Tyler Hardy, Project report

**1. Introduction**

Magnetic Resonance Imaging (MRI), one of the most common internal scanning procedures in modern medicine, requires the use of image processing in order to achieve maximum effectiveness. As a result, special image processing techniques have been developed to more readily identify tumors, edemas, and other medical abnormalities. These medical conditions often lead to impairment or death. Due to the severe nature of these conditions, timeliness and accuracy are an absolute necessity.

Through highlighting areas of concern, doctors are able to provide faster and more accurate grading and diagnosis of MRI results. In the event of a noisy image, denoising algorithms may be utilized to ensure a readable image. Many researchers have found excellent results through simple image thresholding. Some have gone on to perform K-means clustering or a relatively new type of algorithm called neural networking[4]. The majority of researchers studied for this project filter the image using various techniques then apply iterations of thresholding, clustering, and/or neural networking to achieve a resultant image [1]. These results are often post-processed with further filtering to improve accuracy. In this specific program, image thresholding combined with binary image operations were used to draw an accurate contour map around brain injuries.

**2. Methods**

In order to accomplish the goal of isolating medical abnormalities in the provided brain scans, a set of functions was used. The flow of the overall program can be seen in Figure 1. The main function is called by a user at the command line. The user will also pass a number corresponding to the desired dataset for analysis from the local folder. The main function will load the dataset by concatenating the passed variable with a string and then loading the concatenated resultant string. Variables are initialized at a zero value to prevent any interference from previous iterations of the algorithm. In order to assist with statistical analysis, the first image in the dataset is then assumed to be the best current available image. A switch statement inside of a for-loop is used in order to handle selection of the current image to analyze. An unaltered version of the current image is then stored in order to ensure an accurate scan is being represented to the end-user. This is primarily an artifact from a previous version where image enhancement techniques were utilized. However, these image enhancements were found to obfuscate the medical abnormalities rather than assisting in their location.

Thresholding is then accomplished on the currently selected image using the code seen in Figure 3. Thresholding was chosen as the primary method of detection based on research into similar problems and their solutions. The given thresholding values were experimentally determined by creating a separate statistical analysis script that ran the project code against all images in all available datasets and summed the MCC values. This provided a representative value to judge each threshold set against. Threshold values were then altered and the statistical analysis script was executed. This was repeated until a maximum summation of MCC values was achieved. A similar approach was taken to determine if other algorithms would be of greater use (such as TV, Laplace filtering and edge detection, Gaussian filtering, et. al). Unfortunately, the best values were obtained with simple threshold filtering combined with binary image operations.

After thresholding is complete, binary operations are then performed on the resultant thresholded image in order to provide a more “clean” result. First, small blobs were removed from the image. Next, holes in the binary image were filled in. The remaining blobs were then dilated then eroded to provide smoothing. The resultant binary image formed the image that the future contour map would be drawn from.

From this point, statistical analysis is performed in order to ensure the best images are displayed. Statistical analysis is performed via calculating (via code in Figure 4) the Matthew’s Correlation Coefficient and the F1 Score value () of each image compared to the ground truth. As the program has ground truth data available, the MCC value is compared against historical MCC values for images in that dataset and a new “best MCC” image is selected. In the event of no ground truth being available, the program would require minor alteration to prevent this section of code from running. Multiple loops allow for the best overall MCC value to be stored in order to display the image with the best MCC as the contour map for the user. The average MCC value of every image in the dataset is also computed for debugging and algorithm improvement purposes.

|  |  |  |  |
| --- | --- | --- | --- |
| After selecting the best image from the dataset, a contour map is drawn on the selected image. MCC averages, best MCC, and best F1 values are then presented to the user. The contour map displays the ground truth image as a black outline and displays the best contour as a white line. | |  | | --- | |  | | **Figure 1** – Program Flow Chart | |

**3. Results**

Results for image processing generally met a high degree of accuracy. Generally, one image far exceeded expectations for a given set. Per Figure 2, there was a substantial deviation from the average MCC value for each dataset compared to the best MCC in the same set. For ease of analysis, code was used to only display the best contour image to the end-user as demonstrated in Figure 4. Best MCC values of less than 0.5 were present in only 2 sets of images.

