

PROJECT DRAFT

TITLE: BRAIN TUMOR SEGMENTATION FROM MRI IMAGES^[2]

AUTHOR: SANDHYA MURALI (*sm2290@g.rit.edu*)
(*Affiliated to Rochester Institute of Technology*)

1. ABSTRACT:^[1]

Medical Science in Image Processing is an emerging field which has proposed a lot of advanced techniques in detection and analysis of a particular disease. Treatment of brain tumors in recent years is getting more and more challenging due to complex structure, shape and texture of the tumor. Therefore, by advancing in image processing, various methodologies have been proposed to identify the tumors in the brain. The advancement in this field created an urge in me to research more on the techniques and methodologies developed for tumor extraction. Hence, I propose a system to extract Tumor from the brain using MRI images. This technique involves different image processing methodologies such as noise removal, filtering, segmentation and morphological operations. Extraction of Brain tumor can be accomplished successfully by performing these operations on MATLAB software.

Keywords: MRI, noise removal, filtering, segmentation, morphological operations, MATLAB.

2. INTRODUCTION:

This project proposes different methodologies to segment a tumor from an MRI image and determine correlation for all of the methodologies except one. For this several segmentation techniques have been implemented and an analysis is provided regarding the efficiency of the segmentation technique used. Each MRI image is passed through an imaging chain where the image is preprocessed to remove noise and is further enhanced to improve the contrast of the image. This paper proposes five different segmentation techniques which are then applied on the image to extract the tumor. These segmentation techniques include local thresholding, Otsu's segmentation, k-means, Watershed segmentation and MSER. Applying each of the segmentation techniques allows us to determine the most appropriate method to segment the tumor from each of the images. Normalized cross correlation on each of the segmented tumors is then applied to determine the accuracy. Also, the foreground (tumor) regions and the background (skull) regions are taken to compute cross correlation between target variable and tumor region of the texture image in order to determine how pixels of the tumor region are closely correlated with each other. The target variable is a vector which consists of two classes: 0 for skull region and 1 for tumor region. The tumor region represents the pixel values for the foreground points extracted using the `ginput()` command from a texture image. The texture image is generated by applying the `rangefilt()` method. In order to enhance the texture characteristics of the image, smoothing filter is applied to the texture image. This helps me to understand texture features more clearly.

In this project, the major challenge faced by me was to locate and extract the proper tumor region from the image. Due to several lighting issues, unnecessary white portions were present in the image which could wrongly be segmented as a tumor. Also the unwanted noise and reduced contrast displays several regions from the image that are falsely claimed as a tumor. Another challenge faced was degraded quality of the MRI image due to several problems that would have occurred during the acquisition stage. Also, extracting the texture features from the image using appropriate texture filters in order to compute correlation coefficient was another major challenge faced.

The problems of exposure have been minimized greatly by applying appropriate noise filtering techniques like median filter. This noise removal methodology removes most of the unwanted white and black regions from the image. Applying appropriate segmentation techniques followed by different morphological operations helps in retaining only the relevant tumor portions and suppresses the skull portions of the image to a large extent. This will help in analyzing meaningful portions of the image very efficiently. The problem of extracting texture features using appropriate texture filter was solved using the `rangefilt()` method followed by applying a smoothing filter to enhance the texture. In this way, enhancing the texture of the foreground and background region helped me to compute the correlation coefficient of the texture image accurately.

Tumors can be located at any region of the body. That is, tumors can be located in the brain, breast, lungs and so on. Each of these tumors can be segmented based on the region present in the body. For example, tumors in the breast are formed due to calcium deposits in the breast tissue. These are referred to as breast calcifications. Breast calcifications can be of two types: Macrocalcifications and Microcalcifications. Macrocalcifications are large white dots that are dispersed over the breast tissue. They are often non-cancerous. Microcalcifications are small calcium deposits that appear in certain patterns in the breast tissue. They are cancerous. From all the regions, I have proposed to segment tumor from the brain using 5 proposed segmentation techniques and have also established normalized cross-correlation to compute accuracy of the tumor region extracted.^[14] Also cross correlation is calculated on the texture image to establish how the pixels in the tumor region are closely correlated.^[15]

There are various medical imaging technologies to detect presence of tumor in the brain. Some of them include CT (Computed Tomography), MRI (Magnetic Resonance Imaging), PET (Positron Emission Tomography), Ultrasound and X-ray. From the imaging technologies described, MRI (Magnetic Resonance Imaging) is being preferred in this paper as it gives detail anatomy of different parts of the brain by providing a higher contrast of the tissues. Also MRI can be performed on any individual ranging from adults to small children. Therefore, a large dataset is available to test the proposed methodology.^[5]

3. PREVIOUS WORK:

Many researchers proposed different methodologies to detect tumor in the brain from MRI images.

