**EMA C2578958 Bashaar Dhoot**

**Development of a Matlab Application for Detecting Brain Tumours**

**Table of Contents**

**Glossary.. 4**

**Problem description.. 6**

**Preparation.. 9**

**Risks and management.. 11**

**Account of related literature.. 12**

**Project work and outcome.. 20**

**First Cycle.. 20**

**Second Cycle.. 30**

**Review of current stage of project work.. 34**

**The Ethical and Legal Issues of Medical Software.. 34**

**Review of project management.. 35**

**Review of personal development.. 35**

**Reference.. 37**

**Appendices.. 39**

**Journal.. 53**

**Glossary**

**Background Subtraction**: Is any technique, which allows an image's foreground to be extracted for further processing (object recognition etc.)

**Edge maximization**: Edges can be defined as boundary between regions in an image. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. Edge detection process reduces the amount of data and filters out useless information, while preserving the necessary structural properties in an image

**Foreground detection**: Is one of the major tasks in the field of computer vision and image processing whose aim is to detect changes in image sequences. Background subtraction is any technique which allows an image's foreground to be extracted for further processing (object recognition etc.).

**Gaussian mixture model**: Gaussian mixture models are a probabilistic model for representing normally distributed subpopulations within an overall population. ... A model making this assumption is an example of a Gaussian mixture model (GMM), though in general a GMM may have more than two components

**Image segmentation**: The process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse

**Kernel**: In image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image

**Point Spread Function**: The point spread function describes the response of an imaging system to a point source or point object. A more general term for the PSF is a system's impulse response, the PSF being the impulse response of a focused optical system.

**Thresholding**: Analyzing images using image thresholding techniques. Image thresholding is a simple, yet effective, way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images.

**UI**: The means by which the user and a computer system interact, in particular the use of input devices and software

**Matlab syntax**

**bwlabel()**: Returns the label matrix L that contains labels for the 8-connected objects found in BW . You optionally can label connected components in a 2-D binary image using a GPU (requires Parallel Computing Toolbox™). ... L = bwlabel( BW , conn ) returns a label matrix, where conn specifies the connectivity.

**im2bw()**: Is a function 'im2bw' which converts an input image into a black&white image. You can provide the function with a level of thresholding otherwise the default value is 0.5

**imidlate()**: Dilates the grayscale, binary, or packed binary image IM , returning the dilated image, IM2 . The argument SE is a structuring element object, or array of structuring element objects, returned by the strel function.

**imshow()**:Display image

**ismember()**: Returns an vector the same length as a containing logical true ( 1 ) where the elements of a are in the set S , and logical false ( 0 ) elsewhere. In set theoretic terms, k is 1 where a S . a and S can be cell arrays of strings.

**regionprops()**: Measure properties of image regions

**strel()**:represents a flat morphological structuring element, which is an essential part of morphological dilation and erosion operations. ... Use the strel function (described below) to create a flat structuring element. You can use flat structuring elements with both binary and grayscale images.

**subplot()**: Divides the current figure into an m -by- n grid and creates axes in the position specified by p . MATLAB® numbers subplot positions by row. The first subplot is the first column of the first row, the second subplot is the second column of the first row, and so on.

**Z = imsubtract(X,Y)**: subtracts each element in array Y from the corresponding element in array X and returns the difference in the corresponding element of the output array Z. If X is an integer array, elements of the output that exceed the range of the integer type are truncated, and fractional values are rounded.

**Matlab Algorithmic functions**

***Background subtraction***

**Z = imsubtract(X,Y)**: subtracts each element in array Y from the corresponding element in array X and returns the difference in the corresponding element of the output array Z. If X is an integer array, elements of the output that exceed the range of the integer type are truncated, and fractional values are rounded.

***Gaussian filter***

B = imgaussfilt(A) filters image A with a 2-D Gaussian smoothing kernel with standard deviation of 0.5, and returns the filtered image in B.

You optionally can perform the filtering using a GPU (requires Parallel Computing Toolbox™).

B = imgaussfilt(A,sigma) filters image A with a 2-D Gaussian smoothing kernel with standard deviation specified by sigma.

B = imgaussfilt(\_\_\_,Name,Value) uses name-value pair arguments to control aspects of the filtering.

***Edge detection***

BW = edge(I) returns a binary image BW containing 1s where the function finds edges in the input image I and 0s elsewhere. By default, edge uses the Sobel edge detection method.

You optionally can find edges using a GPU (requires Parallel Computing Toolbox™).

BW = edge(I,method) detects edges in image I using the edge-detection algorithm specified by method.

BW = edge(I,method,threshold) returns all edges that are stronger than threshold.

