

Brain Tumor Segmentation and Classification

Likhitha S. (IMT2014030)
Nikunj Gupta (IMT2014037)
Tummala Armitha (IMT2014060)

December 10, 2017

1 Introduction

Brain tumor segmentation seeks to separate healthy tissue from tumorous regions. This is an essential step in diagnosis and treatment planning in order to maximize the likelihood of successful treatment. Due to the slow and tedious nature of manual segmentation, computer algorithms that do it faster and accurately are required. Because of the unpredictable appearance and shape of a brain, segmenting brain tumors from imaging data is one of the most challenging tasks in medical image analysis.

2 Goal Statement

Segmentation of the brain tumour and classify it as either benign or malignant tumor

3 Background

Brain is a complex organ since it contains more than 10 billion working brain cells. Primary brain tumors can be either malignant (contain cancer cells) or benign (do not contain cancer cells). A primary brain tumor is a tumor which begins in the brain tissue. If a cancerous tumor starts elsewhere in the body, it can spread cancer cells, which grow in the brain. These type of tumors are called secondary or metastatic brain tumors. The malignant tumor tends to grow and spread in a rapid and uncontrolled way that can cause death and the Tumor are graded according to how aggressive. They are as

- LGG: Low Grade Tumor (Benign stage)
- HGG: High Grade Tumor (Malignant stage)

4 Related work

A research paper by Dr.A.R. Kavitha [1], talks about using Genetic Algorithm to segment the MRI brain tumor images. Pre-processing was done using Wiener Filter (a 2D adaptive noise removal filter and it uses pixel-wise adaptive wiener method). GLCM features are extracted from segmented images and given to the SVM Classifier which gets trained and ready for classifying test images.

Another research paper by Alan Jose [2] did Brain Tumor Segmentation Using K-Means Clustering And Fuzzy C-Means Algorithms And Its Area Calculation. After the segmentation, (which is done through k-means clustering and fuzzy c-means algorithms) the brain tumor is detected and its exact location is identified.

5 Data

we have collected our data from the BRATS 2015 Challenge [3].

5.1 Data set Folder Structure:

- Training
 - HGG - This folder contains brain images of 220 Patients. There is a different folder for each patient. There are 5 different MRI images for each Patient. The 5 different images are T1,T2,T1C,FLAIR and OT(Ground truth of tumor Segmentation) . All these image files are stored in .mha format.
 - LGG - This folder contains brain images of 54 Patients. There is a different folder for each patient. There are 5 different MRI images for each Patient. The 5 different images are T1,T2,T1C,FLAIR and OT(Ground truth of tumor Segmentation) . All these image files are stored in .mha format.
- Testing
 - HGG.LGG - This folder contains brain images of 110 Patients. There is a different folder for each patient. There are 4 different MRI images for each Patient. The 4 different images are T1,T2,T1C and FLAIR. All these image files are stored in .mha format.

After gaining insights in domain and lot of research, we have decided to use only FLAIR images for our work.

6 Summary of Implementation

We start with taking each image from the training set and segment the tumor from the image using watershed clustering (region-growing) technique. Then, we extract GLCM features on the segmented image. We repeat the above process for every image in the data set and use them to train our model - random forest Classifier. Then, for every testing image, we segment the tumor using the same method (Watershed) and then classify the tumor as benign or malignant using our model.