Vipin Y. Borole, Sunil S. Nimbhore and Dr. Seema S. Kawthekar et al, 2015, in their paper 'Image Processing Techniques for Brain Tumor Detection: A Review', they

compare about different techniques in segmentation, edge detection and noise removal techniques. The different segmentation techniques that are compared include active contour method, Watershed method, Seed region growing and Marker based watershed. The noise removal techniques that they have compared are Mean Filter, Median filter, Wiener Filter and Hybrid filter while the different edge detection techniques compared are Log, Sobel, Canny, Prewitt and Zero Cross operator. The methodology for tumor extraction starts with preprocessing techniques like applying median filter to remove the noise and applying image enhancement techniques to improve the contrast of the image. Then morphological operations like dilation and erosion are applied to sharpen the relevant regions after which thresholding which is an image segmentation technique is applied to extract the region of interest. However, the drawback of this approach is that the threshold values for performing thresholding will depend on the quality of the image. Hence different images can have different thresholding values. Therefore, the percentage of accuracy using a standard threshold value is not specified to determine the efficiency of the methodology.^[6]

Rohini Paul Joseph, C. Senthil Singh and M.Manikandan et al, in their paper ‘Brain tumor MRI Image Segmentation and Detection in Image Processing’, they mainly talk about using k-means clustering technique to detect the tumor present in the brain after applying different preprocessing steps to enhance the quality of the image. They have implemented the entire algorithm of k-means and used a fixed value of k as 4. As k-means clustering technique sometimes results in unnecessary clusters; they have further performed morphological operations to enhance their results. Though their results seem to be accurate, they did not state whether their system will demonstrate accurate results for multiple tumors. Also due to unsupervised classification, it is possible that better results may be obtained for different values of k. Again the value of k depends on the quality of the image. Hence they have not demonstrated the percentage of accuracy if value of k changes.^[7]

Swati Ghare, Nikita Gaikwad, Neha Kulkarni, Monica Nerkar et al 2015, in their paper ‘Detection of Possibility of Brain Tumor Using Image Segmentation’, they talk about applying Fuzzy C mean segmentation to extract the region of interest from the brain. Initially the MRI image is converted to gray scale and noise removal techniques are applied to remove the noise from the image. Thresholding is then applied in order to convert the gray scale to a binary image. Sobel Edge detector is then used to detect the edges after which Fuzzy C mean segmentation technique is used to extract the tumor from the image. The entire algorithm of Fuzzy C-means is implemented where C is the number of clusters that will be formed. Though this methodology gives accurate prediction of tumor cells which are not predicted by k-means; this technique is comparatively slower than the k-means clustering algorithm.^[8]

4. PROPOSED METHODOLOGY:^[1]

The proposed methodology initially involves applying different preprocessing techniques the image to sharpen the relevant portions of the image. Different segmentation techniques and morphological operations are then applied to extract the tumor regions.

The steps are as follows:

- a. Read the MRI image of the brain as input.
- b. Convert the image to gray scale.

- c. Apply median filter to remove noise from the image.
- d. Apply Otsu's Segmentation, Local Thresholding, K-means clustering, Watershed Segmentation and MSER.
- e. Perform Morphological operations.
- f. Display the segmented tumor for each of the methods.
- g. Compute normalized cross-correlation.
- h. Compute cross-correlation for the pixel values of the tumor region of the texture image.

I. Gray Scale Image:

MRI (Magnetic Resonance Imaging) creates a magnetic field to detect tumors that may be present in the organs and tissues. ^[11] These MRI images are given as input to our system for processing. Generally, MRI images look like black and white images on the computer. However, this is a misnomer. Though these images show only black and white pixels, technically the white portions of the image have large amounts of red, green and blue pixels depending on the color channel. Hence it becomes difficult to preprocess the image as preprocessing is done on a monochrome image. Due to the above mentioned reasons, we first convert our given image to grayscale where the only possible shades are pure white or pure black. In grayscale imaging, the darkest possible shade is black whereas the lightest possible shade is white. Remaining shades represent different shades of gray.

II. Noise Removal:

In image processing, it is often desirable to have some amount of noise in the image. Therefore, to get accurate results it is advisable to apply a reasonable smoothing filter to smear the image in order to get rid of the noise. Hence we use the median filter to blur the image and get efficient results. Applying this filter will make it easy to recognize only relevant portions of the image by suppressing the unwanted edges and keeping only the important edges. The median filter is a nonlinear filter which helps in removing the salt and pepper noise present in the image. Since the filter is nonlinear, performance is slower as compared to other noise removal filters like Gaussian. Therefore, greater the size of the filter, slower the operation. In our project we have used a filter of size 2X2. The reason of using a median filter is to remove the unwanted white portions of the image which could be misguided as a tumor. This will help us in segmenting meaningful portions of the image very efficiently.