BW = edge(I,method,threshold,direction) specifies the orientation of edges to detect. The Sobel and Prewitt methods can detect edges in the vertical direction, horizontal direction, or both. The Roberts method can detect edges at angles of 45° from horizontal, 135° from horizontal, or both. This syntax is valid only when method is 'Sobel', 'Prewitt', or 'Roberts'.

BW = edge(\_\_\_,'nothinning') skips the edge-thinning stage, which can improve performance. This syntax is valid only when method is 'Sobel', 'Prewitt', or 'Roberts'.

BW = edge(I,method,threshold,sigma) specifies sigma, the standard deviation of the filter. This syntax is valid only when method is 'log' or 'Canny'.

BW = edge(I,method,threshold,h) detects edges using the 'zerocross' method with a filter, h, that you specify. This syntax is valid only when method is 'zerocross'.

[BW,threshOut] = edge(\_\_\_) also returns the threshold value.

[BW,threshOut,Gv,Gh] = edge(\_\_\_) also returns the directional gradient magnitudes. For the Sobel and Prewitt methods, Gv and Gh correspond to the vertical and horizontal gradients. For the Roberts methods, Gv and Gh correspond to the gradient at angles of 45° and 135° from horizontal, respectively. This syntax is valid only when method is 'Sobel', 'Prewitt', or 'Roberts'.

**Problem Description**

Detecting brain tumours in the mid to late stages using MRI scans is relatively easy for doctors to find, however detecting the early stages of a tumour is difficult. This is because MRI images are extremely susceptible too noise and environmental interference. These factors make it harder for doctors to locate the tumour and its cause. Cancer Research UK (2014) states “Almost half of people who get cancer are diagnosed late, which makes treatment less likely to succeed and reduces their chances of survival”. Finding a tumour in the early stages increases the chances of successful treatment in an individual. Due to the fact that hospital staff can take prompt action when the individual is diagnosed. I propose to create a system using Matlab to detect a tumour in the early stages, using MRI images. This would be done by converting the image to grey scale, applying filters to remove noise and interference, applying image processing steps and unique algorithms and finally image segmentation is added to detect the tumour. All this is done using Matlab. At the time of writing this EMA I have gathered a considerable amount of sources, literature and work logs, which I have added into the appendix. This has given me enough knowledge to confidently create and develop the piece of software outlined. The solution would be an application where a MRI image is inputted and the output would be another image highlighting the affected area. This can be achieved with the proposed system using image segmentation processes and many image-filtering techniques for accurate detection. I would like to deliver this software using Matlab, which will have these key features:

• User must input MRI images into a user interface

• Image is converted into grey scale image for further processing

• Noise and other environmental interference are removed from image

• Image filtering techniques are used

• Image pre-processing steps are applied on image

• Tumour edges are detected and shown on a user interface

What I aim to deliver as a project output is software that produces what I have stated and an in-depth report of the process I have taken including documentation with an evaluation. In-depth knowledge about image processing techniques, algorithms and the Matlab software is needed before starting the iterative cycle to create a fully functioning application. Using the iterative cycle I would start with evaluating myself at the end of each cycle (see Appendix A). I would then need to start evaluating my code by checking if it is working as intended. To do this I would first review the situation, I would then gather and analyse what I have achieved, this would lead me to implement it and finally when I am ready show my findings in the form of TMAs and the final EMA. Appendix ‘A’ shows the cycle I will use until I hand in my EMA.

The state chart below shows what the proposed application should take.

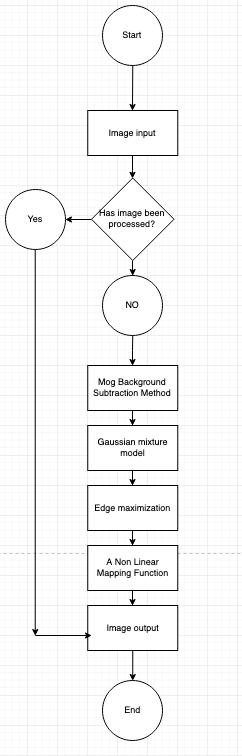


fig. 1

This application will be written and fully realised in the Matlab program. Using the existing algorithms and image processing tools in Matlab will help achieve this. The process is visualised in ‘fig. 1’, while I describe the steps stated in my problem description below.

A. Tumour Image Database: The ‘Cancer Imaging Archive’ is a database, which contains a service that de-identifies and hosts a large archive of medical images of cancer accessible for public download. One of these images is taken and then is subject to tumour detection.

