

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [2]:

```
import os
path=os.listdir('C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Training')
classes={'no_tumor':0,'pituitary_tumor':1,'glioma_tumor':2,'meningioma_tumor':3}
```

In [3]:

```
import cv2
x=[]
y=[]
for cls in classes:
    path1='C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Training\\'+cls
    for j in os.listdir(path1):
        img=cv2.imread(path1+'/'+j,0)
        img=cv2.resize(img,(200,200))
        x.append(img)
        y.append(classes[cls])
```

In [4]:

```
x=np.array(x)
y=np.array(y)
x_updated=x.reshape(len(x),-1)
```

In [5]:

```
np.unique(y)
```

Out[5]:

```
array([0, 1, 2, 3])
```

In [6]:

```
pd.Series(y).value_counts()
```

Out[6]:

```
1      827
2      826
3      822
0      395
dtype: int64
```

In [7]:

```
x.shape,x_updated.shape
```

Out[7]:

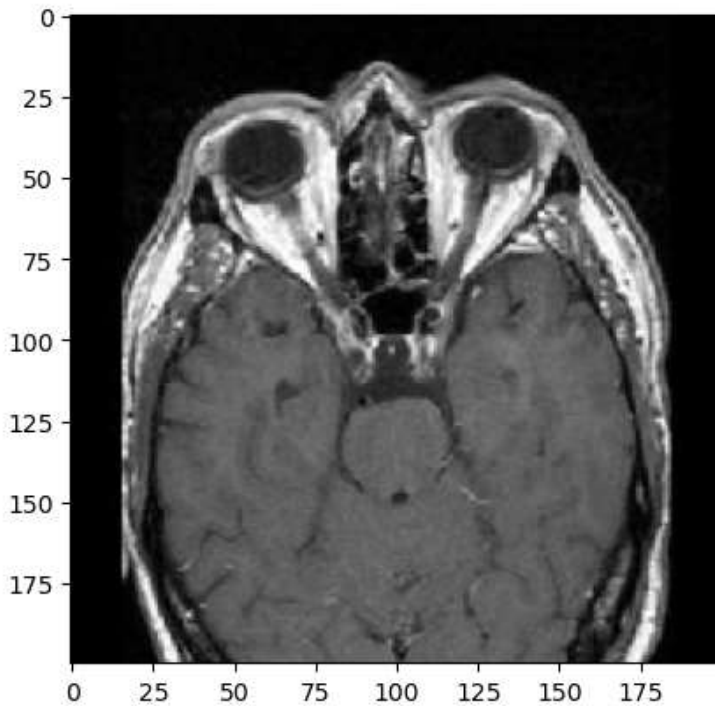
```
((2870, 200, 200), (2870, 40000))
```

In [8]:

```
plt.imshow(x[0],cmap='gray')
```

Out[8]:

<matplotlib.image.AxesImage at 0x256dc78ab20>



In [9]:

```
x_updated=x.reshape(len(x),-1)
x_updated.shape
```

Out[9]:

(2870, 40000)

In [10]:

```
xtrain,xtest,ytrain,ytest=train_test_split(x_updated,y,random_state=10,test_size=.20)
```

In [11]:

```
xtrain.shape,xtest.shape
```

Out[11]:

((2296, 40000), (574, 40000))

In [12]:

```
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 [13]:

```
from sklearn.decomposition import PCA
```

In [14]:

```
print(xtrain.shape,xtest.shape)
pca=PCA(.98)
pca_train=xtrain
pca_test=xtest
```

(2296, 40000) (574, 40000)

In [15]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
```

In [16]:

```
import warnings
warnings.filterwarnings('ignore')
lg=LogisticRegression(C=0.1)
lg.fit(xtrain,ytrain)
```

Out[16]:

LogisticRegression(C=0.1)

In [17]:

```
from sklearn.svm import SVC
sv=SVC()
sv.fit(xtrain,ytrain)
```

Out[17]:

SVC()

In [18]:

```
print("Training score:",lg.score(xtrain,ytrain))
print("Testing score:",lg.score(xtest,ytest))
```

Training score: 1.0

Testing score: 0.759581881533101

In [19]:

```
print("Traing score:",sv.score(xtrain,ytrain))
print("Testing score:",sv.score(xtest,ytest))
```

Traing score: 0.9390243902439024

Testing score: 0.8170731707317073

In [20]:

```
pred=sv.predict(xtest)
```

In [21]:

```
misclassified=np.where(ytest!=pred)
misclassified
```

Out[21]:

```
(array([ 9, 14, 16, 17, 19, 21, 32, 35, 37, 38, 44, 48, 51,
        60, 75, 78, 86, 97, 99, 102, 107, 112, 122, 125, 126, 128,
        129, 130, 142, 146, 150, 152, 155, 160, 164, 166, 174, 183, 187,
        200, 203, 204, 207, 216, 217, 221, 230, 234, 242, 243, 249, 254,
        265, 269, 270, 275, 289, 295, 298, 303, 313, 316, 324, 329, 333,
        338, 351, 356, 359, 370, 372, 379, 380, 386, 390, 394, 401, 414,
        419, 423, 425, 431, 433, 434, 438, 441, 448, 449, 450, 453, 459,
        461, 469, 501, 505, 520, 522, 524, 526, 535, 537, 538, 555, 564,
        565], dtype=int64),)
```