III. Segmentation Techniques: ^[3]

Segmentation is a process of segmenting the image into multiple sections in order to view the region of interest clearly and efficiently. In our project, we have proposed five different segmentation techniques to extract the tumor from the image. These include Otsu's Thresholding, Local Thresholding, K-means clustering, Watershed Segmentation and MSER. Each of these methods are briefly explained as follows:

a) Otsu's Thresholding: ^[12]

Otsu's thresholding is a clustering based segmentation technique which converts the grayscale image to a binary image by assuming only two classes of pixels (foreground and background). In the project, `imtophat()` morphological operation is performed which computes morphological opening and subtracts the result from the original image. This is used to enhance the contrast in the image. Further `graythresh()` function is applied to compute a global threshold value which will help in converting a grayscale image to a binary image using the `imbinarize()` method. By selecting appropriate `graythresh` levels, we can successfully extract the tumor from the image. However, this method sometimes misses out relevant portions of the image or sometimes considers unwanted portions of the image. Therefore, there may be situations where only a portion of tumor may be extracted or certain skull regions along with the tumor may be displayed. The regions that are displayed are completely dependent on the global threshold values that are computed by the `graythresh()` function.

b) Local Thresholding:

Local Thresholding technique is the simplest segmentation method. This technique converts a gray scale into a binary image based on the appropriate threshold value specified by the user. In the project, the threshold value selected is $(t_0 + (\text{maximum pixel value in the image}) + (\text{minimum pixel value in the image})) / 2$ where $t_0=60$. The pixel values above this threshold value are set to 1 (white) whereas the pixel values below this threshold value are set to 0 (black). The threshold value mentioned is selected at random and can differ from image to image. Therefore, there may be situations where expected region of interest may not be displayed or unwanted regions may be displayed along with the region of interest in the output. Efficient results can be obtained by selecting appropriate threshold value.

c) K-means Clustering: ^[10]

K-means clustering technique is another segmentation technique that is proposed to extract the tumor from the brain image. K-means is an unsupervised segmentation technique where the initial set of clusters (K) along with the cluster centers needs to be initialized before segmenting using this algorithm. Therefore, segmenting the tumor using k-means clustering technique entirely depends on the value of K and the mean centers initialized. In the project, the value of K is considered as 4. That means, we will segment our image into 4 different regions each represented by a different color. To perform k-means clustering, we first convert our image to linear space using the `reshape()` command. This resultant image is passed as an input to k-means which generates the set of clusters and cluster centroids. The image in linear space is re-converted back to spatial domain and the individual cluster results are sub-plotted as a result. Based on the selected cluster, the tumor is extracted and displayed as one of the cluster results for most of the images.

d) Watershed Segmentation: ^[13]

Watershed Segmentation is another segmentation technique used to extract the region of interest which in this case is a tumor from the smeared MRI image. This segmentation technique is particularly useful to segment objects when they are touching

each other. This algorithm helps in finding the catchment basins and ridge lines in the image. In this case, the ridge line represents the height that separates the two catchment basins. For this, `bwdist()` method is used to compute the distance from every pixel to every nonzero pixel. Watershed algorithm is then implemented using `watershed()` function. This returns a labeled matrix which consists of positive integer values for different regions and 0 for ridge lines. This image is not very useful as there is only one catch basin spanning the entire image. Here catchment basins are the regions we want to identify. Therefore we take the compliment of our image and apply `bwdist()` on the complimented image after which we negate the distance in order to determine the bright catchment basins that represents individual regions. We then apply `watershed()` method which will return a labeled matrix consisting of positive values along with catchment basins. To display the image, we convert the labeled matrix into an rgb image and display the result. However before taking the compliment of the image to apply watershed algorithm, several pre-processing techniques are applied to enhance the region of interest we want to extract. In the project, Otsu's thresholding followed by morphological erosion of disk size 3 has been applied to enhance the contrast of the tumor region. This technique gives accurate results if the image is of reasonable contrast so that the watershed algorithm can extract the catchment basins efficiently.

e) MSER:^[9]

MSER (Maximally Stable Extremal Regions) is a blob detection segmentation technique that helps in extracting the tumor from the image. This segmentation technique is less sensitive to light changes and detects small regions which makes it useful to extract sensitive region of interest from an image. In the code, we use the function `detectMSERfeatures()` to capture the features of the region we want to extract. We then use these feature to find that region and plot the ellipse. However the `detectMSERfeatures()` has a certain area range and threshold value passed in order to collect features only within the specified range. In the project, the threshold value passed is 12 and the area range is 200 to 5000. For threshold values lesser than 12, more regions were detected whereas for threshold values greater than 12, expected regions sometimes do not appear. By using the threshold value 12, there are situations where more than one tumor in the image is not extracted. This may be because, either the area range specified is too large to identify the important regions or the threshold value is slightly small which prevents it from identifying the important regions. Therefore, in order to provide accurate results, appropriate threshold value and area range must be specified.

