ATI 3D Aneurysm

Final Proposal

Ahmed Almutairi

Scott Barberii

Jackson Tomaszewski

Austin Vest

ME 476C, Fall 2020

Project Sponsor: ATi

Faculty Advisor: Dr. Tim Becker, Chris Settanni

Instrutor: Dr. David Trevas

DISCLAIMER

EXECUTIVE SUMMARY

This is the final report for the 2020 fall semester of ME 476Capstone discussing the progress and future plans of the ATI 3D 2D Machine Learning Capstone team. ATI led by Dr. Becker is the client and has tasked this team with creating a program to locate and measure Aneurysms from DICOM files. This project aims to help medical professionals and patients by automatically performing initial checks of MRI or CT scans of a patient who believes they could have a potentially life threatening or nonlife threatening aneurysm. The goal is that this team's program will save everyone involved time and money. The team members each have their own unique motivations and paste experiences to bring to this project.

Similar to the preliminary report, this report will provide an overview of the project. The introduction and project description will explain all parts of the project. This section will show the motivations and importance of the project to the client, the medical community, and this team.

The middle sections of this report will go into depth on each part and how they relate and impact one another. The customer and engineering requirements will be explained and linked to one another. This will make the performance of the program measurable. This will include a functional and black box model to show the inputs and output of the program as a whole. This will give the reader and the team a big picture of the project.

The latter section will thoroughly explain the plan testing procedure that the team has discussed. These testing procedures will take place next semester once the program is operational. Each engineering requirement will be tested by at least one testing procedure. A computer program is different from standard mechanical engineering projects therefore it must be tested differently and uniquely keeping in mind the clients requirements. The testing procedure will thoroughly test all parts of the program.

The final sections of this report will discuss the current state of the project. It will talk about the current code the team has and is working on. The future plan over break and next semester will be included. The team needs to measure progress thus far and need to plan for the future. This section will also give any changes the team has had to make at this point. The implementation of this plan by the team will show how the team is motivated to lead into next semester strong.

This paragraph will briefly talk about the current progress and results of the team thus far. This capstone team as seen in the report below the team is trying to utilize deep learning in Matlab which requires several initial steps. This team has completed the research phase as seen in the Preliminary Report although MAtlab research is always being done. Code has been made to convert the Dicom files into Tiff images. This will allow for much easier use in all functions of the program. The team is currently collecting and labeling training data to train the R-CNN (regional convolutional neural network) in a deep learning process. The R-CNN needs a solid base of training data to be properly trained. Finally the team will test the neural network by input blank stacks of Dicom files to see if it finds the aneurysm. Then as seen in the testing procedure the entire program will be tested.

TABLE OF CONTENTS

Contents

1 BACKGROUND

1.1 Introduction

This is the final report for the ATI 3D Aneurysm Capstone project to present current progress to the sponsor Dr. Becker and Christopher Settanni at ATI (Aneuvas Technologies Inc.) as well as for credit for the fall semester of 2020, ME 376C Capstone class at NAU (Northern Arizona University). The ATI Aneurysm project's objective is to generate a program / algorithm utilizing MATLAB that will allow the user to input a stack of Dicom files (international industry standard for medical images) and output label images of potential aneurysm location which could also be analyzed in the same program to calculate volume and dimensions. The user or client is considered to be ATI and medical professionals who do not necessarily have a background in coding or programming. The program will involve using functions, applications, and machine learning components built into or created in Matlab. The GUI(graphical user interface) will be generated separately because Matlab has limited options for GUIs.

The importance of this project is its potential to be used by medical professionals who could efficiently use this program to locate potential aneurysms and then analyze them. This would decrease time spent by a highly paid radiologist or doctor looking at hundreds of Dicom files. Specifically, ATI is a lab that is trying to and already has developed new ways to cure aneurysms, so this program could decrease valuable time spent by the lab locating aneurysms they need to find in order to make a cure for that specific to the aneurysm. For patients who might have a mild or life threatening aneurysm, they can visit a hospital or clinic that has this computer program and find out if they need to see a radiologist and or doctor minutes after their MRI or CT scan. This would be in the future if the team's program works well enough to double check a medical professional, radiologist, or doctor. The decrease in time spent ultimately results in decreased cost for patients who are worried about a potential aneurysm and provide more confidence to radiologists and doctors.

1.2 Project Description

> "The scope of this design project is to develop a standalone algorithm to determine the size of aneurysms. The algorithm will upload medical imaging (DICOM) files containing MRI data into MATLAB (or a similar program). Voxel images will be processed to extract the aneurysm space from the surrounding brain scans."

2 REQUIREMENTS

This section will detail both the customer and the engineering requirements. The customer requirements are the goals for the team to hit that will satisfy the client. They can be as simple as the design must fall within the budget provided to the team. The customer requirements are heavily related to the engineering requirements, in that the customer determines the engineering requirements of the design, which are then polished by the team. The team does not have many because this project is computer based with less available physical measurements. The engineering requirements will allow us to measure the performance of the program as the team progresses. The functional models and QFD will continuously help the team have a holistic perspective of the project and how each part relates to the customer requirements and engineering requirements.

2.1 Customer Requirements (CRs)

1. Cost within budget
2. Reliable program
3. Input is DICOM files
4. Output location and volume
5. User friendly interface / simple enough for most to use

Above is a list of the customer requirements that the team and client created. To start the team's priority is to stay within the budget. If they cannot create a system that is within their budget, then the system is unusable or unreasonable as it can not be properly made. This would also defeat the importance of this project because the team and client do not want a program that is too expensive for a regular clinic and for the patient we aim to help.

Next requirement is that the program and UI must be reliable. If the program is likely to malfunction or produce incorrect information, it is once again useless. As engineers, the team values reliability and believe that their customers do as well. This will be measured by the engineering requirements of Time and Accuracy.