In [22]:

```
print("Total misclassified samples:",len(misclassified[0]))
print(pred[36],ytest[36])
```

```
Total misclassified samples: 105
3 3
```

In [23]:

```
dec={0:'No tumor',1:'Positive tumor-Pituitary tumor',2:'Positive tumor-Glioma tumor',3:'Positive tumor-mel
```

In [38]:

```
os.path.exists('C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Testing\\pituitary_tumor'
```

Out[38]:

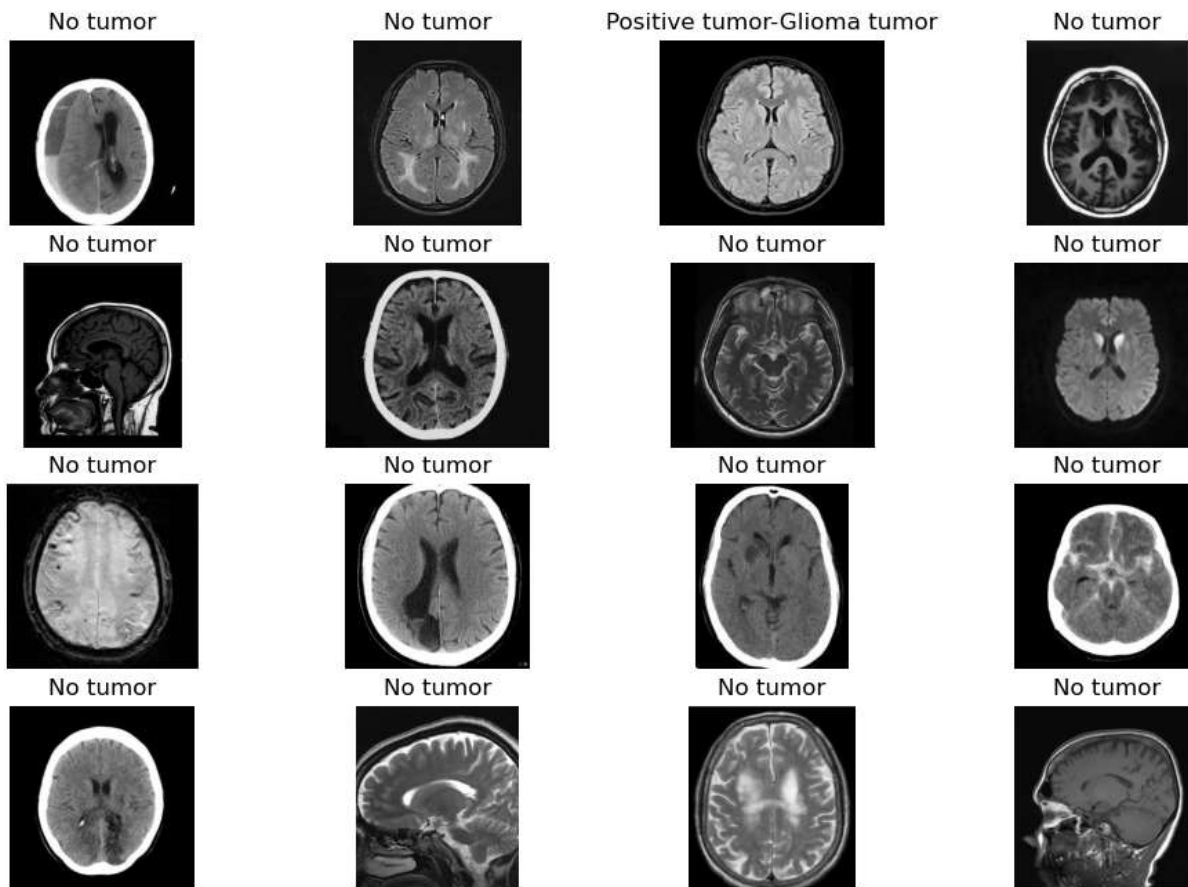
```
True
```

In [40]:

```
img = cv2.imread('C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Testing\\no_tumor'+i, 0
```

In [49]:

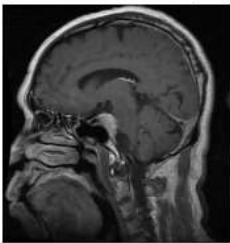
```
plt.figure(figsize=(12, 8))
testing_path = 'C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Testing\\no_tumor'
testing_files = os.listdir(testing_path)[:16]
c = 1
for i in testing_files:
    plt.subplot(4, 4, c)
    img_path = os.path.join(testing_path, i)
    img = cv2.imread(img_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img1 = cv2.resize(gray_img, (200, 200))
    img1 = img1.reshape(1, -1) / 255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(gray_img, cmap='gray')
    plt.axis('off')
    c += 1
plt.show()
```