IV. Morphological Operations:

This is the last step where appropriate morphological operations are applied to remove the unnecessary segments formed after segmentation or enhance the region of interest after segmentation. The morphological operations are opening, closing, erosion and dilation. Morphological dilation is used to enhance the region of interest whereas morphological erosion is used to remove the unwanted clusters formed as a result of segmentation. For example, in the project, after Otsu's segmentation, erosion is performed with a structuring element of disk size 3 to remove unwanted regions whereas after Local thresholding opening operation is performed which is erosion followed by dilation.

V. Normalized Cross-Correlation: ^[15]

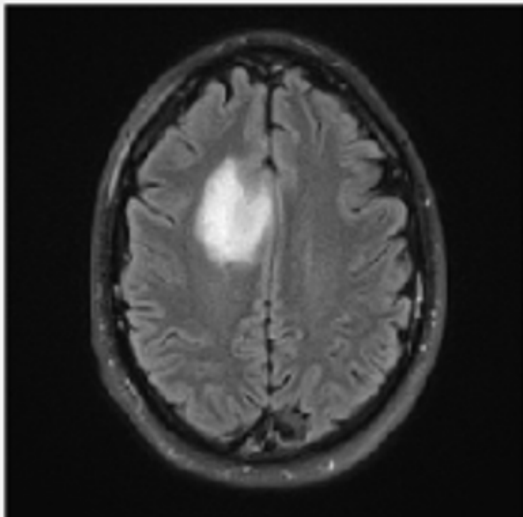
After extracting the tumor from the image by performing appropriate segmentation techniques and morphological operations, correlation is performed to determine if the tumor extracted can be accurately determined as a tumor. For this normalized cross-correlation has been performed using the function `normxcorr2()`. In this process, we compare the tumor extracted with the MRI image. If it results in a high positive value (that is value close to 1), we can say that the given tumor is highly correlated with the image. That is tumor is extracted efficiently with less noise. Otherwise, the tumor extracted has some noise in that image due to which correlation value reduces. This normalized cross-correlation technique has been applied on every segmentation technique except on MSER. The reason is because in MSER, we determine MSER features and plot the regions based on the features extracted using the `plot()` function. Due to this, the region is not actually superimposed on the original image. The region is just highlighted to display the result. Hence it was not possible to calculate normalized cross-correlation for the tumor extracted using MSER technique. I also tried to superimpose the region using the `drawshape()` method. But I was not able to use the features generated from the `detectMSERfeatures()` method. Therefore, using normalized cross correlation, I have determined if the tumor extracted has some noise present in it or not for all of the segmentation technique except MSER.

VI. Cross Correlation: ^[16]

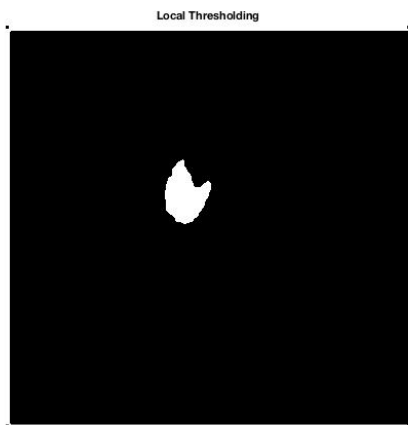
In this technique, image is converted to double and appropriate texture filter method is applied to determine the texture features of the image. In the program, the best texture filter I found was `rangefilt()`. To enhance the textures of the tumor region (foreground) and skull region (background), smoothing filter is applied. In the program, user defined filter matrix is provided to smoothen the image. Greater the size of the filter, better the features can be seen. The foreground (tumor) regions and background (skull) regions are extracted using the `ginput()` command. A target variable vector is initialized which consists of two classes: 0 for skull region and 1 for tumor region. Now cross correlation is applied to compute the correlation between target variable and tumor region in order to determine the degree to which the pixels of the tumor region are closely correlated with each other. This cross correlation is computed using the matlab command `xcorr2()`. The tumor region represents the pixel values for the foreground points extracted using the `ginput()` command from the texture image. In this way, I have determined how the pixels in the tumor region are closely related to each other. However again this value of cross correlation completely depends on the way we select the foreground and background points using the `ginput()` method. If we do not select the points properly; that is for example if the points of the tumor region (foreground) are considered at the edges or borders, there is a possibility that we may not get an accurate cross-correlated value.

