Aim

The aim of this project is to develop a machine learning model that can accurately classify brain tumors from MRI images into two categories: no tumor and pituitary tumor.

Objective

1. To preprocess and analyze MRI images of the brain.
2. To implement and evaluate different machine learning algorithms for classification.
3. To compare the performance of these algorithms and identify the most accurate one.
4. To develop a user-friendly system for predicting brain tumors from new MRI images.

Summary

This project involves loading and preprocessing MRI images, splitting the data into training and testing sets, and training machine learning models for tumor classification. The dataset contains two classes of images: no tumor and pituitary tumor. Logistic Regression and Support Vector Classifier (SVC) were employed to classify the images, and their performances were evaluated based on accuracy scores.

Tools and Libraries Used

- **Google Colab (T4)**
- **Python**
- **Libraries**:
    - numpy
    - pandas
    - matplotlib
    - sklearn (scikit-learn)
    - cv2 (OpenCV)

Procedure

1. **Mount Google Drive and Import Libraries:**

CODE:

```
from google.colab import drive

drive.mount('/content/drive')

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
import os
import cv2
```

2. **Load and Preprocess Data:**
    - Load the dataset from Google Drive.
    - Resize images to 200x200 pixels.
    - Convert images to grayscale.
    - Assign labels to images based on their classes.

CODE:

```
path = os.listdir('/content/drive/MyDrive/brain_tumor/Training/')
classes = {'no_tumor':0, 'pituitary_tumor':1}
X = []
Y = []
for cls in classes:
    pth = '/content/drive/MyDrive/brain_tumor/Training/'+cls
    for j in os.listdir(pth):
        img = cv2.imread(pth+'/'+j, 0)
        img = cv2.resize(img, (200,200))
        X.append(img)
        Y.append(classes[cls])
X = np.array(X)
Y = np.array(Y)
```

3. **Reshape and Normalize Data:**
    - Reshape the image data for model input.
    - Normalize the pixel values.

CODE:

```
X_updated = X.reshape(len(X), -1)
xtrain, xtest, ytrain, ytest = train_test_split(X_updated, Y, random_state=10, test_size=.20)
```

```
xtrain = xtrain/255
xtest = xtest/255
```

4. **Train and Evaluate Models:**
    - Train Logistic Regression and Support Vector Classifier (SVC) models.
    - Evaluate the models on the test set.

CODE:

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
import warnings
warnings.filterwarnings('ignore')


lg = LogisticRegression(C=0.1)
lg.fit(xtrain, ytrain)
sv = SVC()
sv.fit(xtrain, ytrain)
```

5. **Evaluate Models:**
    - Print training and testing scores.

CODE:

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

6. **Predict and Visualize Results:**
    - Use the trained model to predict the class of new images.
    - Visualize the predictions.

CODE:

```
pred = sv.predict(xtest)
misclassified = np.where(ytest != pred)
print("Total Misclassified Samples: ", len(misclassified[0]))
```

```
dec = {0: 'No Tumor', 1: 'Positive Tumor'}
plt.figure(figsize=(12, 8))
c = 1
for i in os.listdir('/content/drive/MyDrive/brain_tumor/Testing/no_tumor/')[:9]:
    plt.subplot(3, 3, c)
    img = cv2.imread('/content/drive/MyDrive/brain_tumor/Testing/no_tumor/' + i, 0)
    img1 = cv2.resize(img, (200, 200))
    img1 = img1.reshape(1, -1) / 255
    p = sv.predict(img1)
    plt.title(dec[p[0]])
    plt.imshow(img, cmap='gray')
    plt.axis('off')
    c += 1
```

Highlights

- The use of Google Colab (T4) provides access to GPU resources for faster computation.
- Image preprocessing includes resizing, grayscale conversion, and normalization.
- The project employs two classifiers: Logistic Regression and SVC.
- Achieved high accuracy with both models, with Logistic Regression achieving a test score of 97.14% and SVC achieving a test score of 95.51%.
- Misclassified samples are identified and analyzed to improve model performance.

Conclusion

The project successfully demonstrates the application of machine learning techniques for brain tumor classification using MRI images. Both Logistic Regression and SVC models perform well, with high accuracy scores on the test set. This work can be extended to include more classes of tumors and further optimized with advanced techniques such as deep learning for even better performance.