The program must be able to accept and process DICOM files. This is because these are the types of file extensions that medical professionals use and will be expected of the code. The team would like to keep the original DICOM file in use rather than convert it into a TIFF or other file type. This is in order to keep quality which could be lost in a conversion from one file format to another.

Once the DICOM is accepted the program needs to perform the tasks in the project description and produce a location and volume for the aneurysm. Is this the main purpose of the project and is one of the most important requirements that the team has.

Finally, the last requirement is that the program must be usable for people who are not necessarily tech savvy or knowledgeable in coding. The team wants anyone who has access to a DICOM file and believes that there is a chance of an aneurysm to be able to use the program with ease and produce useful information to the user and/or patient.

2.2 Engineering Requirements (ERs)

1. Volume: mm^3
2. Location: y,x,z mm
3. Time: s
4. Accuracy: ratio
5. Reliability: number of crashes, accuracy vs. time

Above are the compiled engineering requirements for the program the team is creating. The first being volume. This is referring to the volume of the aneurysm that the program is identifying. The programs are only scanning DICOM files of brain tissue, which means that the size of the vessels and vessel defects will be small, on the scale of millimeters cubed. From preliminary research the program should return a volume of 5 millimeters cubed to 30 millimeters cubed. There are many different types and locations. This will lead into accuracy because the program's return outputs should be close to the actual size of the aneurysm in the DICOM files given.

The next requirement for the program is location. This is the location of the aneurysm within the brain. The team plans to use millimeters for this application due to the size of the parts being dealt with. They also want to create a reference frame in the x,y, and z (image number) coordinates so that the aneurysm can be easily identified from a known perspective and not arbitrarily is space.

Another engineering requirement is time. This applies to both the time it takes for the program to process and produce the location and volume, as well as the time it takes for the user to interact with and run the program. Both of these are trying to be minimized, as time is very important and the faster the process can be done the better. This program will hopefully be an efficient way to do an initial check.

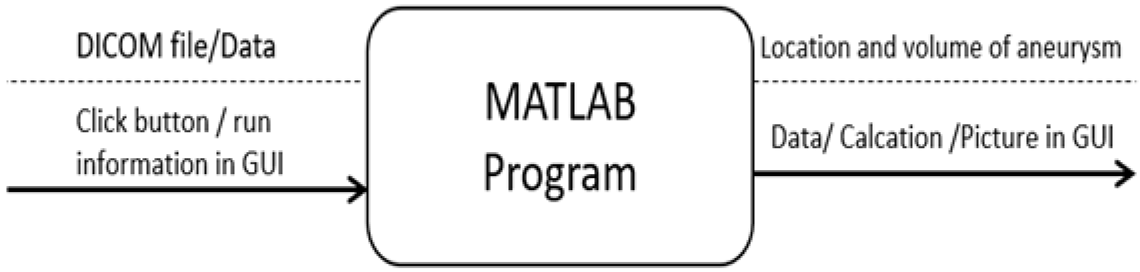
The last engineering requirement is accuracy of the program once it is trained based on how many potential aneurysms were found compared to how many aneurysms there were in the given DICOM stack. The level of this accuracy is still being debated amongst the team. The team hopes to get the number as high as they can, within 90% accuracy. This engineering requirement relates to the customer requirement of reliability. These will allow for measurement of performance in the testing phase when we input new DICOM files/ stacks into our program. The accuracy of the measured volume by the program will also be judged against the actual volume of the aneurysm.

This accuracy measurement will also be our reliability in conjunction with time. Reliability will be measured by the number of crashes per attempt or the number of times the program fails to perform the given function. The program can not crash so often that it is expected. Reliability is an important measurement for any device including this program. Yes the code should last forever, but the team hopes to make a code that will perform given tasks properly nearly every time.