5. RESULTS: ^[4]

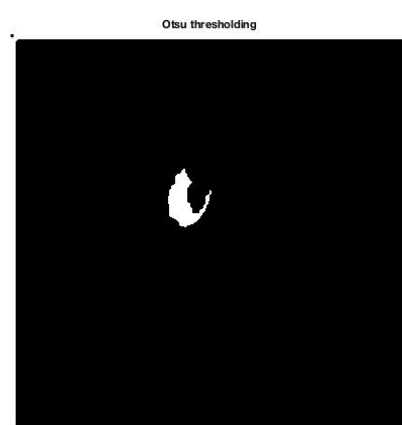
Input1:



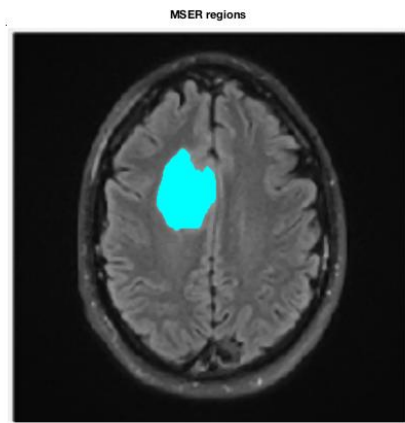
Local Thresholding



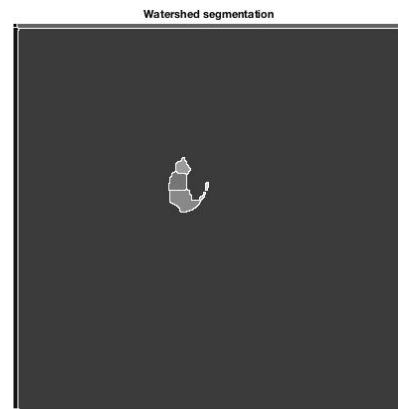
Otsu's Thresholding



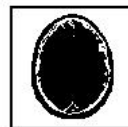
MSER



Watershed Segmentation



K-means



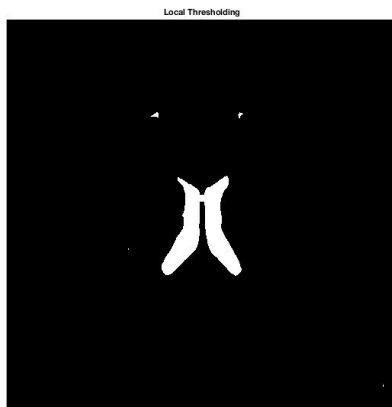
Normalized Cross Correlation and Cross-Correlation Values:

Otsu's Segmentation	0.6131
Local Thresholding	0.6162
Watershed Segmentation	0.4140
K-means clustering	Cluster 1: 0.4830 Cluster 2: 0.1554 Cluster 3: 0.7184 Cluster 4: 0.3481
Cross-Correlation value	0.8007

Input2:



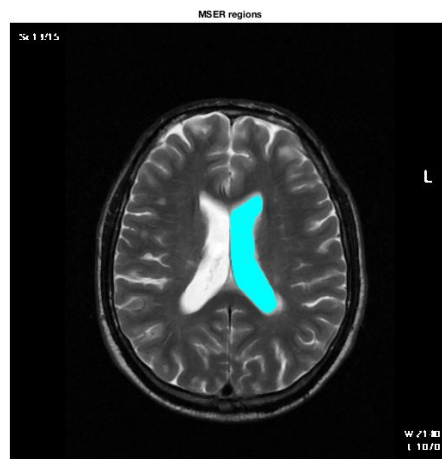
Local Thresholding



Otsu's Thresholding



MSER



Watershed Segmentation



K-means



Normalized Cross Correlation and Cross-Correlation Values:

Otsu's Segmentation	0.5867
Local Thresholding	0.6951
Watershed Segmentation	0.4945
K-means clustering	Cluster 1: 0.2172 Cluster 2: 0.4276 Cluster 3: 0.4787 Cluster 4: 0.7110
Cross-Correlation	0.7089