B. Image Pre-Processing: This image is of low quality and is not particularly good for analyses. This is because the image may contain distortions, speckle noise and the contrast may be low. This where locating where exactly the tumour is vital in aiding the medical team to help in the patients recovery.

1. Image Restoration

The purpose of this stage is to take a corrupt or noisy image and ‘cleaning’ it to become the image it once was. Reversing the process of the corrupted image by using the ‘Point Spread Function’ preforms it. This will restore the image’s information that was lost in the blurring process.

2. Smoothing and Sharpening

Image smoothing is done to persevere the images quality. It is another way to remove noise, by not removing principle features. Its aim is to remove three main types of noise. These are impulsive, addictive and multiplicative. Multiplicative is the most difficult to remove, as its intensities vary along with signal strength. Speckle noise is one example of this.

Image sharpening can be classified into two different groups. This includes spatial based and frequency based depending on the required image domain. The spatial based domain operates over the pixel, while the frequency based domain operate with the Fourier and wavelet coefficients of the image.

3. Contrast Enhancement:

Contrast enhancement is also known as histogram equalization. This method increases the contrast of the whole or part of the image. This method particularly works when the backgrounds and foregrounds are either bright or dark. The values of the image are transformed by increasing the image intensity values. This is then compared to match the specified histogram, approximately.

Detection Methodology

The “Image pre-processing” and “Image segmentation” stages (shown in fig. 1) can be defined as shown below:

A. Image Pre-processing:

In the “Image pre-processing” stage we have three steps namely

1) MRI Input Image

2) Grey Scale image

3) High pass filter

1) MRI Input Image:

This step is where an individual uses the database and adds the MRI image they wish to process. It has the RGB (Red Green Blue) mixing present in it. These colours are found in most digital images.

2) Grey Scale Image:

This is the next step of the processing stage. There are still the three main colours found in the digital image, which make it difficult to process. The image is then converted to grey scale, which turns it into a black and white image.

The image contains grey in it, however it ranges from black and white. Where 0 represents the whitest and 255 represents the blackest.

3) High Pass Filter:

In this third step the image that has been converted to grey scale will have noise on it. So the image is passed through a high pass filter to remove, sharpen and bright the image. This filter emphasizes the fine details in the image. It uses a different convolution kernel. In image processing a kernel is a small matrix. It is used for blurring, sharpening and edge detection.

B. Image Segmentation:

In this stage the goal is to simplify and change the look of the image to make it more meaningful to extract data. Partitioning an image into multiple segments does it and is mainly done to locate a subject’s boundary. Threshold segmenting is the simplest form of image segmentation. It partitions the image into foreground and background.

**Preparation**

I have created a Gantt chart, which includes dates of each subtask. I have also written a brief description of what will occur in these week/s. I have also added a Gantt chart from previous TMAs and of my work shifts for the month up to the EMA deadline. Please see appendix ‘B’.

This Gantt chart below is the current one I am working with to keep me on track.

Gantt chart for subtasks

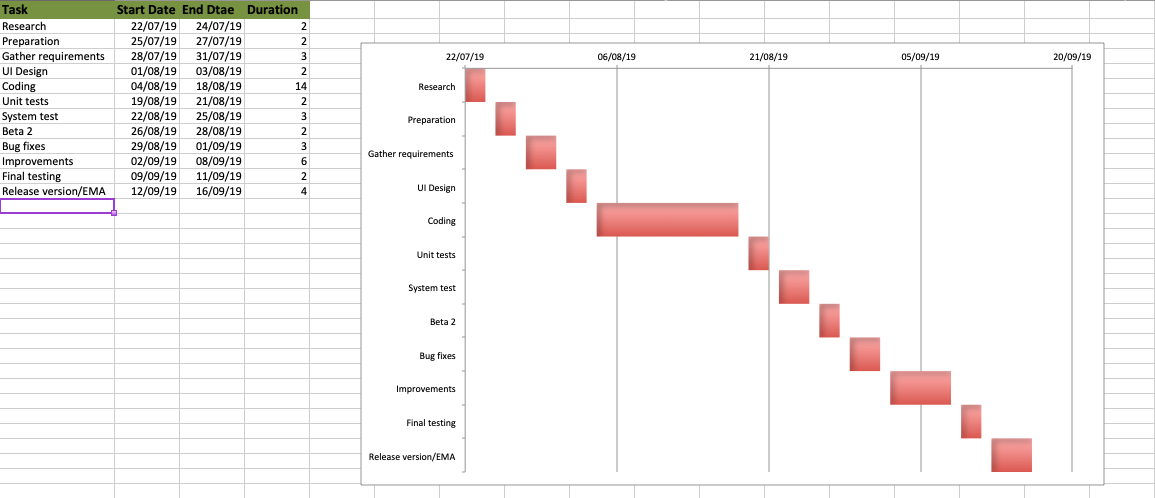


fig .2

Subtasks description

Research:

• Look into connecting script code to individual buttons on the UI

• Look into converting algorithmic function to Matlab syntax (pervious methods out-dated.)