2.3 Functional Decomposition

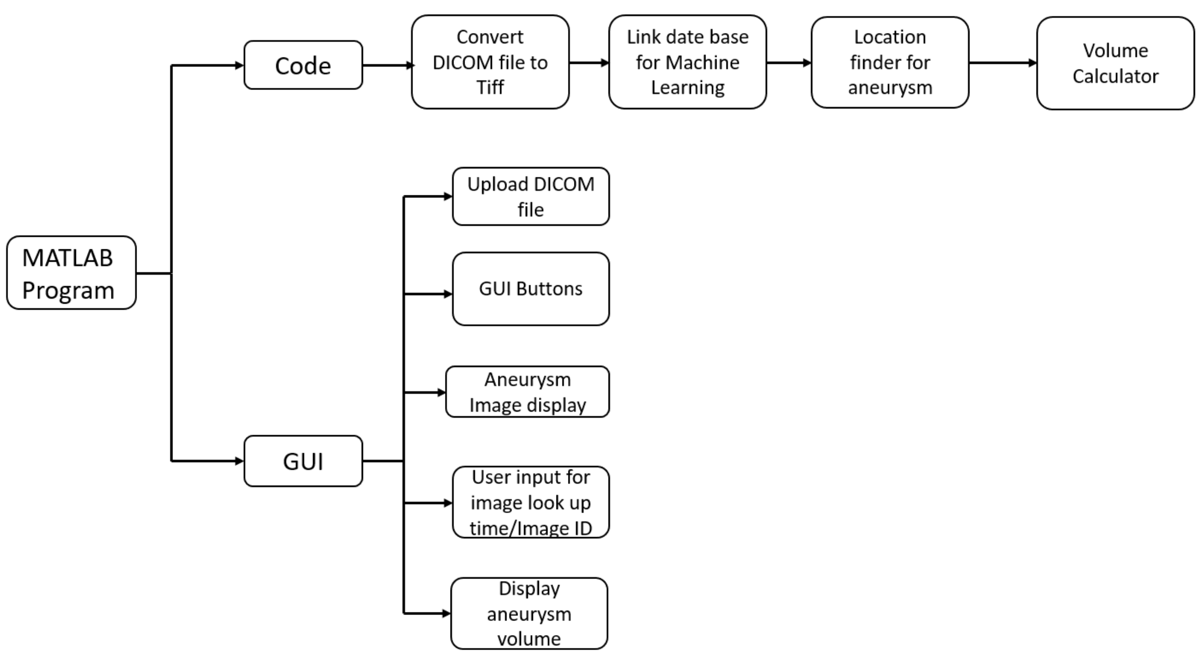
2.3.1 Black Box Model

In the black model the team focused on design and developed a standalone algorithm program in MATLAB that can identify boundaries of aneurysm and measure vascular defects. For the design inputs there are two types, the first one is information input which imports a DICOM file to the MATLAB program. Second type is a physical input of a click button /run information in the GUI. This process allows untrained medical professionals who are not familiar with MATLAB to be able to import their own DICOM files, which contain a suspected aneurysm, and display the aneurysm in the program. Results that will be obtained from the program are location and volume of aneurysm. Users will be able to display the detected aneurysm in the GUI which will help medical professionals find location and calculate volume of aneurysm.



2.3.2 Functional Model

The functional model below represents the work process diagram for our final program design systems. It is split into two different parts, the code which can be thought of as the behind the scenes portion of the program, and the graphical user interface component, which is the part that the user interacts with. In the code process there are four distinct different functions.The first work process in the code part is to convert the DICOM file to Tiff format in order to make the general process easier. After locating the aneurysm in the images, dates are going to be linked in the database for machine learning to provide future reference for any change made by this team's program. Finally, once located the volume finder can calculate the volume of the aneurysm. This is all out of sight to the user, who will only see the final products from each of these functions: the 3-D DICOM file displayed, and the location and volume of the aneurysm if there is one. The GUI is where the user will provide inputs to and receive outputs from. It should be simple to use and uncluttered as is a requirement mentioned in the above customer requirements. It will provide the displays of DICOM files and where the aneurysm is to the user. A great way to think about the code and GUI and their relationship is that the code performs all of the engineering requirements which are then transformed into understandable visuals and data by the GUI.



2.4 House of Quality (HoQ)

A QFD is a tool used by engineers to sort through their ideas and organize them into categories. The team utilized this tool to compare their requirements and constraints.

The QFD is the aid to the team in deciding what were the most important aspects of the project. The values seen in the QFD are tabulated from what the team assumed would be the general consensus as well as what other medical professionals are saying. It was found that accuracy is the most important aspect of the engineering requirements. This follows through with the scope of the project. It needs to be accurate otherwise it will not be useful to medical professionals. The least important engineering requirement was found to be the time it takes to process information and use the program. While it is still important to not be a time consuming process, the team would much rather have it take 3 minutes but provide accurate information than have it take 5 seconds but produce incorrect results. This relates back to accuracy in that the team values accuracy the most. The full QFD can be found in the appendix of the report.

As for the testing of the engineering requirements, most are very straightforward. Due to the nature of this project the team can perform tests in a matter of seconds. By running the given section of code that is going to be tested and comparing it to results the team already knows, the team can complete tests for each requirement in under a minute. This is very important for the project because the team will likely be running tests all throughout the next semester. By constantly testing the program the team can be constantly refining it while testing. Things like the location can be run and then compared to a known location provided by the client. This way the team can judge how the program is fairing with a specific task.

2.5 Standards, Codes, and Regulations

Standard and Codes for this project are difficult to address because we do not know the full reach of this program but the team does want to recognize Dicom standards. The standards from Dicom Standards.org as seen in the table below mostly describe the confidentiality of the patient. PS3.15 mainly shows how any application working with Dicom files must have security protocols. PS.31.1.4.3 discusses how Dicom files need to be kept unique to a specific patient. The team does not want the user to have a chance to mismatch two different patients. The team is addressing this concern by giving the inputs and the outputs the same tracking number so that any outputs made by the program can be linked to an inputed Dicom file. The team is protecting patient confidentiality by giving any Dicom files given to us. Discession is also given by the medical professionals dealing with any Dicom files including this team. The team often references Dicomstandards.org when questions arise regarding the patient since we are not medical professionals.

Table 2.5 A: Standards of Practice as Applied to this Project

Standard Number or Code	**Title of Standard**	**How it applies to Project**
DICOM PS.31.1.4.3 2020d	Introduction and Overview	Patient records need to be kept unique so that different files are not confused with one another.
DICOM PS3.15 2020d	Security and System Management Profiles	Discusses how any application needs security policies inplace.

DICOM PS3.2.7.6 2020d	Conformance	Patient information needs to be kept confidential to ensure safety of patients.

3 Testing Procedures (TPs)

To ensure that our program is accurate it is important to test it with known values and do multiple checks on what the program outputs. Our first test is to check the volume of an aneurysm that is detected using our program. To do this a DICOM file will be sourced which contains the actual volume of an aneurysm. With this our program will process these DICOM images and calculate the volume of the aneurysm. This will then be compared to that of the known volume. The next testing procedure is verifying the location of the aneurysm in the image. This is done using the known location of the aneurysm and comparing it to what the machine learning process found as the aneurysm. If this is incorrect more training data will be input and possible changes will be made to the labeling of the aneurysm. The third test is finding the time the program takes to complete a cycle. To test this the program will be run multiple times with various sized stacks of images. in doing so each function of the program will be tested on various computers and the average time will be taken to complete each task. The fourth test is checking the overall accuracy of the program. This will be done by finding the combined accuracy of both the volume check of aneurysm and location of aneurysm. The last check is the reliability of the program. To complete this check the program be overloaded to determine when it will crash. This check will also include dicom stakes to be run multiple times through the program on multiple computers. Where our team will take note if the program crashes or cannot complete a take at any time throughout the process.