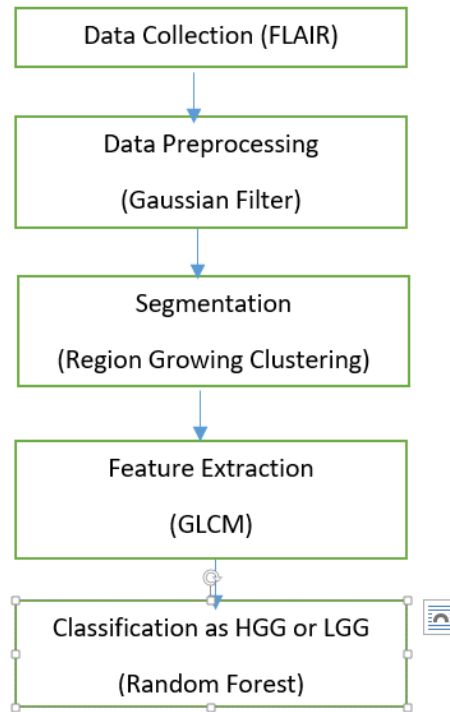


Figure 1: Flow Chart

7 Implementation

The whole project is basically divided into for major sections, namely,

- Data Processing

- Segmentation
- Feature Extraction
- Classification

7.1 Data Processing

Gaussian filtering is highly effective in removing Gaussian noise from the image. The images we got from BRaTS were very noisy and had to be de-noised to be used for segmentation.

Gaussian Blur [4]: In this approach, a Gaussian kernel is used. It is done with the function, `cv2.GaussianBlur()`. We should specify the width and height of the kernel which should be positive and odd. We also should specify the standard deviation in the X and Y directions, `sigmaX` and `sigmaY` respectively. If only `sigmaX` is specified, `sigmaY` is taken as equal to `sigmaX`. If both are given as zeros, they are calculated from the kernel size. If you want, you can create a Gaussian kernel with the function,

`cv2.getGaussianKernel()`.

7.2 Segmentation

The data set has brain images with tumor which we segmented to provide it to the feature extraction module. After segmentation we obtained two major regions in the image- tumor and non-tumor. The segmentation was done using Watershed Algorithm.

Watershed Algorithm [5]: OpenCV implemented a marker-based watershed algorithm which is an interactive image segmentation method. It gives different labels for the object we know. Label the region which we are sure of being the foreground or object with one color (or intensity), label the region which we are sure of being background or non-object with another color and finally the region which we are not sure of anything, label it with 0. That is our marker. Then apply watershed algorithm. Then our marker will be updated with the labels we gave, and the boundaries of objects will have a value of -1. This is basically a **region-growing approach**.

7.3 Feature Extraction

Now we need to extract some features from images as we need to do a binary classification of them using a classifier which needs these features to get trained on. We chose to extract GLCM (texture-based features).

GLCM Features [6]: After obtaining the segmented images from the above subsection, we extract GLCM features and store them. GLCM stands for

Gray-Level Co-occurrence Matrix. Texture Analysis Using the Gray-Level Co-Occurrence Matrix (GLCM) is a statistical method of examining texture that considers the spatial relationship of pixels.

7.4 Classification

We used Random Forest for classification. The classifier was trained on the GLCM features obtained from the segmented images and used to classify benign (LGG) and malignant (HGG).

Random Forests [7]: When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance.

After each tree is built, all of the data are run down the tree, and proximities are computed for each pair of cases. If two cases occupy the same terminal node, their proximity is increased by one. At the end of the run, the proximities are normalized by dividing by the number of trees. Proximities are used in replacing missing data, locating outliers, and producing illuminating low-dimensional views of the data.

8 Tools

We have implemented our project in Python

- Read .mha images: SimpleITK
- Gaussian Blur: OpenCV
- Watershed algorithm: SciPy, scikit-image, and OpenCV
- GLCM : Package mahotas
- Random forest: RandomForestClassifier from sklearn.ensemble

9 Various other approaches explored

We have attempted several approaches for the classification of tumor in brain images.

- Data Preprocessing: Removing noise and enhancing the image.
 - Median filter
 - **Gaussian blur**

The results of segmentation were not good on images that are filtered using Median Filter. The outline of brain is also segmented as tumor on images which are filtered using Median Filter. So, we have decided to use Gaussian Filter which gave good results for segmentation (The outline of brain is not segmented as tumor).

- Segmentation: Identifying the tumored region in the brain image. We have tried using the following techniques:
 - Binary thresholding
 - **Watershed algorithm**

Binary thresholding can be used to convert a gray scale image to binary image based on the selected threshold values. The problems associated with such approach are that binary image results in loss of texture and the threshold value is coming out to be different for different images. Hence, we are looking for a more advanced segmentation algorithm, the watershed algorithm.