Preparations:

• Book study room in Swiss cottage library

• Talk to manager for days off during the upcoming months. (Suggest 2 days off a week or shorter shifts.)

Gather requirements:

• Show User UI

• Ask questions from the previous iteration

• Note down advice

UI:

• Apply changes from conversation with user

• Add more image panels for different algorithmic functions

• Add login UI

Coding:

• Add algorithmic functions that where promised at requirement. (i.e. subtraction method, Gaussian mixture, Edge maximization)

• Code each Image panel to show each step or outcome

• Code download button

• Code login UI

• Compile code

Unit tests:

• Write unit test that checks that the system only uploads ‘.jpg’ and ‘.png’ images

• Write unit test for the ‘for loop’ function to make sure it is working as desired

Systems test:

• Check system is running as intended

• Write up findings and new questionnaire

• Check if it meets requirements

Beta 2:

• Show user new applications

• Note down required improvements

Bug fixes:

• Correct any bugs found in Beta 2

Improvements:

• Add improvements from the users input

• Add improvements that I feel are needed

Final testing:

• Update Unit test

• Test with different parameters

• Try and break application

Release version:

• Have a working application that detects Brain tumours at early stages

• Write and hand in EMA

**Risks Assessment and Management**

There are two factors that should be considered when thinking about the risk that can affect the project. They are ‘the significance of things that might happen’ and ‘how likely they are to happen’. We can rate these two factors on a scale of ‘high’, ‘medium’ or ‘low’. These can then be categorized into two more events, which are ‘impact’ and ‘likelihood’. For example a ‘high-likelihood’ event means the occurrence of it is high, where a low one means the opposite. A ‘high-impact’ event can threaten a projects success, where a low one can mean delays.

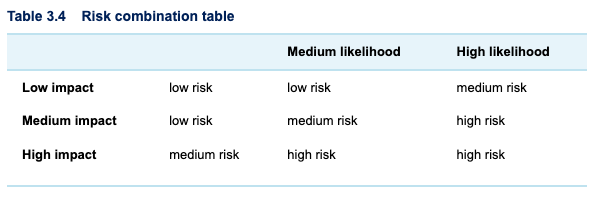


fig . 3

The table above is from the ‘TM470’ module website and weights impact versus likelihood. Below I have examples of what characterises as high, medium and low impact.

Low impact – Not completing further research, creating more UI designs, Unit testing

Medium impact – Improvements on the design, time management, organisational skills

High impact – Adding algorithmic functions, fixing bugs

From the subtasks I have said I will complete this is where I feel they lay on the impact categories. I believe that all the low impact subtasks have a high likelihood of not being completed, while the high impact subtasks have a low likelihood of non-completion. This is because I believe I will concentrate more of my time on them if they are not completed.

Determining which risks to address I believe that low risks are acceptable and only high and medium risks should be addressed. This because these high and medium risks are what I believe will make sure of the projects success. I believe that I have to accept the risks, as this project is following the iterative cycle. In theory there are more cycles in the model for these minor risks to be completed.

**Account of related literature**

I have gathered additional information sources that have aided me with this project. They are also fully cited in the reference section. I have added and summarized them in this part of the document.

1. An Enhanced BE-GGMM-EI Algorithm for Medical Image Denoising. K.C. Patra; Millee Panigrahi; Sushil Kumar Mahapatra; Minu Samantaray. IEEE, Published: 11-11 Jan. 2016

The problem the article is trying to address is the vital need for a brain tumour detection system that does not lose any data that surrounds the edge of the tumour. The solution was using a combination of mathematical filtering methods and a software called Matlab. With these tools the team where capable of solving this problem. The article purposes a method for providing the ‘minimum loss of information in brain tumour MRI images’. This is done by first using a technique called ‘background subtraction’ on the MRI images. This is a technique, which allows an image’s foreground to be extracted. The edge of the tumour is then enhanced using a technique called ‘maximization’. This proposed technique was evaluated by using the process on several MRI scans of individuals with tumours. This article stops short of what I propose, as there are new and improved ‘Mog Background Subtraction Method’ that can be implemented to provide an image with less noise. These use a mixture of models for an improved and adaptive background model. The work is fairly similar, however I will be adding more processes using Matlab. I will be using the same filtering techniques used in their process and finding ways to optimize the procedure.