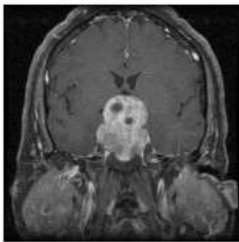
In [50]:

```
plt.figure(figsize=(12, 8))
testing_path = 'C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Testing\\pituitary_tumor'
testing_files = os.listdir(testing_path)[:9]
c = 1
for i in testing_files:
    plt.subplot(3, 3, c)
    img_path = os.path.join(testing_path, i)
    img = cv2.imread(img_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img1 = cv2.resize(gray_img, (200, 200))
    img1 = img1.reshape(1, -1) / 255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(gray_img, cmap='gray')
    plt.axis('off')
    c += 1
plt.show()
```

Positive tumor-Pituitary tumor



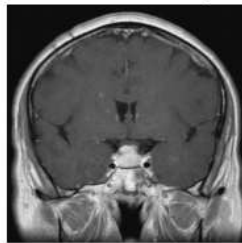
No tumor



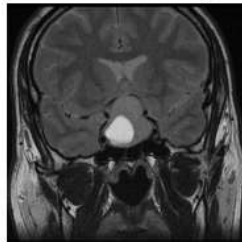
Positive tumor-Pituitary tumor



Positive tumor-Pituitary tumor



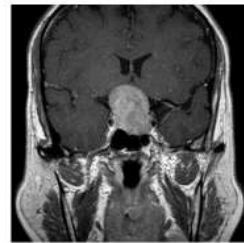
Positive tumor-Pituitary tumor



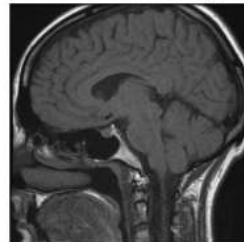
Positive tumor-Pituitary tumor



Positive tumor-Pituitary tumor



Positive tumor-Pituitary tumor



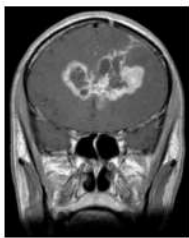
Positive tumor-Pituitary tumor



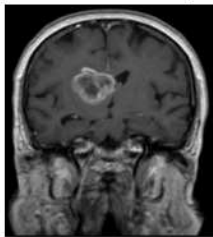
In [51]:

```
plt.figure(figsize=(12, 8))
testing_path = 'C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Testing\\glioma_tumor'
testing_files = os.listdir(testing_path)[:9]
c = 1
for i in testing_files:
    plt.subplot(3, 3, c)
    img_path = os.path.join(testing_path, i)
    img = cv2.imread(img_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img1 = cv2.resize(gray_img, (200, 200))
    img1 = img1.reshape(1, -1) / 255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(gray_img, cmap='gray')
    plt.axis('off')
    c += 1
plt.show()
```

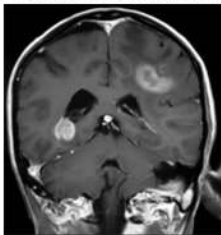
No tumor



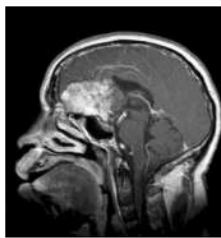
Positive tumor-Pituitary tumor



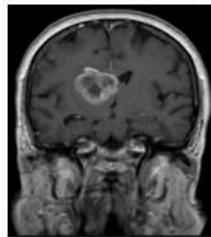
Positive tumor-meningioma tumor



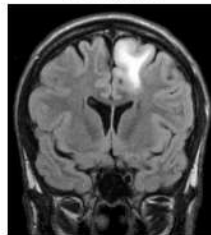
No tumor



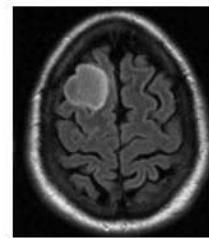
Positive tumor-Pituitary tumor



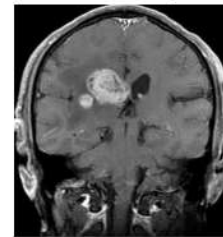
No tumor



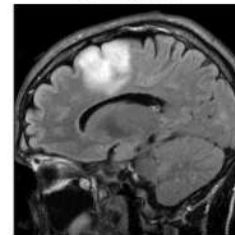
No tumor



Positive tumor-meningioma tumor



No tumor



In [52]:

```
plt.figure(figsize=(12, 8))
testing_path = 'C:\\Users\\racha\\OneDrive\\Desktop\\Datascience\\brain_tumor\\Testing\\meningioma_tumor'
testing_files = os.listdir(testing_path)[:9]
c = 1
for i in testing_files:
    plt.subplot(3, 3, c)
    img_path = os.path.join(testing_path, i)
    img = cv2.imread(img_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img1 = cv2.resize(gray_img, (200, 200))
    img1 = img1.reshape(1, -1) / 255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(gray_img, cmap='gray')
    plt.axis('off')
    c += 1
plt.show()
```

