BME3053 Final Report
Team Name: Biomedical Gals
Team Members: Gianna Scibilio, Sarah Parrett
Date: 12/11/21

Identifying Gliomas through Computer Led MR Image Analysis

Introduction:

A glioma is a brain tumor that originates from glial cells that surround and support neurons in the brain. Gliomas are typically malignant, or cancerous, meaning that they can spread and destroy nearby tissue. Unfortunately, gliomas have no known cause [2].

Gliomas come in 3 common subtypes depending on the primary location and the spread rate of the glioma. Ependymomas are the least common subtype of gliomas. Ependymomas come from ependymal cells and typically don't spread into the nearby brain tissue. They are still considered malignant due to their high recurrence rate after treatment. Astrocytomas originate from brain cells called astrocytes. Most astrocytomas spread rapidly throughout the brain tissue and cannot be easily cured by surgery. Like astrocytomas, oligodendrogliomas also spread rapidly through brain tissue [9]. Oligodendrogliomas form from oligodendrocytes, the supportive tissue cells of the brain located in the cerebrum. Oligo-astrocytomas, or mixed gliomas are also possible. They are made up of more than one type of glial cell, thus being a combination, or mix, of different gliomas.

Most gliomas are not diagnosed until symptoms appear. Symptoms typically consist of headaches, numbness, and weakness. More significant symptoms include seizures, memory loss, speech problems and even vision loss. Some gliomas may appear to be asymptomatic and may take a while to be diagnosed. Others may differ in severity of symptoms as the glioma spreads over time [2]. Depending on the symptoms, the health and history of the patient, and the type of tumor suspected, different tests and forms of treatment may be performed.

In general, the initial testing for a suspected brain tumor is magnetic resonance imaging or MRI. An MRI uses magnetic fields to produce detailed images of the body. Typically, MRIs can help identify the location and existence of a glioma and even the size of the glioma. Following an MRI, depending on the location of the glioma, a biopsy, or even surgical removal of the tumor is performed, if possible. The sample received from the biopsy is carefully studied under a microscope. This allows for a final diagnosis of the type of glioma along with the grade of the glioma [1].

Gliomas themselves are graded following a biopsy to further determine the severity of the glioma and the treatment plan. The grade of the glioma differs from a cancer stage as glioma grading is dependent on tumor growth rate while cancer staging is dependent on primary location, migration, size, and lymph node involvement. Although gliomas are typically cancerous, they don't typically spread outside of the brain, causing the process of staging gliomas to be irrelevant. Instead, gliomas are graded on a scale of one to four, with one and two considered as low-grade gliomas. A low grade indicates that the gliomas are slow growing and relatively contained. Low grade gliomas are still dangerous as they can cause harm by pressing and damaging areas of the brain. Additionally, they can block the flow of cerebrospinal fluid, causing a buildup of pressure on the brain. Meanwhile, gliomas graded at a three or a four are considered high-grade gliomas. These high-grade gliomas are fast spreading to other parts of the brain. Treatment often requires more intense options than surgery like radiotherapy and chemotherapy. High-grade gliomas are also much more likely to be reoccurring and come back even after being treated [8].

Gliomas are highly dangerous due to the intense effects the glioma could have on the brain. Due to the fast rate in which gliomas spread throughout brain tissue, destroying the tissue in its path, it is important to diagnose gliomas quickly. By creating our code, we can make it easier and quicker to diagnose gliomas through a computer scan of an MR image. This eliminates the need for a doctor to go through each image individually to identify a glioma, and rather allows the computer to analyze the presence, size, and location of a glioma, and a doctor to verify these results. This saves time, allowing the

condition to be diagnosed sooner, and treatment to begin quickly. Our project will not be able to identify the grade of the glioma because grading must be diagnosed by cell types and movement indicated through a biopsy. However, our project will be able to identify the presence of a glioma, estimate the size of the glioma, and predict the scope of treatment based on location and size estimate of the glioma. The result will be an image file highlighting the glioma, and a file listing the results such as size, location and treatment scope.