3.1 Testing Procedure 1: Volume Check of Aneurysm

An engineering requirement for this project is to find the volume of an aneurysm using our ATi MATLAB program. Finding the actual volume of the aneurysm is a very challenging task because our program has an input of DICOM images. In the initial statement made by Dr. Becker and Chris, they said that a volume could be tested in their lab using an artificial aneurysm. An issue with this is that there are no DICOM files that accompany the model. One method of testing that our team came up with is to find a stack of dicom files, from an outside source, where the size of the actual aneurysm is already known. In doing so the program calculated volume will be able to be compared to that of the actual.

3.1.1 Testing Procedure 1: Objective

A volume calculation of an aneurysm is essential inorder to treat the patient. It will allow the doctor to choose the correct size coil to fill the bubbled out area and prevent further injury to the patient. The process of testing our program for the correct volume will first start with inserting Dicom files into our program. These DICOM files will be associated with an aneurysm that already has a known volume. Once the files are inserted into our program it will be run and the volume will be calculated. The calculated volume of the aneurysm will then be compared to that of the actual. A percent difference will be calculated, showing the error of the volume calculation. To have accurate results multiple dicom files with known aneurysm volumes must be sourced.

3.1.2 Testing Procedure 1: Resources Required

To complete the test to determine if the volume calculation is accurate or not there are a few required resources. The first step is to have a working MATLAB program that will accept DICOM files, find the aneurysm and calculate its volume. Arguably the most important step of the process is to have a user that will upload the dicom images and run the program. Without this step the entire process would not be possible. Another important resource that must be acquired are dicom files of aneurysms that have a known volume. With that there must be multiple stacks of dicom files so that the program can be tested on different size and shapes of aneurysm. Sourcing these stacks will be difficult but we will reach out to Dr Becker, Chris, and radiologists to help us find this information.

3.1.3 Testing Procedure 1: Schedule

This task will be the last part of the program to be tested, due to its place within the model. This will be tested in the second half of the second semester, after the team has completed the program to locate the aneurysm. The test should take no longer than 10 seconds for the program to calculate the volume of a given aneurysm.

3.2 Testing Procedure 2: Location of Aneurysm in Image

Location of aneurysm is one of the engineering requirements that it's important for our design to ensure that aneurysm parts are located in the right place. The test process for finding the location of aneurysm in image can be done by have stack of DICOM file running into Machine learning.The team will compare the outputs from machine learning and comparing the place of aneurysm in the image manually to know how accurate the outputs form the machine learning.

3.2.1 Testing Procedure 2: Objective

The test will focus on the accuracy of the machine learning to find the aneurysm in images. The team will test multiple stacks of images that have an aneurysm part in machine learning. In the machine learning program in our design we built a training date that programmed to find the location of aneurysm in images. These training dates are designed with a stack of images that have aneurysm labels manually, so machine learning can recognize all different shapes of aneurysm in the image. The test process is to compare the result of location of the aneurysm place from machine learning with the image that aneurysm is already located by a specialist.

3.2.2 Testing Procedure 2: Resources Required

The test will require training dates that are built in MATLAB with multiple images of aneurysm that have different shapes and location.These images of aneurysm can be found in the internet or by the client dr.Becker.

3.2.3 Testing Procedure 2: Schedule

This is the next test the team will be performing in the current time line. Once the CNN has been trained to locate an aneurysm in a file, the team can simply run the network on new files where there is known whether or not there is an aneurysm. This will be tested in the beginning of the second semester. While the test itself should only take 5 seconds at most, there will need to be many tests done to calculate the accuracy of the CNN.

3.3 Testing Procedure 3: Time to Run Functions

The Run Time test is related to the Customer's desire for the program not to take too long to run and the Time engineering requirement. The program will take some time to run, but as seen in the objectives and description of this project, this program should be an easy initial check for potential aneurysm patients and medical professionals.

3.3.1 Testing Procedure 3: Objective

This test will test the amount of time it takes to run different functions of the program. Each function will be tested five or more times on different computers with three or more Dicom stacks with aneurysms. The time to run each attempt will be timed internally (if possible) and by hand and recorded. An average time for each function will be calculated. If a function takes too long the program will be adjusted accordingly. If major changes to the code are made, that part will be retested. Testing all parts of the program will ensure fulfillment of the Time engineering requirement. Time is a valuable resource to any user and needs to minimize.

3.3.2 Testing Procedure 3: Resources Required

This test will require an accurate method of keeping time and recording it. This will probably be done in Excel. Two or more team members will be needed. As seen above, three or more stacks of Dicom files will be needed. These stacks will be used for multiple tests. A consistent stop watch and a consistent user.

3.3.3 Testing Procedure 3: Schedule

As with most of this testing, testing will be done once the program is functional. This testing should take less than three four hour sessions. Formal testing should take place next semester, but informal testing will be done continuously.

3.4 Testing Procedure 4: Total Accuracy

This test will combine the accuracy of the Volume Check of Aneurysm and Location of Aneurysm in Image. This is the overall accuracy and measurement of performance for the whole program. Time will also be taken into consideration. This test will allow the team to have a more average way to measure if the program is functioning correctly.

3.4.1 Testing Procedure 4: Objective

This is not a separate test on its own but a combination of the other tests. This will take the form of a weighted average of Volume and Location accuracy as well as a time consideration. As discussed in the Volume section, the program will measure the volume of the aneurysm and the team will compare that to the known volume to come up with a percentage. As seen in the Location test, a similar process will be done to make a percentage of how close the program calculated the aneurysm to the known location. Time will be different because the Time Test will have to be done first. Once an average run time is found, then each test iteration run time can be judged against the average run time. For example if the run time for that iteration was lower than the average then the time percentage will be above one and below one if that iteration takes longer than the average time. All three of these will be inputted into the weighted average where location will be given the largest weight and time the lowest because volume probably will not be accurate if the location is off. Again this will allow the team to take these three factors into account. For example if the location and volume is accurate but it takes above average time, the weighted average will still calculate that attempt as significantly above average. This will be adjusted and added to as the team sees fit.