From the above images, Input1 manages to show fairly accurate results for all of the methods. The tumor region is represented by a white portion for Otsu thresholding and Local Thresholding. In Watershed segmentation, the tumor is represented by a gray region whereas for MSER the tumor extracted is colored in blue. For k-means, each of the cluster is sub plotted from where we can understand the cluster in which the tumor is present. The normalized correlation values that is obtained by computing the pixel values between the tumor extracted and MRI image also has values close to 1. This is because, there is hardly any noise seen in the image. However this technique is not applied on the tumor generated from the MSER technique. The reason is because in MSER, we determine MSER features and plot the regions based on the features extracted using the plot() function. Due to this, the region is not actually superimposed on the original image. The region is just highlighted to display the result. Hence it was not possible to calculate normalized cross-correlation for the tumor extracted using MSER technique. However by looking at the output, MSER produces the best result as compared to other results. Also after extracting the foreground and background portions of the image, I have determined how the pixels in the tumor region are closely related to each other by computing cross-correlation between target variable and tumor region. The tumor region represents the pixel values for the foreground points extracted using the ginput() command whereas the target variable consists of 0's and 1's where 0's represent skull region and 1's represent tumor region. This foreground and background points are obtained using the ginput() command from the texture filtered image. This texture filtered image is obtained using the texture filter command, rangefilt() method. Texture enhancement is done by applying a smoothing filter on the texture image. From the above result, we can see that the given result is highly correlated. From this we can conclude that, the tumor extracted is clear and fairly accurate

In Input2, all of the methods except MSER manages to extract the tumor correctly. Also compared to the first output, we can see some noise in the output. Due to this, the normalized cross correlation values displayed are not tending towards 1. This is because of the inefficient threshold values used to segment the tumor. This is because, there is hardly any noise seen in the image. However this normalized cross-correlation technique is not applied on the tumor generated from the MSER technique. The reason is because in MSER, we determine MSER features and plot the regions based on the features extracted using the plot() function. Due to this, the region is not actually superimposed on the original image. The region is just highlighted to display the result. Hence it was not possible to calculate normalized cross-correlation for the tumor extracted using MSER technique. However the output of the tumor segmentation is so clear that by just looking at the output, one can predict the accuracy. For eliminating the noise and display better results, better threshold values can be selected. Also in MSER only one tumor is extracted instead of both the tumors. This has resulted due to two possibilities. First is the area range provided is smaller which neglects the required tumor portion of the image. Second possibility is due to higher threshold value. Therefore, on increasing the area range and decreasing the threshold, the region of interest can be extracted accurately. In this way unnecessary noise can be eliminated and correlation values that are tending towards one are displayed. However, in any of the segmentation techniques, doing any changes may result in accurate results for some images and produce noisy output for some images. Also after extracting the foreground and background portions of the image, I have determined how the pixels in the tumor region are closely related to each other by computing the cross-correlation between target variable and tumor region. The tumor region represents the pixel values for the

foreground points extracted from the texture image using the `ginput()` command whereas the target variable consists of 0's and 1's where 0's represent skull region and 1's represent tumor region. This texture filtered image is obtained using the texture filter command, `rangefilt()` method. Texture enhancement is done by applying a smoothing filter on the texture image. From the above result, we can see that the given result is highly correlated.

6. DISCUSSIONS:

In the project, the segmentation techniques proposed have been tested only on 187 images of the dataset. These images are taken under different lighting conditions and at different angles. Also normalized cross-correlation has been performed to determine the efficiency of the tumor extracted. That is if the tumor extracted has some noise in it, the value of correlation will not tend to 1. Otherwise, the value will be close to 1. This normalized cross correlation is applied on all the segmentation techniques except in MSER. Cross correlation has been computed between foreground and background pixels of the texture feature extracted image to determine how the pixels of the tumor region are closely correlated with each other. Each methodology has some drawbacks and may work for some images and may not work for some images.

In Otsu's threshold, segmentation of the tumor completely depends on the global threshold computed from the `graythresh()` method. For the set of images tested, this methodology is successful in extracting the tumor region for only some of the images; however, for some images, the skull regions are also displayed along with the tumor. This is due to the inaccurate threshold value computed by the `graythresh` method that converts a grayscale to a binary image using `imbinarize()` method.

In Local thresholding and Watershed Segmentation, similar problem arises where for some images, small portions of external regions around the tumor is displayed along with the tumor. This can be improved by selecting a better threshold value. These techniques extract the tumor with lesser noise as compared to Otsu's threshold.

By using k-means segmentation to extract the tumor, though some of the images that have been tested show additional regions apart from the tumor region, this technique manages to extract the tumor region fairly accurate as compared to the Otsu's method. However, I am unable to display only the cluster that contains the tumor. Therefore, each of individual clusters are sub plotted to determine the cluster in which the tumor belongs. After performing k-means, dynamically understanding in which cluster the tumor will be present in order to display only the cluster image rather than sub plotting all of the clusters was not successful.