Methodology:

Our approach to this project relies heavily on image filtering and analysis. We will need to be able to distinguish the gliomas from the rest of the noise in the MR images, accurately calculate the information we are looking for, and format an easy to read result. Below is an outline of our matlab approach to this.

The user inputs the filename of the desired patient, in .jpg format. All of the patient files are uploaded into GitHub, and must be in a designated folder, along with the code, that the code will be run through. As long as the patient files and the code file are in the same folder, and the code is run through that folder, it will run smoothly and you will have all of the patient files to choose from. This can be done through direct downloads, or through the GitHub desktop app. If using the GitHub Desktop app, make sure all files are fetched before trying to run the code. Then, we confirm if the file exists. If not, the code will terminate and the user will be notified that the file does not exist. This is done using an if statement, determining if the file is in the folder or not.

If the file exists, the user will choose the number of reference points for the edge of the tumor. The image will be loaded into matlab, and filtered using Imadjust to create a higher contrast, separating the edge of the tumor more clearly and allowing a greater recognizability for the user. The user will then select the edge of the tumor, choosing whatever number of reference points they input earlier. For example, if 3 points were chosen, then the user will go in and select 3 points along the edge of the tumor. This is done using Ginput, so the user can select the points while viewing the high contrast image. Using the points selected by the user, the pixels are analyzed for the lowest pixel intensity, to create a threshold filter based on that minimum. The binary image created is broken into connectivity points using bwlabn. The values are singled out based on the user selected points, and everything but this connectivity is deleted. The original image and new image are shown side by side, and the user confirms if the tumor has been outlined or not. The edge should be mostly connected, and the image should show just the tumor and no other parts of the brain. If the tumor is not accurately outlined, the code loops back to step 3 (user chooses the number of reference points) and the process is repeated.

Once the tumor is successfully outlined, the image is filled using the imfill function. The user is then asked if the tumor is filled, showing it side by side with the outlined image for comparison. If the tumor is not filled, the code dilates the image by a factor of 10, and Imfill is used again to fill the tumor. This is done in a while loop that continues until the user confirms that the tumor is filled. Bwarea is used to calculate the area of the tumor in pixels, which will be used later to determine the location.

The code will now identify a rough location of the tumor within the brain. First the tumor will be classified as centralized or non-centralized using a circle masking technique. The code will create a predetermined mask of a consistent size, and if more than 50% of the tumor's area lies within this circle, the tumor will be considered centralized, and the location variable will be set as centralized. If the tumor is not centralized, a quadrant will be determined (front left, front right, back left, or back right). Image is split into 4 equal quadrants, and the area of the tumor lying in each quadrant is calculated. Whichever quadrant has the most area of tumor, is the quadrant that the location variable will be set as. The tumor cannot be centralized AND within a quadrant - the quadrant function will only be used if a tumor is non centralized. All of this lies within an if statement, so that the location is accurate.

Next, the size (area) of the tumor will be converted from pixels to cm^2, using the scale bars given in the MR images. Based on the area (size) of the tumor, and the location calculated, a treatment scope will be determined. For non centralized tumors, the size will be evaluated first. For anything with an area larger than 6cm^2, surgery or biopsy is possible but it is likely that chemotherapy will be needed

due to the large size. For anything smaller, surgery or biopsy is a likely option for treatment. For centralized tumors, the treatment scope will always be that surgery or biopsy is possible, but chemotherapy is likely needed based on the location, regardless of the size. The glioma is highlighted using imbinarize to convert the tumor image to a fully white shape for the tumor (black for everything else), and then this is turned into a colored shape using grayconnected. Imfuse is used to combine the products of grayconnected if there are multiple tumors in the image. Lastly, the product is blended with the original image using labeloverlay, creating a colored overlay. The final output of the code will be the original MR image with the tumor highlighted, and a caption listing the size found, location estimate found, and the treatment recommendation.

