

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY



Machine Learning Project on Cancer Detection

**Semester 04
ECAM 02**

Submitted to:
Ms Pratima Singh

Submitted by:
Aayush Kumar Singh
Naman Kumar
Prakash Choudhary

INDEX

S.No	Topic	Signature
1.	Introduction	
2.	Abstract	
3.	Methodology	
4.	Code and output	
5.	Conclusion	
6.	Reference	

Introduction

The human body is composed of numerous types of cells. Each cell has a specific function. These cells in the body grow and divide in an arranged manner and form some new cells. These new cells help to keep the human body healthy and ensures proper functioning. When some of the cells lose them ability to control their growth, they grow without any order. The extra cells formed form a mass of tissue which is called tumour. A brain tumour is a collection of abnormal cells in the brain.

Tumours can be benign or malignant. Malignant tumours lead to cancer while benign tumours are not cancerous.

The conventional method for tumour detection in magnetic resonance brain images is human inspection. The observation from human in predicting the tumour may mislead due to noise and distortions found in the images. This method is impractical for large amount of data and also very time consuming.

So, automated tumour detection methods are developed as it would save radiologist time. The MRI brain tumour detection is complex task due to complexity and variant of tumours. Tumour is identified in brain MRI using Machine Learning algorithms. The proposed work is divided into three sections: Pre-processing steps are applied on the brain MRI images, then texture features are extracted using Gray Level Co-occurrence Matrix (GLCM) and finally classification is performed using machine learning algorithm.

Abstract

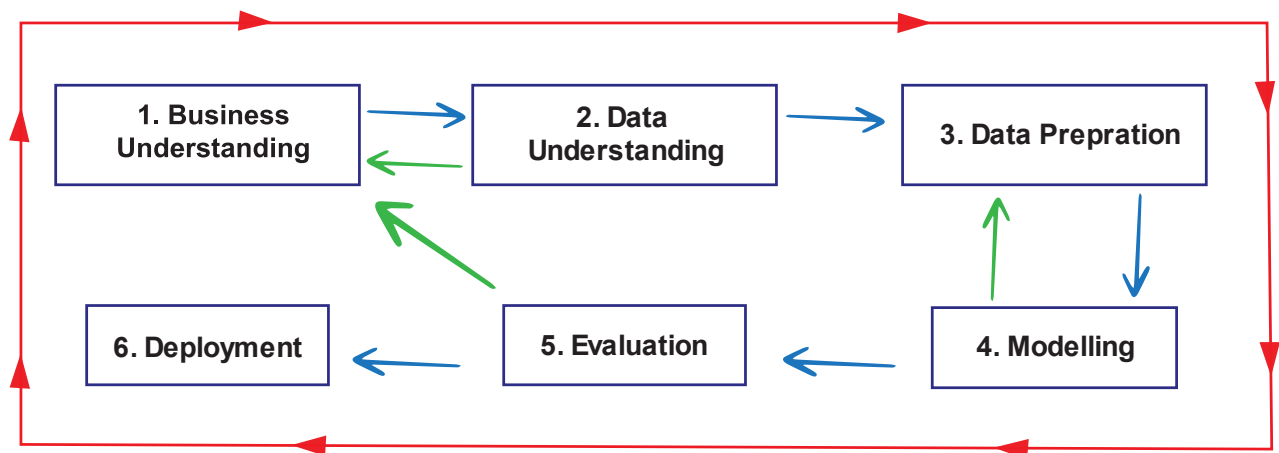
A Brain tumour is considered as one of the aggressive diseases, among children and adults. Brain tumours account for 85 to 90 percent of all primary Central Nervous System (CNS) tumours. Every year, around 11,700 people are diagnosed with a brain tumour. The 5-year survival rate for people with a cancerous brain or CNS tumour is approximately 34 percent for men and 36 percent for women. Brain Tumours are classified as: Benign Tumour, Malignant Tumour, Pituitary Tumour, etc. Proper treatment, planning, and accurate diagnostics should be implemented to improve the life expectancy of the patients.

The best technique to detect brain tumours is Magnetic Resonance Imaging (MRI). A huge amount of image data is generated through the scans. These images are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumours and their properties.

Application of automated classification techniques using Machine Learning (ML) and Artificial Intelligence (AI) has consistently shown higher accuracy than manual classification.

Hence, proposing a system performing detection and classification by using Deep Learning Algorithms using Convolution-Neural Network (CNN), Artificial Neural Network (ANN), and Transfer Learning (TL) would be helpful to doctors all around the world.

