mean_squared_error

          MSE = mean_squared_error(ytest, ypred)
          print ("MSE:{0}".format(MSE))

          MSE:0.23519163763066203
```

```
In [ ]:
```