Experiments and Results:

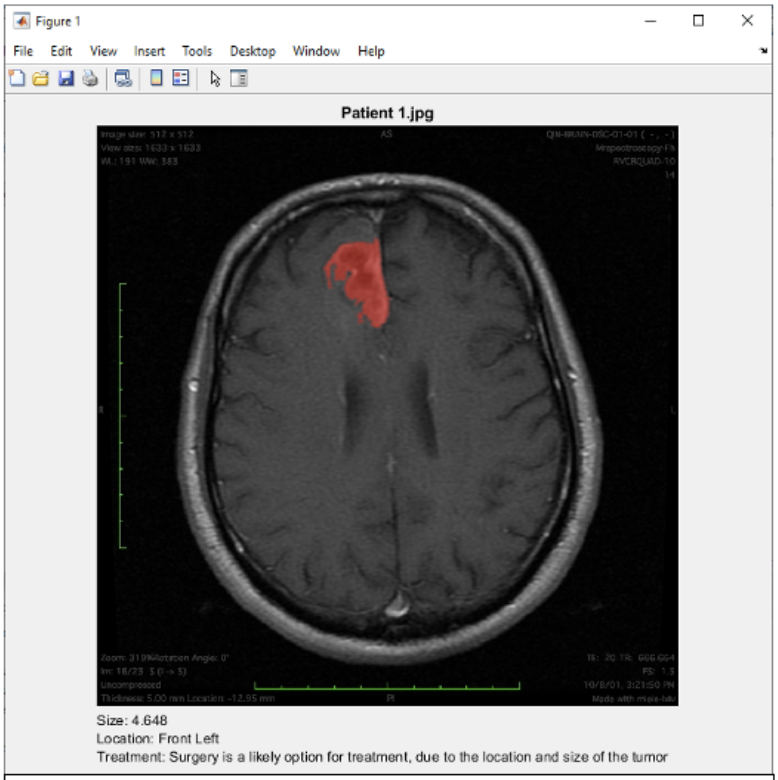

Figure 1: Final Output of the Code

From our program, our output is a figure with the original MR image of the brain that was used as the input, with a highlight over the tumor. Underneath this image is a text box, indicating the estimated location of the tumor, the estimated size of the tumor, and a treatment recommendation based on these variables. The location consists of either centralized, or a quadrant the tumor lies in. If the tumor is centralized, then the treatment recommendation is that surgery or biopsy is possible, but chemotherapy is likely needed. If the tumor is not centralized, then a quadrant is determined (front left, front right, back left, or back right), and the treatment recommendation is based on size. If the tumor is large, then the treatment recommendation is the same as centralized (surgery is possible, but chemotherapy is likely). If the tumor is small, then surgery is a likely option due to the size and location. The only issue that we have run into has been our data set. The code only works for about 50% of the MR images provided, because some tumors are difficult to even see with the naked eye, making it hard to use our code for them. There are also some images that are lower quality, making it harder for the code to recognize the tumors. However, most patients work well with our code and the ones that don't are complicated tumors that would likely need a special diagnosis anyways.

Discussion:

Separating a brain tumor from the brain itself through image filtration is a very difficult process. Several different filtering techniques and combinations were used based on previously done research [3]. Each filtration method was tested on it's own and used in different combinations.

One of the combinations of filters included the following filters: a noise reduction, top-hat filtering to increase the sharpness, and a watershed segmentation to isolate the tumor from the rest of the image. This methodology was based on the tested methods produced by Rabia Ijaz et al. in the paper "Brain Tumor Extraction from MRI Images Using MATLAB". While the process was successful for some

images, it only worked for about 40% of the dataset. Another combination of filters included noise reduction, top-hat filtering, gaussian filters, and thresholding. This and several other combinations of filters were used and inspired by GitHub user princeedey [7]. Again, while this was successful, it only worked for about 35% of the images tested.

It was determined through the help of Dr. Fang that if filtration was to properly be done to isolate the tumor, the user themselves would have to select the tumor to set limits for thresholding. This led to the final process of tumor segmentation. A high contrast filter was applied to the original image, followed by a thresholding filter based on pixel values selected by the user. At one point, thresholding was the only filter used, but it was determined through experimentation that a high-contrast filter produced better results and allowed the user more room for error when selecting points.