Methodology



- 1. Business Understanding:** Intention of our project is outlined from a business perspective which aims to successfully classify brain tumour patients using MRI scans.
- 2. Data Required:** We require brain MRI scans of patients diagnosed with pituitary tumour as well as some normal scans, we scraped our data from Kaggle (<https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>) under License CC0: Public Domain. And this data represents our problem.
- 3. Data Preparation:** Data we are using is much ready to use, for good measures we have resized the images, converted them to suitable dimension arrays, feature scaling and principal component analysis etc before fitting.
- 4. Modelling:** Here the data is expressed through apart models to give meaningful insights, new knowledge and patterns. Here we have used Logistic regression as well as support vector machine and then compared their accuracy with svm leading.
- 5. Evaluation:** We used. score function of svc and lr to obtain the score for each model.
- 6. Deployment:** This tested model is now used on new data outside the scope of dataset, we can now successfully run our model on

Code and Output

```
In [74]: # Importing Modules
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 [75]: # Collecting Data
import os

path = os.listdir(r'C:/Users/Aayush/OneDrive/Desktop/Brain Tumour DATA/Training/')
classes = {'no_tumor':0, 'pituitary_tumor':1}
```

```
In [76]: # Resizing data
import cv2
X = []
Y = []
for cls in classes:
    pth = r'C:/Users/Aayush/OneDrive/Desktop/Brain Tumour DATA/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])
```

```
In [77]: # Converting data in numeric form
X = np.array(X)
Y = np.array(Y)
```

```
In [78]: # Checking what are the sub arrays in our software. i.e. 0 - No Tumor , 1 - Tumor
np.unique(Y)
```

```
Out[78]: array([0, 1])
```

```
In [79]: # Checking total number of samples in data set which are having Tumor and No Tumor
pd.Series(Y).value_counts()
```

```
Out[79]:
```

1	827
0	395

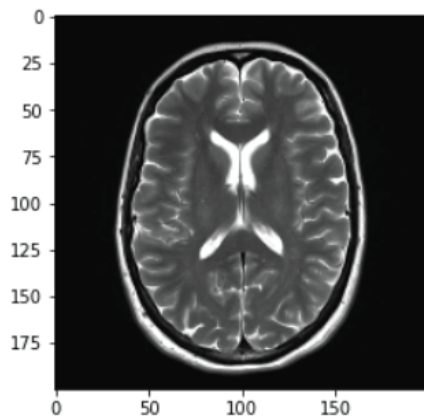
dtype: int64

```
In [80]: # Checking shape - this tells us Total number of samples and dimension of samples
X.shape
```

```
Out[80]: (1222, 200, 200)
```

```
In [81]: # VISUALING DATA
plt.imshow(X[5], cmap='gray')
```

Out[81]: <matplotlib.image.AxesImage at 0x22887bd55e0>



```
In [82]: # PREPARING DATA
# SK Learn works on 2 Dimensional data. But we had 3 Dimensional data above.
# So by reshaping it into 2 dimensional data we can continue
X_updated = X.reshape(len(X), -1)
X_updated.shape
```

Out[82]: (1222, 40000)

```
In [83]: # Using Train - Test split
xtrain, xtest, ytrain, ytest = train_test_split(X_updated, Y, random_state=10,
                                                test_size=.20)
```

```
In [84]: # 977 sampels for Training the Data
# 245 samples for Testing the Data
xtrain.shape, xtest.shape
```

Out[84]: ((977, 40000), (245, 40000))

```
In [89]: # Training Model
# high C means "Trust this training data a lot", while a low value says "This data may not be fully represent
import warnings
warnings.filterwarnings('ignore')
lg = LogisticRegression(C=0.1)
lg.fit(pca_train, ytrain)
```

Out[89]: LogisticRegression(C=0.1)

```
In [90]: sv = SVC()
sv.fit(pca_train, ytrain)
```

Out[90]: SVC()

```
In [91]: # Evaluation [Logistic Regression]
print("Training Score:", lg.score(pca_train, ytrain))
print("Testing Score:", lg.score(pca_test, ytest))
```

Training Score: 1.0
Testing Score: 0.9591836734693877

```
In [92]: # Evaluation [SVM]
print("Training Score:", sv.score(pca_train, ytrain))
print("Testing Score:", sv.score(pca_test, ytest))
```

Training Score: 0.9938587512794268
Testing Score: 0.963265306122449

```
In [93]: # Prediction
pred = sv.predict(pca_test)
np.where(ytest!=pred)
```

