Identifying Tumors in MRI Scans using Fuzzy C-Means Clustering

Matthew Bouchard  
1. Undergraduate Student, University of North Georgia, Dahlonega, Georgia, U.S.A  
Email Address: mabouc4699@ung.edu

**Abstract – Brain tumors function similar to other tumors regarding genetic malfunctions effecting cell division being corrupted causing uncontrolled and excessive growth in the body. The danger is ever-present, not only that the nervous system is vital and sensitive to damage, but also that without accurate identification of the tumorous mass, treatment cannot be delivered effectively. Withal, manual methods of identification of brain tumors are matched by algorithms and neural networks in accurately identifying tumor regions; however, counting pixels associated with tumor clusters manually is unreliable and inaccurate. The purpose of this experiment is to accurately identify tumor location in various Magnetic Resonance Imaging scans and count the number of pixels associated with the tumor region for use in more accurate identification in the future.**

*Keywords – MRI; Fuzzy C-means; Image Segmentation; Clustering; Tumor*

1. INTRODUCTION

As iterated before, tumors can reveal themselves at any point in time and are usually the product of a genetic mutation. Genetic mutation does not cause the cells to proliferate; however, as mutations continue to occur deviating from the standard cell division process, uncontrollable cell growth can occur termed hyperplasia [1]. Relating this information to the problem at hand, they can either metastasize or remain benign. Additionally, they may develop at any part of the human body. The problem emerges with identification of these tumors because identifying the location of the tumor regions with evaluating mammograms, for example, remains matched when comparing AI systems and radiologists [2]; however, evidence does suggest that AI systems and effective computer aided identification practices reduces both false positive and false negative detection rates [3]. This information reveals that the use of algorithms and AI systems for aid in detecting tumorous regions has additional benefits. Therefore, similar efficacy is hoped to be reached by this project in identifying the tumor clusters in MRI scan images using fuzzy c-means to accurately define and identify tumor regions.

1. INTRODUCTION TO CLUSTERING WITH FUZZY C-MEANS

The fuzzy c-means algorithm works by initializing cluster seeds and their corresponding centers values based on a given data set (The data set must be configured to work with the Matlab fcm function before calculations are made). After the centers of each cluster are initialized and saved to a separate matrix, the membership values (U) for all data points in the data set are calculated and saved to a matrix as well. The membership values represent each data points’ affiliation to corresponding clusters. The number of clusters used in the function can either be user defined or hard-coded and are represented, typically, by value k. The k value also relates to the number of membership values each data point contains in matrix U (as each data point has an affiliation to every cluster; equal to number k).

1. IMPLEMENTATION DETAILS

The process to utilize fuzzy c-means in this project to cluster the tumor region of each MRI image given involves the same steps indicated in the previous section outlining how fuzzy c-means works. The process starts by reading in the MRI image data into a matrix array. After reading in the image data, reshaping of the matrix is done to ensure the pre-defined fcm function in Matlab operates without error. To elaborate, the fcm (fuzzy c-means) function operates with the command:

[centers, U] = fcm(data, k)

where centers holds the values of the central points of each cluster, U holds the membership values of each data point in reference to each cluster, data is the data set given, and k is the number of clusters the user sets for the function (can be hardcoded). After performing these operations, a for loop is used to iterate through every row in the U matrix where the maximum membership value is located. This process is represented by the following code:

maxU = max(U);

for i = 1:k

temp = find(U(i,:) == maxU);

index(i, [1:length(temp)]) = temp;

end

The result of this operation is saved to a variable matrix called temp which is then used to populate a zeros matrix (called index) at row location of the iteration number of I. After parsing through every row within the U matrix, the original MRI image is displayed for user reference (unneeded and can be omitted). Afterwards, another for loop is utilized to parse through the index matrix to: find the number of non-zero elements in each row, save the positions of each non-zero element in the indicated row, hold the values indexed to the original MRI image data, effectively obtaining the original data that belongs to each cluster, and display each cluster to a separate subplot.

for i = 1:k

num = nnz(index(i,:));

sublist = zeros(reshape\_value\_final,3);

subpositions = index(i, 1:num);

sublist(subpositions, :) = mriDouble(subpositions, :);

subimg = reshape(sublist, reshape\_value\_one, reshape\_value\_two, 3);

subplot(4,4, i);

imshow(subimg);

end

The program then requests for user input using a prompt to identify the cluster that contains the tumor. This information is saved and used to perform the same operations as previously executed in the for loop; however, the iteration through the index matrix is done once using user input value c. The resulting image data is then displayed for the user to use in identifying the top left and bottom right coordinates of the tumor region. This data is gathered using Matlabs ginput() function and returns the x and y coordinates of the two data points that define the bounding box for better accurately defining the tumor region. The coordinates are used in tandem with two for loops to parse through the image data identified by the bounding box. I utilizes the nnz() or non-zero function to find the elements within that section of the image data that are non-zero and display the result as a title to the image.