Another issue that was to be solved was stripping the skull and other brain debris from the selected tumor. Again, many methods were tested. One of the closer methods included skull stripping via a histogram and more filtration of the image. This was inspired by the solution given by MathWorks user Image Analyst [4]. Unfortunately, the filtration worked only partially for our images as it sometimes removed the tumor from the output. Through more research, it was discovered that the function bwlabel could also be used, which led to our final result [5].

Overall there were several barriers that had to be overcome when producing the final code. Many hours were spent exploring new functions and methodologies via the MathWorks website [6]. Several errors occurred based on the image type not working with specified functions, leading to more research on converting images from grayscale to RGB or to other image types like 8uint. Overall, it was very rewarding once a problem was overcome and the time and effort put into the code left us very proud of our results.

Credit:

Sarah Parrett completed the introduction and discussion portions of this report. Gianna Scibilio completed the methodology and results portions of this report. Coding was worked on jointly through GitHub, but majority of the code was completed by Sarah, with additions and cleaning performed by Gianna. Data collection and management, and all GitHub duties (data uploading, READ.ME file, organization, etc) were completed by Gianna. The Spotlight slides were completed by Gianna, with additions by Sarah. The presentation was done together, with editing by both. This final report was completed by both members equally. We worked very well together, frequently collaborating on all work done, reviewing each other's work and helping each other out whenever needed. We split up the work in a way that each member could work on what they felt they could accomplish strongly, and always listened and considered each other's perspectives with care.

Unfortunately, we began the project with another member, Jenny Noa, who ended up contributing minimally to the project - her only contributions were limited comments on existing work in the project proposal, and organizing information for the brainstorming milestone. She unfortunately did not contribute to the rest of the project at all, despite our best efforts of communicating with her, both directly and through Dr. Fang. Evidence of these claims is available - if needed, please feel free to contact either Sarah or Gianna.

References:

[1] *Brain tumor - diagnosis*. Cancer.Net. (2020, June 22). Retrieved October 16, 2021, from https://www.cancer.net/cancer-types/brain-tumor/diagnosis.

[2] *Gliomas*. Johns Hopkins Medicine. (n.d.). Retrieved October 16, 2021, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/gliomas.

[3] Ijaz, Rabia, et al. "Brain Tumor Extraction from MRI Images Using MATLAB." *International Journal of Multimedia and Ubiquitous Engineering*, vol. 12, no. 12, 2017, pp. 1–6., https://doi.org/10.14257/ijmue.2017.12.12.01.

[4] Image Analyst. "How to Perform Skull Stripping Using MATLAB?" *How to Perform Skull Stripping Using Matlab? -*, https://www.mathworks.com/matlabcentral/answers/172701-how-to-perform-skull-stripping-using-matlab.

[5] Image Analyst."Skull Removal of Brain Using MATLAB." *Skull Removal of Brain Using Matlab -*, https://www.mathworks.com/matlabcentral/answers/86233-skull-removal-of-brain-using-matlab.

[6] "Makers of Matlab and Simulink." *MathWorks*, https://www.mathworks.com/.

[7] Princeedey. "Princeedey/Brain-Tumor-Detection-and-Segmentation-Using-MRI-Images: This Repository Contains the Source Code in MATLAB for This Project. One of Them Is a Function Code Which Can Be Imported from MATHWORKS. I Am Including It in This File for Better Implementation.detection of Brain Tumor Was Done from Different Set of MRI Images Using MATLAB. the Concept of Image Processing and Segmentation Was Used to Outline the Tumor Area in the given Set of Images." *GitHub*, https://github.com/princeedey/BRAIN-TUMOR-DETECTION-AND-SEGMENTATION-USING-MRI-IMAGES.

[8] *Tumor grade fact sheet*. National Cancer Institute. (n.d.). Retrieved October 16, 2021, from https://www.cancer.gov/about-cancer/diagnosis-staging/prognosis/tumor-grade-fact-sheet.

[9] WebMD. (n.d.). *Brain cancer and gliomas*. WebMD. Retrieved October 16, 2021, from https://www.webmd.com/cancer/brain-cancer/malignant-gliomas.