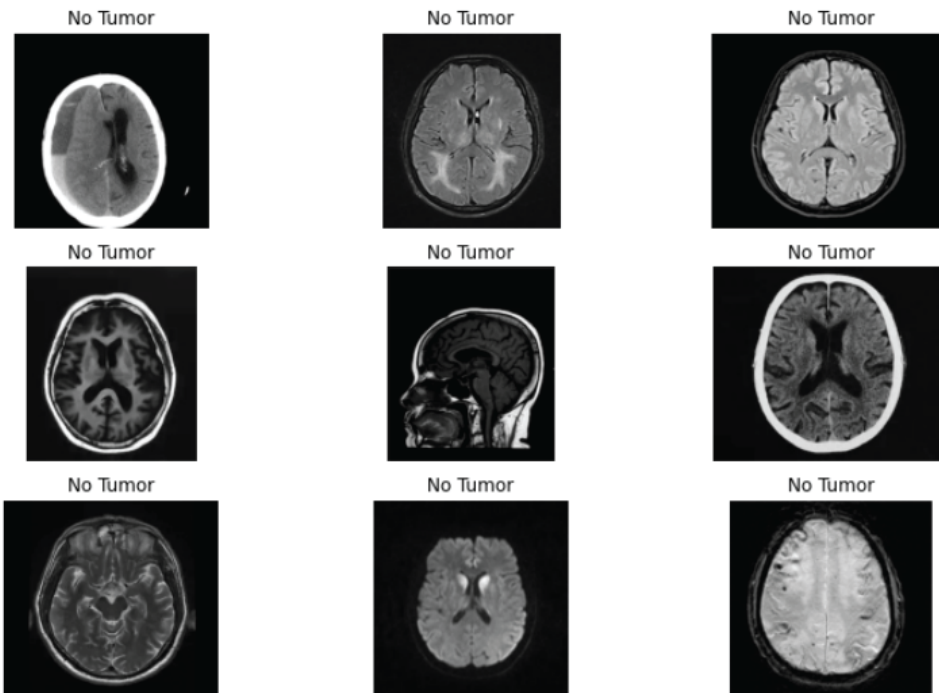
Out[93]: (array([36, 51, 68, 120, 212, 214, 220, 227, 239], dtype=int64),)

```
In [94]: # test Model
dec = {0:'No Tumor', 1:'Positive Tumor'}
```



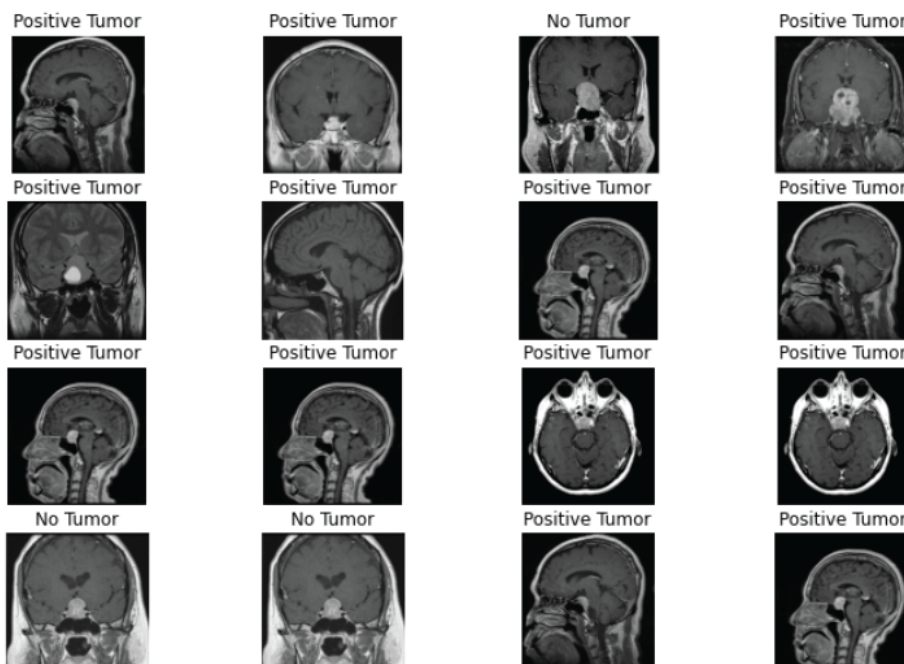
```
In [95]: plt.figure(figsize=(12,8))
p = os.listdir(r'C:\Users\Aayush\OneDrive\Desktop\Brain Tumour DATA\Testing')
c=1
for i in os.listdir(r'C:\Users\Aayush\OneDrive\Desktop\Brain Tumour DATA\Testing\no_tumor')[:9]:
    plt.subplot(3,3,c)

    img = cv2.imread(r'C:\Users\Aayush\OneDrive\Desktop\Brain Tumour DATA\Testing\no_tumor'+ '/' +i,0)
    if img is None:
        print('Wrong path:', path)
    else:
        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
```



```
In [96]: plt.figure(figsize=(12,8))
p = os.listdir(r'C:/Users/Aayush/OneDrive/Desktop/Brain Tumour DATA/Testing/')
c=1
for i in os.listdir(r"C:\Users\Aayush\OneDrive\Desktop\Brain Tumour DATA\Testing\pituitary_tumor")[:16]:
    plt.subplot(4,4,c)

    img = cv2.imread(r"C:\Users\Aayush\OneDrive\Desktop\Brain Tumour DATA\Testing\pituitary_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
```



Conclusion

Q. Why did we use SVM over Logistic Regression?

We have used SVM (Support Vector Machines) over Logistic Regression because the training score in logistic regression was 1.0 and the testing score was 0.9591 whereas, the training score in SVM algorithm was 0.9938 and the testing score was 0.9632. Hence, Support Vector Machine (SVM) algorithm was more suited for our project as over fitting would have occurred if we used Linear Regression.