2. Image Segmentation with Watershed Algorithm. Alexander Mordvintsev; Abid K. RevisionImage. opencv-python-tutroals.readthedocs.io, Published: 2013

This is a website dedicated to teach individuals coding, in this case python image filtering techniques. There is a segment of code here that converts images into grey scale, however it takes it a step further by eliminating the noise around the image. The solution to eliminating the noise was using image segmentation with an algorithm called ‘Watershed’. This was not evaluated as, it was a teaching website, however I have used the code and it works as specified. This code works in Python and encompasses my work, however it is using a different language. I believe Matlab is capable of delivering better outputs than python. This code is similar to the work I am doing. It follows the steps, where it converts an image to grey scale and then uses image segmentation to eliminate noise. It is different because it uses separate algorithm methods and software. Going back to this exercise and playing around with this code I found the ‘Watershed Algorithm’ interesting and useful. This algorithm can also be translated to Matlab and it would be interesting to see the results.

3. BMOG: boosted Gaussian Mixture Model with controlled complexity for background subtraction. Isabel MartinsEmail; Pedro Carvalho; Luís Corte-Real; José Luis Alba-Castro. August 2018, Volume 21, Issue 3, pp 641–654

The problem the authors are trying to address here is developing methods for unsupervised segmentation of moving objects. The authors’ purpose using ‘BMOG’ can boost performance of other methods. The proposed algorithm was assed by a set of extensive experiments conducted on different videos. They where grouped into 11 categories like bad weather and low frame rate, among others. Pervious methods like ‘MOG’ were also assessed using these categories. The ‘BMOG’ preformed better in all outcomes except pan-tilt-zoom videos. Here ‘MOG’ preformed slightly better. This article encompasses the work I am doing and also takes it a step further by making the ‘MOG’ much more robust. The methods used are similar, however they are used on video instead on still images. This method can also be translated in Matlab and is worth looking into.

4. MathWorks Segmentation methods in image processing and analysis. , uk.mathworks.com, viewed: 15/02/2019

There are four types of image segmentation to learn from in this site. These include Otsu’s method, K-means clustering, watershed segmentation and texture filters. Each method divides images into different parts, however each one succeeds better depending on the images. These methods are with in the scope of my project, as they show different ways of segmenting images. These are only a few that can be used within Matlab. I plan to use all these methods and try and test each one weighing their strengths and weakness. What is similar in these methods is that they all divide images, however they go about in different ways.

5. Gaussian mixture model based segmentation methods for brain MRI images. M. A. Balafar. Springer Science and Business Media B.V. 2012, Published online: 6 March 2012

The technology being addressed here is ‘Gaussian Mixture Model’. This is the base segmentation algorithms used in most MRI images and is crucial for correct diagnosis. The solution here was using different ‘Gaussian Mixture Models’ and comparing each one to each other. It was evaluated by providing datasets and similarity indexes for researches to use and work on. The scope is with-in my project because the ‘Gaussian Mixture Model’ is the base model used in MRI images to eliminate noise. This relates to my work because the ‘Gaussian Mixture Model’ is used in most processes and I believe that I should use it as a base model as well.

6. A novel modelling approach of aluminium foam based on MATLAB image processing. Xiaolei Zhu, Shigang Ai, Daining Fang, Bin Liu, Xiaofeng Lu, Computational Materials Science, Volume 82, 1 February 2014:

This paper discusses the outstanding characteristics that lightweight metal foams have and there use in a wide range of industries. Nevertheless their are several shortcomings in the manufacturing of them. These included uneven pore structure and poor reproducibility. Solving this required Matlab imaging process techniques and reconstructing the images in to a geometrical mesh model. This was then evaluated by calculating the compression performance of the foam base. This calculation can show the mechanical behaviour of the foam and the results show that methods used in the paper are applicable. The scope of this paper is to find ways to better maintain the structural integrity of these metallic foams. In some ways it encompasses the work I am doing, as it uses Matlab Software and Image processing techniques. Similarities include the image processing techniques used in Matlab, while the differences are the non-organic materials the techniques are used on. The specific contribution I would like to focus on is there use of image segmentation as I feel it will aid in my project.

7. Image Segmentation, Crucial Tasks In Image Processing ‐ And Computer Vision, Low‐Level Image Processing And Image Analysis Transition. Marques, Oge. Image Processing, Chapter 15, p.365-386:

This is a book that I found using the online OU library resource. It focuses on image and video processing. Both are interesting topics, but I mainly focused on reading part one, which was the image processing section. The book’s motive is to show how extracting data from images and interpreting them has been major driving factor in the development of image processing software and techniques. The outcome of the book is to give the reader a fair amount of knowledge about the uses and techniques needed in image processing. Towards the end to part 1 there were examples of different image processing techniques used in Matlab. The scope of this book is integral to the work I am doing, as the majority of the techniques are useful to my work. The main difference is that the book goes further with its knowledge about image processing, with a second part focusing on video processing. The book also shows specific techniques used in Matlab using mathematical equations converted into Matlab syntax.