3.4.2 Testing Procedure 4: Resources Required

This test will require excel to record data and calculate weighted average. The team will need to provide input and check calculations as this is an important test for the whole program.

3.4.3 Testing Procedure 4: Schedule

The data for this test will be compiled as the other tests are being performed. This will take place in the second semester. This will take several sessions spanning over a week to calculate and then check the results. The results will also be analyzed to see which functions of the program and inputs change the overall score of that iteration.

3.5 Testing Procedure 5: Reliability

This test will run the different parts of the program multiple times with various inputs and record if the program runs properly or crashes. This test will be completed formally as its own test but also informally as the team tests other parts of the program. Reliability is a common test in an engineering project but it is usually tested differently. Reliability is related to accuracy and time but is unique enough to be its own test.

3.5.1 Testing Procedure 5: Objective

This test should measure the number of crashes per attempt for the different functions built into the program. This will be tested by running or using a certain part of the program for a relatively large number of attempts under various conditions and inputs. Five or more different dicom stacks will be run through the program. The program will be run at least ten times with each stack. The program will be open and closed ten or more times on different computers. Other tests may be added as needed and reliability will inherently be tested continuously as the team tests other engineering requirements. The team and the client need to know that the program will run reliably under all usage. These tests will be slightly different but are essentially trying to find any possible conditions or inputs that would cause the program to fail consistently.

3.5.2 Testing Procedure 5: Resources Required

There are few additional resources needed for this test. Most of the required resources will have already been found for the other tests. This test will require five or more stacks of unlabeled Dicom scans with aneurysms. Two team members will need to need to be present. At Least two PCs will be used. Those PCs will need to have MATLAB installed.

3.5.3 Testing Procedure 5: Schedule

This set of testing should take two coding sessions or less. These tests will be completed once the program is finished which appears at this point to be some time next semester. Again this is the formal testing but this will be tested throughout the coding process.

4 Risk Analysis

To introduce this team's risk analysis section, a FMEA was completed as seen in Appendix B. This was completed based on the provided slides on risk analysis. Since this team is building a computer program and not a physical model, the code was broken down into subsystems and analyzed. The team chose the four main subsystems to be GUI, the R-CNN, Volume Calculator, and the Dicom to Tiff file converter. Different parts and functions of each were analyzed and given risk values. In the Trade Off section, the potential failures are related. Since this team's computer program does not have very many different parts, fewer parts and functions were found to analyze.

4.1 Critical Failures

The critical failure can be seen in the table below. These are the failures that received the highest score in the three critical categories of severity, occurrence, and detection. Incomplete conversion of all DICOM files in a stack would result in the R-CNN detector not scanning all images of interest. This could result in missing parts of the scan and potentially miss a life threatening aneurysm. This scored very high in severity and in detection. The solution is to have testing be completed by the team and having a pop up message to the user reminding them to check that all files were converted.

The next failure is incorrect training of the neural network to look for objects other than aneurysms or not look for all objects that could be an aneurysm. This received a high severity. This is the main reason the team is taking so long to choose the best neural network and label all the training data and trying to get as much training data as possible. These are the solutions that will help the most, but this program and any program will not be a stand alone aneurysm test. Any program will always need a trained professional to check like a doctor or radiologist. This applies for all critical failures and the program as a whole.

An image label not being saved after the R-CNN creates it is the next critical failure. This is because the user would have to run the program again or not even notice a label is missing when the labeled images are later referenced. Again missing potential aneurysm is given a high severity rating. This is also semi-user dependent. The solution is to check this function fully tested to make sure saving the labels is simple and reliable. The team will also explore making an algorithm or code to automatically save the original labels with and in the converted Tiff images.

Incorrect Execution of the whole program has some need for the user but mostly on the code. The main concern is any main part of the code not working properly but making the user believe it is working fine and therefore decreasing detection. This failure will be severe and undetectable which could result in aneurysms not being located properly. This will be tested reliability and accuracy testing to make sure the program performs not just for this team but for the clients and stakeholders.

Incorrect Location of final aneurysm label by the neural network. This could cause inaccurate results both in label location but also volume calculation. This will be tested by all the tests and will be detected by the team throughout the testing process.

Table 4.3.1.a: Critical Failure Data

Critical Failures					
Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	Potential Causes and Mechanisms of Failure	RPN	Recommended Action
Incomplete Conversion	Code Failure	Whole stack would not be processed. Possible undetected anurysm	Incorrect MATLAB code	105	User message. Check number of files processed
Incorrect training	Designer error	Output inncorrect. Wrong Location	Not enough training data	200	As much correct training data as possible. Check uncertainties with a medical professional
Image label not saved	Incomplete training	Less accurate	Coder error	108	Double check labels as they are saved
Incorrect Execution	Wrong detection	Double Labels or no label	Incomplete training data	360	Extensive testing on multiple known DICOM files
Incorrect location	Coding Error	Inaccurate Volume	R-CNN output	96	Check R-CNN Accuracy for attempt

4.2 Risk Trade-Off Analysis

This paragraph will discuss how each critical failure relates to one another and how they will affect one another. Since all of the above critical failures are a part of the same program, they are all related. Incomplete Execution is the main failure that will affect the program outputs and reliability. If the neural network does not find all the aneurysms then the labels will not be corrected and the volume will not be calculated correctly. This is good and bad. This program relies heavily on the training of a deep learning processed neural network. This is why most of the testing will be regarding accuracy and reliability. The trade off is training the neural network to locate all potential aneurysms without labeling every single oval it sees. Much training data will be needed. Another trade off is time to perfect this program, but this is always a trade off.

The other main trade off is efficiency being thorough. This program is supposed to save the patient time but the user needs to check their work and occasionally the programs. This comes down to having too many pop-up windows reminding the user to check the outputs of the program because every pop-up window requires more time. Again the user is supposed to save time with this process not make them feel like they have to do everything twice. All of these need to be kept in mind by the team moving forward especially during the testing phase.