Q. What were the differences in accuracy in SVM and Linear Regression?

SVM (Support Vector Machines):

Training Score: 0.99385

Test Score: 0.963265

Difference: 0.030585

Linear Regression:

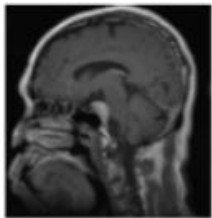
Training Score: 1.0

Test Score: 0.963265

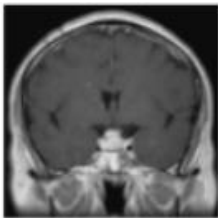
Difference: 0.036735

Result

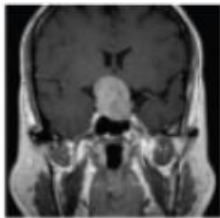
Brain Tumours very successfully detected:



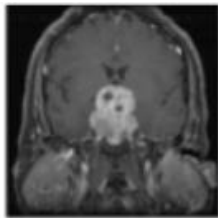
Positive Tumour



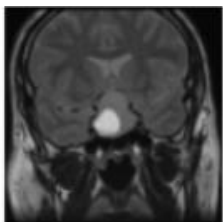
Positive Tumour



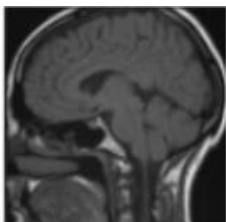
Negative Tumour



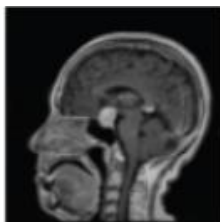
Positive Tumour



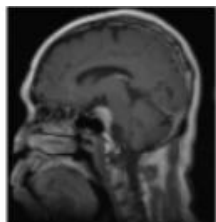
Positive Tumour



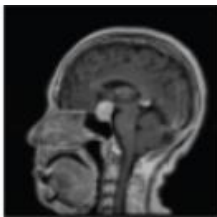
Positive Tumour



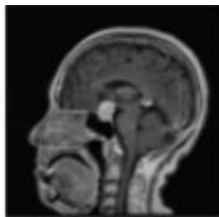
Positive Tumour



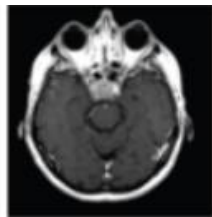
Positive Tumour



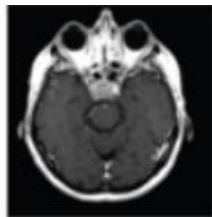
Positive Tumour



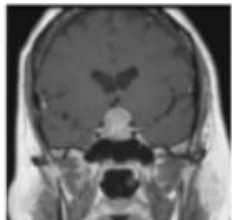
Positive Tumour



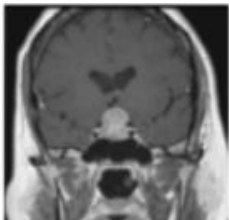
Positive Tumour



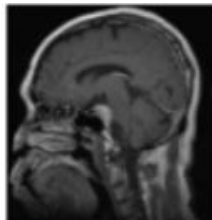
Positive Tumour



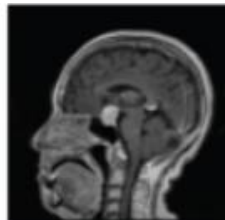
Negative Tumour



Negative Tumour



Positive Tumour



Positive Tumour

Reference

Youtube: <https://www.youtube.com/watch?v=5lgrlddp-98>

Kaggle:

https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqBcXFBcFVndU1nM05QUEJ5UHVTMmxjWEZuUHF5QXxBQ3Jtc0trUXdiTUhOS1JjbORVNmpPdlpnLTh1bTI2ZnVlaFB1LXJac1huYkUyR19YMEJvUVFNLXNWbC1sUWRSTTBYYQ1RRTnY4MIVGRWR2V1V2VkU3N01QZjI5UzRPd0JyaVR4emZoMGdQOXUyaURMVWtOSjhCWQ&q=https%3A%2F%2Fwww.kaggle.com%2Fsartajbhuvaji%2Fbrain-tumor-classification-mri&v=5lgrlddp-98

Geeks for Geeks: <https://www.geeksforgeeks.org/tumor-detection-using-classification-machine-learning-and-python/?ref=gcse>

THANK YOU!!