8. An Optimal Partial Differential Equations-based Stopping Criterion for Medical Image Denoising. Maryam Khanian, Awat Feizi, Ali Davari. J Med Signals Sens. 2014 Jan-Mar; 4(1): 72–83:

This medical journal is aimed at addressing the need for quality medical imaging and how it is necessary in aiding individuals in a quick recovery. The solution proposed by the journal, is using partial differential equations-based models, such as denoising, multiscale image analysis, and edge detection. A technique called Perona-Malik Model is heavily used in all the medical images. These images are then evaluated using this model compared to pervious models. The outcome shows that a mixture of pervious models and the Perona-Malik Model produce better results. The scope of this journal is similar to my project work as, it is concerned with medical imaging. It also uses similar image processing model that will aid in my work, however it relates to every aspect of medical imaging. My aim is to focus on tumour detection imaging. I believe using the Perona-Malik Model with other models in my work can help denoise the images I am using.

9. Background Subtraction: Theory and Practice. Ahmed Elgammal. Morgan & Claypool, Synthesis Lectures on Computer Vision, Page 8:

The technology addressed by the author is background subtraction used to highlight moving objects in video imaging. The solution was to use background subtraction models mixed with Gaussian and non-parametric models to achieve this. It then evaluates different motion subtraction methods and highlights the advantages and disadvantages of each one. The scope of these methods and ideas are beyond my project work, as it deals with moving images. Nevertheless ideas like image segmentation and different image processing models are core foundations to my work.

10. MATLAB WORKBOOK. Eric Darve, Hung Le. CME 102 Matlab Workbook 2008-2009:

This Matlab workbook starts by showing basic syntax and ends with complex scientific equations. Its aim is to teach and test you on the Matlab software. The purpose of this workbook was to give me a ‘crash course’ in the Matlab software. There was a solution page at the end of the workbook, where I would check my work against the solutions. These exercises might not be used when I start developing my application, however I have not used Matlab so I thought it was essential to start with the basics. This will make the development stage much more manageable.

11. Getting Started with Image Processing Toolbox. MathWorks Team. MathWorks.INC. 1994-2019:

The MathWoks website is the official website for the Matlab software. Here a user can get introductory knowledge of all facets of the Matlab toolbox. In this case I used the site to get familiar with the image processing toolbox. This ranged from basic import and export of images to deep learning for image processing. The latter was very interesting, however not useful to this project. Useful tutorials were the image filtering, enchantment and image segmentation sections. These part of the website is integral to the project, as these techniques will form the base of the project.

12. Why software engineering courses should include ethics coverage. Arvind Narayanan, Shannon Vallor. Communications of the ACM. Volume 57 Issue 3, March 2014 Pages 23-25:

Feedback from the previous TMA my tutor pointed I did not have legal and ethical section that addresses the application I am creating. This led me to find this article, which I believe relates to the work I am doing. The article addresses the issues of why teaching ethics to software engineering student is vital for the development of software, for example the application I am developing will be intended for human use. So issues like data inaccuracies, privacy and confidentiality need to be addressed. The goal of the article is to get students in the habit to write code to serve the public good. The scope of the article encompasses my work, as I will have to write and update the issues I am and will face concerning the ethical and legal issues of it.

13. Create a Simple App Using GUIDE. MathWorks Team. MathWorks.INC. 1994-2019:

When creating the table (see Appendix C) of the tasks that needs to be accomplished. I realised I needed to also create a GUI, so users can interact with the application. This led me back to the MathWorks website. There is a section titled ‘Create a Simple App Using GUIDE’, which takes you on a step by step tutorial in creating a UI. By following these steps I am fairly confident that I will create suitable GUI for users.

14. Testing Frameworks. MathWorks Team. MathWorks.INC. 1994-2019:

Unit and framework testing is integral in software development. This is so a quality product is produced free from bugs and data inaccuracies. It is extremely vital when developing software for use in the medical industry, as an individual’s health is decided by its accuracy. Lucky the Mathworks website also has literature, which is a step-by-step guide of how to create these tests.