It is likely that a completely different processing approach would be required in order to obtain usable data from the aberrant image sets. Due to this impacting only 13% of images, the decision was made to not make substantial changes in code for this irregularity. F1 values tended to track with MCC values and also demonstrated a high degree of accuracy within each dataset. Similar problems were noted in two of the datasets which is to be expected as MCC and F1 are both calculated from the same variables. Figures 5 demonstrates particularly high MCC value contours contrasted against the ground truth value. The ground truth will appear as a black contour while the calculated contour will appear as a white outline.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset Number | Dataset Best MCC Value | Dataset Average MCC | Dataset Best F1 |
| 1 | 0.8912 | 0.4497 | 0.8975 |
| 2 | 0.7007 | 0.2911 | 0.6706 |
| 3 | 0.7456 | 0.4793 | 0.7385 |
| 4 | 0.8073 | 0.3548 | 0.8142 |
| 5 | 0.5425 | 0.3340 | 0.4619 |
| 6 | 0.7282 | 0.5873 | 0.6957 |
| 7 | 0.2980 | 0.2180 | 0.2104 |
| 8 | 0.8621 | 0.5186 | 0.8788 |
| 9 | 0.1330 | 0.0815 | 0.2192 |
| 10 | 0.5721 | 0.1768 | 0.5014 |
| 11 | 0.8525 | 0.4492 | 0.8643 |
| 12 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 |
| 14 | 0.7590 | 0.2903 | 0.7630 |
| 15 | 0.5088 | 0.2129 | 0.4589 |
| **Figure 2 – MCC and F1 Values** | | | |

**4. Conclusions**

This program accomplished brain abnormality detection through several processes. Through image thresholding, binary image operations, and statistical analysis scripts, a set of highly accurate contour maps was developed. By overlaying these maps with actual brain scans, imaging results can be expeditiously interpreted to prevent injury and death.

**5. Acknowledgements**

Brain tumor image data used in this work were obtained from the NCI-MICCAI 2013 Challenge on Multimodal Brain Tumor Segmentation organized by K. Farahani, M. Reyes,B. Menze, E. Gerstner, J. Kirby and J. Kalpathy-Cramer. The challenge database contains fully anonymized images from the following institutions: ETH Zurich, University of Bern, University of Debrecen, and University of Utah and publicly available images from the Cancer Imaging Archive (TCIA).

**6. References**

[1] K. Verma, A. Mehrotra, V. Pandey, S. Singh, “Image Processing Techniques For the Enhancement of Brain Tumor Patterns,” IJAREEIE, vol. 2, no. 4, pp. 1611-1615, April, 2013.   
  
[2] D. Dahab, S. Ghoniemy, G. Selim, “Automated Brain Tumor Detection and Identification Using Image Processing and Probabilistic Neural Network Techniques,” International Journal of Image Processing and Visual Communication, vol. 1, no. 2, Oct, 2012.

[3] K. Somasundaram, P. Kalavathi, “Analysis of Imaging Artifacts in MR Brain Images,” Oriental Journal of Computer Science and Technology, vol. 5, no. 1, pp 135-141 Jun, 2012.  
  
[4] M. Egmont-Petersen, D. Ridder, H. Handels, “Image Processing with Neural Networks – A Review,” The Journal of the Pattern Recognition Society, vol. 1, no. 35, pp 2279-2301 Aug, 2001.

|  |
| --- |
| w\_Clean = strel('disk',4); % Disk for "rolling" binary image  D = (u>545 & u<875); D = bwareaopen(D,22);  D = imfill(D,'holes');  D = imdilate(D,w\_Clean);  D = imerode(D,w\_Clean); |
| **Figure 3** – MATLAB Code for Threshold Setting and Binary Image Operations |

|  |
| --- |
| totalMCC = project\_main(1)+project\_main(2) … +project\_main(15) … function [MCC] = project\_main(A) …  TP = sum(sum([D==1 & G==1]));  TN = sum(sum([D==0 & G==0]));  FP = sum(sum([D==1 & G==0]));  FN = sum(sum([D==0 & G==1]));  F1new = 2\*TP/(2\*TP+FP+FN);  MCC\_new = (TP\*TN-FP\*FN)/(sqrt((TP+FP)\*(TP+FN)\*(TN+FP)\*(TN+FN)))  MCC\_sum = MCC\_new + MCC\_sum;  if MCC\_new > MCC  best\_answer = D;  MCC = MCC\_new;  u\_best = u\_unfiltered;  end  if F1new > F1  best\_F1 = D;  F1 = F1new;  end   MCC\_avg = MCC\_sum/4 |
| **Figure 4** – MATLAB Code Snippet Used For Statistical Analysis |

|  |
| --- |
|  |
| **Figure 5** – Example Output Images (Ground truth in black, calculated contour in white) |