- Classification: Classify whether the tumored part belongs to malignant or benign
 - Support Vector Machine
 - **Random forest**

The accuracy after training the model using SVM is only 40 percent. So, we have decided to use Random Forest, which gave better accuracy.

10 Results

Since, we do not have ground truth labels in our testing dataset, we have decided to use only training dataset for both training and testing. As mentioned earlier, there are only 54 images in LGG and there are 220 images in HGG, we have decided to consider all 54 images in LGG and only 54 out of 220 images from the HGG. Now, we have 54 LGG images and 54 HGG images. We have trained our model using 50 HGG and 50 LGG images. The remaining 4 HGG and 4 LGG are used for testing. The results are shown in below figure.

```

[STATUS] Creating the classifier..
[STATUS] Fitting data/label to model..
../data/test/HGG1.jpg      HGG
../data/test/LGG2.jpg      HGG
../data/test/LGG4.jpg      LGG
../data/test/LGG1.jpg      LGG
../data/test/HGG3.jpg      HGG
../data/test/HGG4.jpg      HGG
../data/test/HGG2.jpg      LGG
../data/test/LGG3.jpg      LGG

```

Figure 2: Results

Form the above figure , we can see that 6 testing images are classified correct and other 2 images are classified incorrectly .
For evaluating our model,we have calculated measures like Accuracy, Precision, recall and F1-Score.The results for 6 different training datasets and corresponding testing datasets can be found below.

A	B	C	D	E
Iterations	precision	f1_score	recall	accuracy
1	1,0.8	0.857,0.889	0.75,1	0.875
2	0.667,0.6	0.571,0.667	0.5,0.75	0.625
3	1,0.571	0.4,0.727	0.25,1	0.625
4	1,0.667	0.667,0.8	0.5,1	0.75
5	0.75,0.75	0.75,0.75	0.75,0.75	0.75
6	0.6,0.667	0.667,0.571	0.75,0.5	0.625
Results	0.836 ,0.675	0.652,0.734	0.59,0.86	0.71

Figure 3: Evaluation Measures

The accuracy of our model is nearly 71 percent.

11 Conclusions

Since, we are automatically segmenting the tumor and classifying it as Benign or Malignant, it significantly reduces the amount of time that a doctor takes to identify and classify the tumor. It also to some extent reduces the error made while segmenting and classifying tumor.

12 Constraints

As explained in earlier sections, due to imbalance in number of images in LGG and HGG classes, we have trained our model with only 50 LGG and 50 HGG images. Due to some Constraints, we have used only classical ML techniques like SVM, Random Forest etc instead of Deep Learning Techniques like CNN, which could have given more accuracy.

13 Future work

- Improving the accuracy of our model by training the model with more number of images.
- Usage of Deep Learning techniques like Convolution Neural Networks to make our model better.
- Identifying or labelling the tumor sub-regions i.e specifying which sub-region is edema, non-enhancing solid core, necrotic/cystic core and enhancing core.

14 References

- [1] Link for Data Set
- [2] "Brain Tumor Segmentation using Genetic Algorithm with SVM Classifier" by Dr.A.R. Kavitha, L.Chitra, R.kanaga
- [3] "Brain Tumor Segmentation Using K-Means Clustering And Fuzzy C-Means Algorithms And Its Area Calculation" by Alan Jose, S.Ravi, M.Sambath.
- [4] https://docs.opencv.org/3.1.0/d3/db4/tutorial_py_watershed.html
- [5] <https://www.smir.ch/>
- [6] <https://in.mathworks.com/help/images/texture-analysis-using-the-gray-level-co-occurrence-matrix-glc.html>
- [7] https://www.stat.berkeley.edu/breiman/RandomForests/cc_home.htm#overview
- [8] <https://pyscience.wordpress.com/2014/10/19/image-segmentation-with-python-and-simpleitk/>