[x, y] = ginput(2);

x = round(x);

y = round(y);

width = x(2,1) - x(1,1);

height = y(2,1) - y(1,1);

bbox = [x(1,1) y(1,1) width height];

grayLevel = [];

for s = y(1,1):y(2,1)

for t = x(1,1):x(2,1)

grayLevel = [grayLevel gray\_fin\_subimg(s, t)];

end

end

non\_zero\_vals = nnz(grayLevel);

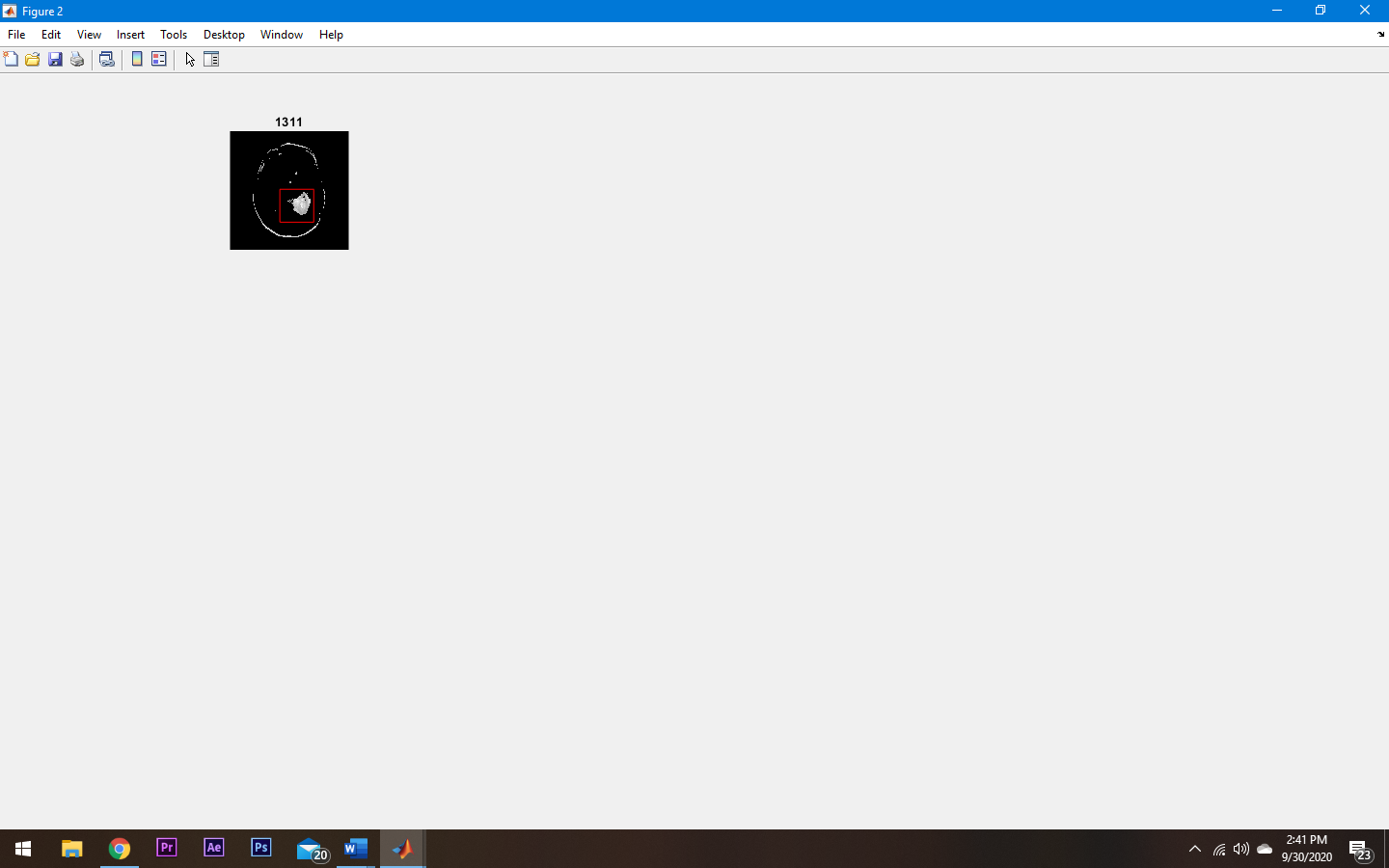
str = string(non\_zero\_vals);

rectangle('Position', bbox, 'EdgeColor', 'r'), title(str)

This result represents the number of pixels the tumor region encompasses in the tumor cluster without additional error pixels.

1. EXPERIMENTAL RESULTS

The MRI scan images are initially read into the program and are reshaped for use in the pre-defined Matlab function for executing the fuzzy c-means algorithm. The function calculates the center values for every cluster in addition to the membership values for every data point in the image data set. Afterwards, the data associated with each cluster is saved to an index array for use in displaying the clustered data to a figure. The cluster containing the tumor section is selected by the user using a prompt to specify the location of the pixels which are affiliated with the tumor region. After the user selects the coordinates of the top left and bottom right corners of the bounding box that encapsulates the tumorous region, the calculations are made to count the number of non-zero elements in the bounding box for use in displaying the result which represents the pixels only associated with the tumor region.



This result more accurately defines what elements within the MRI image represent the tumor as compared to counting the pixels located within the cluster defined by the fuzzy c-means function calculation.

1. CONCLUSION

One can conclude that utilizing the fuzzy c-means algorithm to cluster MRI scan image data to identify tumor regions is feasible and can result in a program that can handle user input to better accurately count the pixels associated with the select anomaly. The likelihood that future collaboration between machine learning and AI systems with the medical field in efforts to identify tumorous masses and other abnormalities is very likely as current data suggests.

1. REFERENCES

[1] Weinberg, R., 1996. How Cancer Arises. Scientific American, 275(3), pp.62-70.

[2] A. Rodriguez-Ruiz, et al. “Stand-Alone Artificial Intelligence for Breast Cancer Detection in Mammography: Comparison With 101 Radiologists,” Journal of the National Cancer Institute, 01-Sep-2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6748773/. [Accessed: 29-Sep-2020].

[3] McKinney, S., et al. 2020. International Evaluation Of An AI System For Breast Cancer Screening. [online] nature, (pages used). Available at: <https://www.nature.com/articles/s41586-019-1799-6> [Accessed 28 September 2020].