5 DESIGN SELECTED – First Semester

Throughout this semester our team has made progress in creating a program that has inputs of DICOM files and then locates, calculates volume and displays an aneurysm. Thus far our group has successfully created a MATLAB script that converts an entire selected folder of DICOM files into tiff format images. These images are currently being processed in the image labeler tool box to locate the aneurysm and save the data into a R-CNN. The neural network allows for the detection of an aneurysm after many data points have been acquired. Furthermore these functions will be the backend code for a user friendly interface. This interface will allow the user to upload stacks of dicom images and run the MATLAB code for the R-CNN and all calculations. The interface will also allow the user to save the results of the processed images.

5.1 Design Description

This paragraph will talk about the current progress and results of the team thus far. This capstone team as seen in the report below the team is trying to utilize deep learning in Matlab which requires several initial steps. This team has completed the research phase as seen in the Preliminary Report although Matlab research is always being done. Code has been made to convert the Dicom files into Tiff images as seen below. This will allow for much easier use in all functions of the program. The team is currently collecting and labeling training data, in the Image Labeler application as seen below, to train the R-CNN (regional convolutional neural network) in a deep learning process. The R-CNN needs a solid base of training data to be properly trained. The GUI is also being constructed as seen in the image below. Finally the team will test the neural network by inputting blank stacks of Dicom files to see if the program locates all of the potential aneurysms. Then as seen in the testing procedure the entire program and the individual part and functions will be tested.

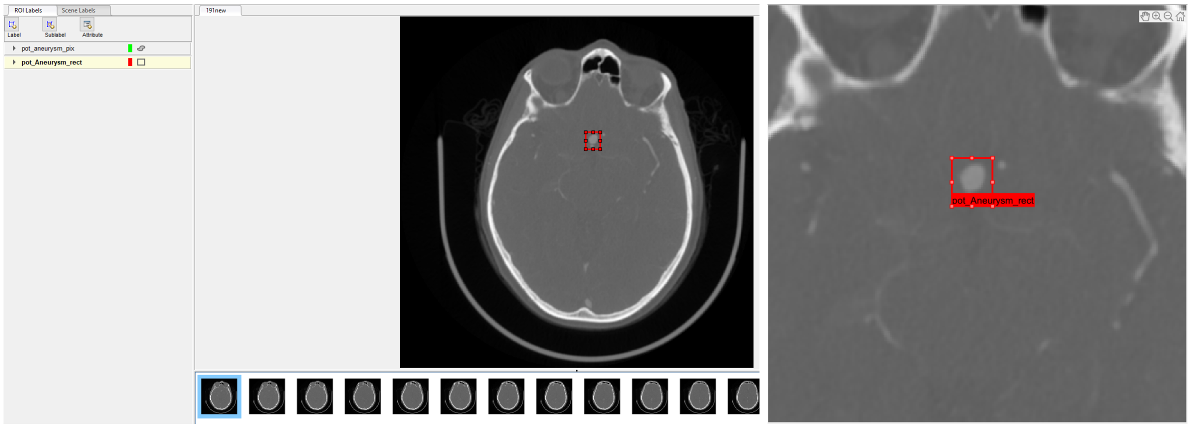


Image 4.1.a: Labeled Training Data in Matlab Image Labeler

As seen in the functional model, there are two main parts to this program/algorithm. The first is the main code in Matlab discussed above. The second is the GUI (graphical user interface). This is the window that the user will see and interact with. It already has some main features and will in the future have all the necessary features to utilize the program completely. The loading bar has been encoded to give the user feedback as seen in the figure below about the progress of the process. The GUI is currently trying to be linked to the Matlab code. It is being made in a different language because of Matlab's limitations.

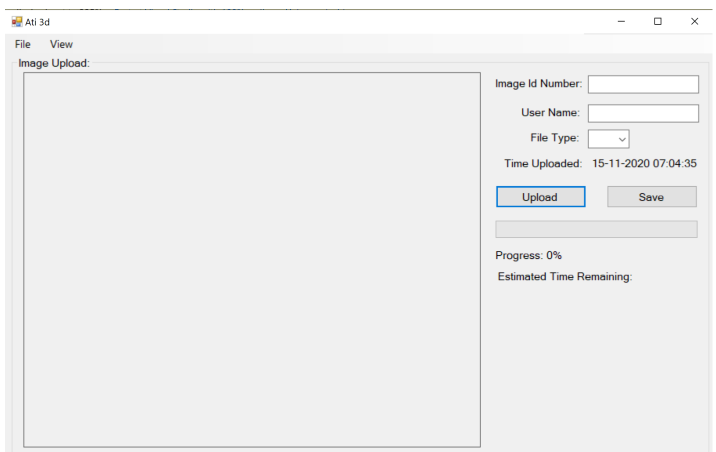


Figure 4.1.b: Current GUI and Loading Bar

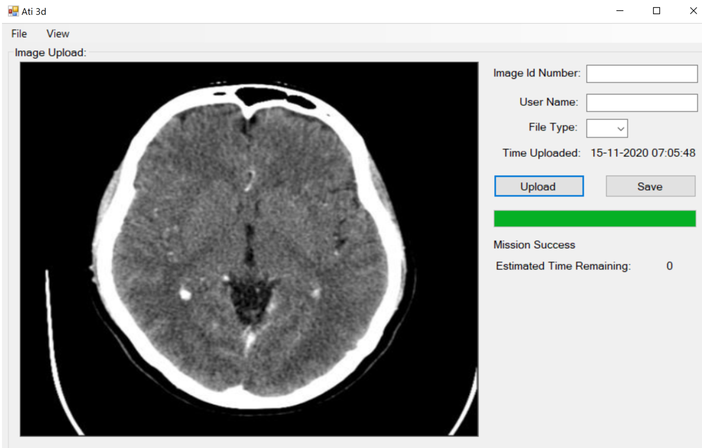
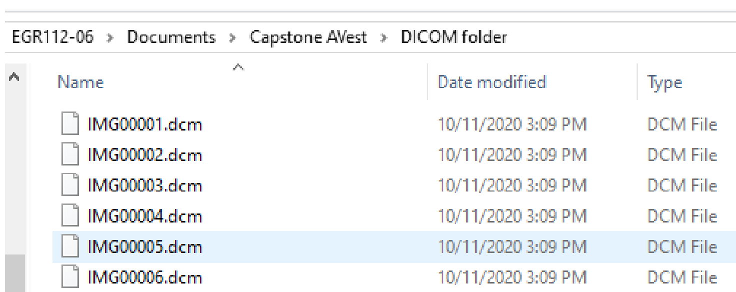