15. Programing Pearls. Jon Bentley. 2nd ed. New Jersey: Pearson Education. 2018:

In my spare time I like to learn programing languages as a hobby. I am currently learning how to program in C. Programing pearls was recommended to me by friend. The book talks about common problems software developers face when programming and how to get around them. This book is not in the scope of this project, as it is concerned with C and the C ++ language. Nevertheless I believe surrounding myself with information sources like this will get me in the software development mind set.

16. Ultra-low field MRI food inspection system prototype. Kawagoe, Satoshi; Toyota, Hirotomo ; Hatta, Junichi ; Ariyoshi, Seiichiro ; Tanaka, Saburo. KB+ JISC Collections Elsevier ScienceDirect Freedom Collection 2017-2021

Available from Elsevier1995 volume: 241 issue: 1

In this article this team have developed a way for inspecting food, which contain high levels of water. The solution that was developed was to use ultra-low field magnetic resonance imaging system (ULF and MRI respectively). These two systems would use “high-temperature superconducting quantum interference devices”. These techniques would produce two-dimensional MRI images that are reconstructed from a grid using raw data. The way this technique was evaluated was by preforming a test, where a disc sample is encapsulated with 7.7ml of water. It is 35mm in diameter and 8mm in thickness. Two 2D MRI images are produced. The first in a semi magnetic shielded room and the other in a compact magnetic shielded box. The latter removed a vast amount of noise and produced a much clearer image than the first image. The scope of this work is impressive, however does not particularly align with my work. Nevertheless it has given me some understanding of MRI images and how they can be used in other applications. The main similarity is that they are using MRI images to solve the problem they have stated. The difference is that they are creating MRI images, where I will be processing MRI images to solve the issues I have stated.

17. Implementation of Clustering Algorithms for real datasets in Medical Diagnostics using MATLAB B. Venkataramana ; L. Padmasree ; M. Srinivasa Rao ; G. Ganesan ; K. Rama Krishna. Journal of Soft Computing and Applications, 01 March 2017, Vol.2017(1), pp.53-66

As the numbers of patients are increasing exponentially, there sample size also increases. A disease in the medical field requires different samples given by diagnosis and analysed by doctors. This team proposes using clustering algorithms, where they would be able to classify each of the samples. There are varying algorithms using data clustering techniques that have their pros and cons. Several of these techniques where used by this team and evaluated side by side. They concluded FPCM method had the highest percentage of correctness. FPCM stands for Fuzzy Possibilistic c-Means algorithms. Out of all the methods this one came out on top. Reading this work I could not help but see similarities to my work. The scope of this work is creating a system of classifying a larger amount of samples for use in the medical field. It seems that this team has delivered three different methods of preforming this. The main difference is that they are using Matlab for data analysis, where I am using it for image processing. However there are several similarities, which include converting algorithms to Matlab syntax and creating a functional application for the medical field.

18. Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms. Ozdemir, Caner. Hoboken: John Wiley & Sons, Incorporated

This book contains description and ways of performing Synthetic Aperture Radar Imagery. It only deals with signal processing techniques used within Matlab. I was told by a friend that I might this book useful, however it is nowhere within the scope of this project. However I have found it useful the way the author describes and shows you the steps in converting Mathematical equations into Matlab code.

19. Image processing algorithms and techniques III. Sullivan, J. R. – Dawson, B – Rabbani, M. 10-13 February 1992, San Jose, California /’, in. Bellingham, WA: The Society

While on the OU library I was lucky to find an archive of Image processing Algorithms and techniques, which contain 54 papers all edited by the authors above. Full papers are available, however I found reading the summarised notes of each paper and choosing what felt suitable was less time consuming. 18 papers where about Image segmentation and Modelling. This technique is integral to the application I have proposed. As of writing I have yet to complete all 18 papers and I plan to use their techniques for later iterations of my project cycle. I also believe that the other paper can be useful further on in development.

20. Automated Software Testing for Matlab. Eddins, S. L. Computing in Science & Engineering. IEEE, 11(6), pp. 48–55

The problem this paper is trying to address is that testing each module in your code can be time consuming, however writing code that tests each module, so that it produces the desired variables can save us this time. Unit testing is a way of solving this and essential to any programing language and development. I have created several Unit tests in the Python programing language, however I have not in the Matlab language. This paper introduced me to the Matlab xUnit automated testing framework. It has helped me write code, which can help create cases that checks the determined value or even compare that value. The scope of this work is all about testing different types of Matlab programs and testing them using predetermined values. This encompasses the work I am doing and has helped the project work and produce a stable application.

While reading through the comments regarding “The Ethical and Legal Issues of Medical Software” section from my last TMA I felt that needed to improve on this. This has led me to sources (6 and 7).