Using MSER technique to segment the tumor shows best result of the tumor region for the area range and thresholding value specified for most of the images as compared to other segmentation techniques. However, for some images that have multiple tumor regions or images that have very small tumor regions or are over exposed, this technique fails. This problem has occurred due to inappropriate threshold value or area range. By using appropriate pre-filtering techniques and selecting appropriate parameters to detect the features, one can improve the efficiency of this approach as well.

Among all of the segmentation techniques implemented, I have calculated normalized cross-correlation to determine the accuracy of the tumor extracted for all the segmentation techniques except MSER. If the value of normalized cross-correlation is nearing one, it means that the tumor extracted is accurate and is free of noise. If the tumor does not tend to one, it means that along with the tumor, some noise is also present in the image. Hence the tumor obtained is not accurate and efficient. The reason of not computing in MSER is because in MSER, after extracting the features we use the hold on option after displaying the original image to display the regions that has those features extracted. These regions are displayed using the plot() function. I tried other techniques like drawshape() method to superimpose the region permanently on the image in order to compute normalized cross correlation. However the drawshape() method did not take all the parameters generated from the detectMSERfeatures() method. Due to this, it became difficult for me to superimpose the region permanently on the image and compute normalized cross-correlation. After testing on a set of 175 MRI images of the dataset, least noise is seen using the MSER segmented technique and most noise is seen in Otsu's thresholding and Watershed segmentation. Though normalized cross-correlation is not computed on MSER technique, looking at the output of the tumor segmented using the MSER algorithm, one can realize that there is hardly any noise in the image.

Also using ginput() method, I have extracted the tumor and skull regions to compute cross-correlation between the target variable and tumor region. The target variable is a vector consisting of 0's and 1's. 0's represent the tumor region and 1's represent skull region. The tumor region represents the pixel values for the foreground points extracted using the ginput() command from the texture image. This texture filtered image is obtained using the texture filter command, rangefilt() method. In order to enhance the texture of the foreground and background region, smoothing filter is applied on the texture generated image. In this way cross-correlation is used to determine how pixels in the tumor region are closely related to each other. However again this value of cross correlation completely depends on the way we select the foreground and background points using the ginput() method. If we do not select the points properly; that is for example if the points of the tumor region (foreground) are considered at the edges or borders, there is a possibility that we may not get an accurate cross-correlated value.

Also the proposed segmentation technique fails to produce accurate results if the image is over exposed and has no tumor in the image. This is because each of the segmentation techniques segments the brightest portion of the image based on the area range and the threshold value specified. Therefore, these segmentation algorithms consider the over exposed white regions as tumor and display the needed result. Hence, it does not produce an output as expected. Also there are images where they are over-exposed and tumor regions are present. In such cases, along with the tumor regions, the over-exposed portion is also considered as a tumor which leads to inaccurate results. Similarly, this technique also fails if the image is under exposed and has a tumor present. This is because, the tumor region that is expected to be extracted will not be segmented due to inappropriate threshold value. These problems occur due to the exposure of the image. Hence if we are successful in controlling the exposure of the image by using histogram equalization or imadjust() method, there is a possibility that we could get better results.

Also my original idea was to compute correlation coefficient between the target variable and tumor region by extracting the features using feature extraction techniques like DCT (Discrete Cosine Transform) or GLCM (Gray Level Co-Occurrence Matrix) and not perform cross-correlation on an image that has texture features. That idea was not successful because I was unable to extract the proper features accurately using those methods. Therefore, this idea did not help me in computing the correlation coefficient. These problems would have occurred due to the exposure of the image. Due to this reason, I extracted the foreground and background pixels accurately and determined cross-correlation between foreground pixels of the texture image and target variable. The target variable is a vector that represents 1 for non-skull region (tumor region) and 0 for skull region. This foreground and background points are obtained using the `ginput()` command from the texture filtered image. This texture filtered image is obtained using the texture filter command, `rangefilt()` method. This computation is done to determine how pixels of the tumor region are closely related to each other. Also I have compared the efficiency and accuracy of tumor extracted after every segmentation technique. The tumor region extracted is compared with the image to check if noise is present in the image. If noise is present, we can conclude that tumor segmented is not accurate as it has some noise in it. Otherwise, the tumor segmented is accurate and correct.

7. FUTURE WORK:

In the project, currently these methodologies have been tested only on 187 images of the dataset. These images are taken under different lighting conditions and at every possible orientation of the tumor region. This project can be extended to building a classifier to classify the tumor as cancerous or non-cancerous. Different supervised classification techniques like Artificial Neural Networks or SVM (Support Vector Machines) can be used to classify the tumor. Also we could use the supervised technique Mahalanobis to segment the tumor. Since it will consider the spread and data, it may be possible that better results are obtained as compared to k-means.