Figure 4.1.c: Successful Loading of Image Slice

The main change the team made stems from the Image Labeler application in Matlab not opening Dicom files at all. Dicom files are supported by Matlab but as the team learned through trial and error, not in this specific application. Therefore, the team decided to go ahead with the creation of an automated Dicom to Tiff converter. This set the team back slightly while the converted was being coded. The converter needed to convert the entire stack not just one image at a time. Now that the team has overcome this challenge, the program will be able to convert any Dicom input into a usable Tiff format with minimal quality loss. This code implementation into the larger program takes pressure off the user to convert the Dicom files on their own through a third party. It also opens more possibilities to many neural networks, not just the ones that run Dicom. The team wishes this challenge was initially realized but now has a useful Matlab solution.




```
%% convert all files in that folder
for i=1:260

%filenameee=preI(i);  Z:\Documents\Capstone AVest\DICOM folder
allfiles=fullfile('Z:\','Documents','Capstone AVest','DICOM folder',fileNames);
preI(i)=allfiles(i)
location=char(preI(i))
I=dicomread(location);
 imwrite(I, sprintf('%dnewish.png',i));%'brain(20).tif');

end
```

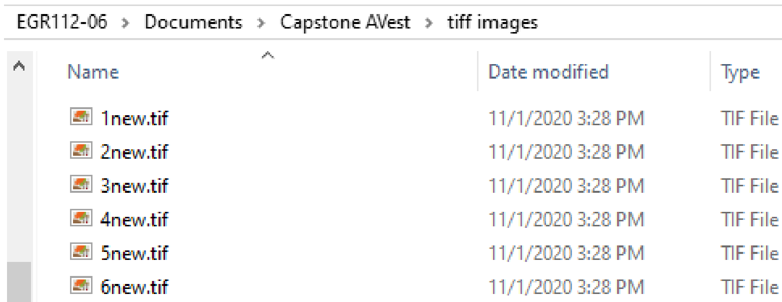


Image 4.1.d: Dicom to Tiff Converter and example

5.2 Implementation Plan

The implementation plan for next semester revolves around finishing the code. The main parts of the code are training the R-CNN so the team can test its accuracy and finishing the GUI and linking it to the Matlab code so we can test inputs, functions, and features. The team plans to be mostly done training the R-CNN neural network over break so that we can have a trained neural network to test next semester. Much of the GUI will also be done over break. These over break tasks will require team meetings once a week and access to PCs with Matlab. Some of the work may be done in the engineering building computer lab.

Next semester, within the first couple of weeks the program will be ready and testing will begin as explained in the testing section of this report. Some of the tests can be done at the psame time but may take up to a week each. Testing will require multiple PCs that can run Deep Learning in Matlab efficiently as well as three or more stacks of blank Dicom files that the team knows the location, size and quantity of aneurysms in. Another week will be needed to compile all final results and data. These will be analysed and explained in the final reports for next semester.

6 CONCLUSIONS

This is the conclusion for the final report for ME 476 Capstone as a reference for this team and its stakeholders in the fall semester of 2020. The main objective or task of this project is to create a program, or algorithm, utilizing machine learning in Matlab that will intake a stack of Dicom files then locate and measure all potential aneurysms. This program, as specified by the client, must be user friendly for their own use and even the use of other medical professionals. This project aims to increase the efficiency for medical professionals and decrease the cost to patients who are worried they might have a potentially life threatening aneurysm. More detail can be seen in the early sections of this report.

The main requirements on this project, as seen in the Customer Requirements and Engineering Requirements, are that the program can be used by any medical professional. It is also required that the program is accurate and conservative enough to not completely miss an aneurysm in multiple slices of a MRI or CT scan. This is an important part of this report and has been adjusted as the team reevaluates the project. The middle of this report also contains the Functional Decomposition generated by the team to provide clear inputs and outputs as related to different functions and parts of the program.

Testing procedures the team plan to start early next semester are explained in the testing section of this report. These tests will be completed formally but also informally as the team codes each required section. The main tests are based on the core engineering requirements so that the team can measure the performance of the program as it is built. The testing process hopes to test every part and function of the code to ensure it works reliably as intended.

The risk analysis section gives an overview of the FEMA the team completed. The Critical Potential Failures are discussed individually. This brings to light the failures that will cause the team the most concern. These inform the team's current design by showing the team where failure is most likely to occur. These areas are areas of the program that will be tested the most. This section also discusses some of the main trade-offs. Money is not a large concern so time is the main factor. The risk analysis again helped the team gain a broader understanding of the different parts of the program and what to test more thoroughly.

The final section demonstrates the progress of the project and how the team is going to implement our future plan in ME 486C at NAU. This team is challenged with generating a trained neural network linked to a user friendly GUI that can locate and measure potential aneurysm. The main code will be done in Matlab while the GUI will be created in Visual Studios. The uniqueness challenges the team to learn about topics we would not have learned otherwise gaining us experience that will help these team members in our future career. This project has interest for each member, but more importantly it has great potential to help the medical industry which will motivate us throughout the break and following semester.

7 REFERENCES

[1]Medical Imaging & Technology Alliance, “Current Edition,” *DICOM*, 2020. [Online]. Available: https://www.dicomstandard.org/current. [Accessed: 13-Nov-2020].

[2]“Image Labeler,” *Getting Started with Object Detection Using Deep Learning - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/vision/ug/getting-started-with-object-detection-using-deep-learning.html. [Accessed: 18-Oct-2020].

[3]Settanni C. (2020) Patient 3D Image Scans-Analyzing the Volume of Vessel Defects [PowerPoint presentation]

[4]“Image Processing Toolbox,” *Image Processing Toolbox Documentation*. [Online]. Available: https://www.mathworks.com/help/images/index.html?s_tid=srchtitle.

[5]“alexnet,” *Pretrained Deep Neural Networks - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html. [Accessed: 18-Oct-2020].

[6] “Image Labeler,” *Get Started with the Image Labeler - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html. [Accessed: 18-Oct-2020].

8 APPENDICES

8.1 Appendix A: QFD with Testing Procedures

			EGR requirements		
Customer Requirements	**Importance (1-9)**	Volume	Location (x,y,z)	Processing Time	accuracy
Find location of aneurysm	9	3	9	1	9
Find Size of Aneurysm	7	9	3	1	9
Easy to Use	6	1	1	9	1
Accept DICOM Files	5	1	1	3	1
Cost within Budget	7	3	3	3	3
Technical Importance: Raw Score		17	17	17	23
Technical Importance: Relative Weight		0.2267657993	0.249070632	0.1970260223	0.3271375465
Techanical Target Value				60 sec	90%
Upper Target Limit				5 min	99%
Lower Target Limit				5 sec	60%
Units		mm^3	mm	seconds	ratio
Testing Procedure (TP#)		Run code and compare to known		Run code to get estimate	Compare the ratio to actual

8.2 Appendix B: FEMA Risk Analysis

Product Name: ATi Aneurysm locator		Development Team					Page No 1 of 1			
System Name: MATLAB Code & GUI							FMEA Number:1			
Subsystem Name: File Converter							Date: 11/15/2020			
Component Name										
Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	Severity (S)	Potential Causes and Mechanisms of Failure	Occurance (O)	Current Design Controls Test	Detection (D)	RPN	Recommended Action	
Choose Folder	Choose wrong folder	Incorrect DICOM file to analyze	10	Multiple options to select. User input erro	4	Time Test	2	80	None	
Wrong File Conversion	Code Failure	Inncorrect output type	10	Incorrect MATLAB code	1	Reliability	1	10	None	
Wrong Input File Type	Code Failure	Program would not run	7	User input error	2	Reliability	1	14	Add a error in GUI displaying wrong input type	
Incomplete Conversion	Code Failure	Whole stack would not be proccessed. Possible undetected anurysr	7	Incorrect MATLAB code	3	Reliability/Volume	5	105	User message. Check number of files processed	
MATLAB	Program Crash	inoperable code	5	Insufficient computing power	3	Reliability	1	15	None	
Product Name: ATi Aneurysm locator										
System Name: MATLAB Code & GUI										
Subsystem Name: R-CNN										
Component Name										
Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	Severity (S)	Potential Causes and Mechanisms of Failure	Occurance (O)	Current Design Controls Test	Detection (D)	RPN	Recommended Action	
MATLAB	Program Crash	Inoperable code	5	Insufficient computing power	3	Reliability	1	15	None	
Input tiff files	Error To user	Program would not run	2	Code error	2	Reliability	1	4	None	
Incorrect training	Designer error	Output inncorrect. Wrong Location	10	Not enough training data	4	Accuracy	5	200	As much correct training data as possible. Check uncertainties with a medical professiona	
Image label not saved	Incomplete training	Less accurate	6	Coder error	3	Accuracy	6	108	Double check labels as they are saved. Automatically save labels with tiff images	
Incorrect Execution	Wrong detection	Double Labels or no label	9	Incomplete training data	5	Accuracy/Location/Volume	8	360	Extensive testing on multiple known DICOM files	
Product Name: ATi Aneurysm locator										
System Name: MATLAB Code & GUI										
Subsystem Name: GUI										
Component Name										
Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	Severity (S)	Potential Causes and Mechanisms of Failure	Occurance (O)	Current Design Controls Test	Detection (D)	RPN	Recommended Action	
MATLAB	Program Crash	Inoperable code	5	Insufficient computing power	3	Reliability	1	15	None	
Wrong Input File Type	Code Failure	Program would not run	7	User input error	2	Reliability	1	14	Add a error in GUI displaying wrong input type	
Wrong Button Click	Incorrect Function	Wasted time incorrect order of execution	2	User input error	5	Code Run Test	1	10	Do you want to continue pop up window for important buttons	
Visual Studio	Program Crash	Inoperable code	5	Insufficient computing power	3	Reliability	1	15	None	
Unsaved Results	User execution error	Wasted time	2	User execution error	4	Code Run Test	1	8	Do you want to continue pop up window for important buttons	
Product Name: ATi Aneurysm locator										
System Name: MATLAB Code & GUI										
Subsystem Name: Volume Calculation										
Component Name										
Part # and Functions	Potential Failure Mode	Potential Effect(s) of Failure	Severity (S)	Potential Causes and Mechanisms of Failure	Occurance (O)	Current Design Controls Test	Detection (D)	RPN	Recommended Action	
Incorrect location	Coding Error	Inaccurate Volume	8	R-CNN output	4	Volume Test	3	96	Check R-CNN Accuracy for attempt	
Incorrect formula	Coding Error	Inaccurate Volume	8	Wrong Fomula	1	Volume Test	1	8	Check with known volume	
Inaccurate input	Coding Error	Inaccurate Volume	8	Wrong Bounds	2	Volume Test	1	16	None	
Inaccurate scale	Coding Error	Inaccurate Volume	8	Incorrect or incomplete Meta-data	1	Volume Test	2	16	None	
MATLAB	Program Crash	Inoperable code	5	Insufficient computing power	3	Reliability	1	15	None	