Also by extracting good features using appropriate feature selection algorithm like GLCM or DCT, correlation co-efficient can be computed more accurately for every tumor extracted using different segmentation techniques from the image.

8. CONCLUSION:

To summarize everything, this project talks about segmenting the brain tumor from MRI images and computing normalized cross-correlation between the extracted tumor and MRI image. This value helps us to determine if the tumor extracted has some noise present and determines the level of accuracy of the extracted tumor. Also the foreground and background pixels are extracted from the texture image using the `ginput()` command. The foreground pixels are the tumor region and the background pixels are the skull region. Cross-correlation is computed between the target variable vector and the tumor region to determine how pixels values of the tumor region are closely related. The target variable is a vector consisting of 1's and 0's where 0's represents the skull region and 1's represent the tumor region. The tumor region represents the pixel values for the foreground points extracted using the `ginput()` command from the texture image. The segmentation techniques implemented to extract

the tumor include Otsu's thresholding, Local thresholding, k-means clustering, Watershed segmentation and MSER ((Maximally Stable Extremal Regions).

On the whole, I understood how each of the segmentation technique works. Based on its working, its advantages and disadvantages can be determined. ^[3] For example; in Otsu's thresholding, the `graythresh()` method determines the levels based on the brightest pixel values in the image. Based on the threshold value, it classifies into two classes: foreground and background. Due to this the drawback is that, many regions could be classified as foreground if the image is over exposed. ^[12] Similarly, other segmentation methodologies along with its drawback is understood clearly.

I also learnt the importance of the computing correlation between two regions of the images. ^[15] Using this technique, it helped me determine the level of noise present in the image of the segmented tumor based on the value of correlation. If value is tending to one, it means tumor extracted is correct and accurate. Otherwise there is some noise present in the image. After testing on a set of 187 MRI images of the dataset, least noise is seen using the MSER segmented technique and most noise is seen in Otsu's thresholding and Watershed segmentation. Though normalized cross-correlation is not computed on MSER technique, looking at the output of the tumor segmented using the MSER algorithm, one can realize that there is hardly any noise in the image. Therefore, the percentage of accuracy is more while using the MSER technique to segment the tumor and percentage of accuracy is less while using the Otsu's segmentation and Watershed technique to segment the tumor. It also helped me understand how to determine whether the pixels of the tumor region are closely related to each other. This is computed by determining the cross correlation between the pixel values of the foreground region of texture image and the target variable. This target variable is vector consists of two regions: tumor and non-tumor region while the foreground region represents the pixel values of the tumor region from the textured image.

In conclusion, this project taught me how different segmentation techniques can produce different output effects based on its working. Therefore, for extracting a tumor from a region of the brain, the exposure of the MRI image and characteristics of the tumor (that is location, area and the size), forms the deciding factor for selecting an appropriate segmentation technique.

9. REFERENCES:

1. <http://www.ijecscse.org/papers/apr2012/Brain-Tumour-Extraction-fromMRIImages-Using-MATLAB.pdf>
2. <http://www.electronicshub.org/matlab-projects-for-engineering-students/>
3. <https://www.mathworks.com/matlabcentral/fileexchange/51153-automaticsegmentation-of-brain-tumor-in-mr-images/>
4. <http://www.medinfo.cs.ucy.ac.cy/index.php/downloads/datasets/>
5. <http://www.medicalimaging.org/about-mita/medical-imaging-primer/>
6. [http://www.ijettcs.org/Volume4Issue5\(2\)/IJETTCS-2015-10-01-7.pdf](http://www.ijettcs.org/Volume4Issue5(2)/IJETTCS-2015-10-01-7.pdf)
7. <http://esatjournals.net/ijret/2014v03/i13/IJRET20140313001.pdf>
8. <https://www.rroij.com/open-access/detection-of-possibility-of-brain-tumorusingimage-segmentation.pdf>
9. <https://www.mathworks.com/help/vision/ref/detectmsrfeatures.html>

10. <https://www.mathworks.com/help/stats/kmeans.html>
11. <https://www.radiologyinfo.org/en/info.cfm?pg=bodymr>
12. <https://www.mathworks.com/help/images/ref/graythresh.html>
13. <https://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html>
14. https://ww5.komen.org/uploadedfiles/_komen/content/about_breast_cancer/tools_and_resources/fact_sheets_and_breast_self_awareness_cards/breastcalcifications.pdf
15. <https://www.mathworks.com/help/images/ref/normxcorr2.html>
16. <https://www.mathworks.com/help/signal/ref/xcorr.html>