21. Innovation in Medical Technology : Ethical Issues and Challenges. Eaton, M. L. and Kennedy, D. (2007). Baltimore: Johns Hopkins University Press, pp. 1–155

In this study the author talks about the ethical, legal and social problems that arise with new medical technologies. The author uses four examples of unregulated technologies including off label use of drugs, innovative surgery, assisted reproduction and neuroimaging. The author concludes that when deploying new technologies oversight, disclosure, educating and learning are important consideration. She continues on saying that there must be core concerns in addressing over sight, patient consent and information sharing and how there is a lack of consensus in the medical profession. She finally ends with haw the goal of medical technology is to advance it with the interest of science, medical practice, patients and society. The scope of this study was to look at unregulated studies focusing on four in particular. I believe this encompasses my work and how I should approach it. One particular quote comes to mind. Eaton (2007) states, “… is to slow the deployment of technology until it has been thoroughly examined for risk.” I feel there are plenty of similarities in this work for any one producing an application in the medical field, as it is only concerned with the ethical issues and challenges. This will help me create a more ethical application and make me think about the issues while preforming my subtasks.

22. The Ethics of Medical Progress. Reitsma, Angelique M (2007). Nature Medicine. Nature Publishing Group, 13(6), p. 665.

This is a short article, which is similar in ideology to the previous study. It tries to address and debate the boundaries between research and practice in the medical field. It looks into unproven drugs, assisted reproduction and brain imaging among others. This article does not give a definitive solution, however it lets the reader come to ones own. For example the author concludes neuroimaging leads access to brain imaging information and Reitsma A.M (2007) states, “will the data be used to define normality or personal identity, and is it stigmatizing?” This article encompasses the ethical issues and challenges of the project at hand. It does not come to a conclusion, as I believe that these issues are very complex and that you will find differing opinions. Nevertheless it does help the reader like my self to think about how complex the issue is.

These last two literature resources where kindly recommended to me by my tutor. These have been some of the most useful resources I have read as they encompass my work and are similar to the project I am conducting.

23. Brain Tumour Detection In Medical Imaging Using Matlab. Pankaj Kr. Saini, Mohinder Singh. International Research Journal of Engineering and Technology, Volume: 02 Issue: 02 | May-2015.

The issue this article is addressing is image segmentation and its use in the medical field for MRI scans. The main use is to extract details from an image where a ‘merger’ or ‘split’ occurs. Using Matlab the team has proposed using a data classification system to detect a brain tumour in an MRI image. The system was carried out on multiple Matlab simulations, with much success when viewing the images in the article. There is a clear gradient when viewing both images. The scope of the work is clear and to the point showing the steps they have taken and is what I aspire to. This how ever stops short from my work, as I would like to add several algorithms before image segmentation. The similarities are clear here, as they are using The Matlab software and the image segmentation process. The way this can help me in my work is that I can learn from them and hopefully build upon the work.

24. Brain Tumor Detection based on Multi-parameter MRI Image Analysis. Rajeev Ratan, Sanjay Sharma, S. K. Sharma. CGST-GVIP Journal, ISSN 1687-398X, Volume (9), Issue (III), June 2009.

The problem this paper is addressing is image segmentation, how to improve it, its pros and its cons. The outcome that was proposed was using the Matlab application to perform different types of segmentation. This application was given different variables to test this. This included images with images with different intensities between the tumour and the brain matter. It was concluded that watershed segmentation was the best. The scope of this project is in line with mine and it does encompass it. It is similar that it uses the Matlab application and use algorithmic functions in its system, however I wish to use more functions.

**Project work and outcome**

I have added the subtasks of the previous iterative cycle and the subtasks that I have completed in this cycle. This is to show the software development process fully.

**First cycle**

I have broken this part down into ten sections, which consists of the ten subtasks I have stated in TMA02. I have added my previous Gantt chart with the list of subtasks for reference (see Appendix B). I have also added a weekly journal entry, which I have summarised into several sections (see Journal)

**Gathering Requirements**

At this stage of the project I felt I needed a second hand opinion form someone in the medical field. I arranged a meeting with a friend (Harry Stewart-Smith) to show him my state chart, showing the steps that my application would take (see Appendix C). I felt this was a good way to obtain requirements. I have then (with the help of Harry) created a table of functional requirements and non-functional requirements. We did this by looking at the systems requirements, as the state chart slightly confused him.

Systems requirements

• User must input MRI images into a GUI

• Image is converted into grey scale image for further processing

• Noise and other environmental interference are removed from image

• Image filtering techniques are used.

• Image pre-processing steps are applied on image

• Tumour edges are detected and shown on GUI

Functional Requirements

|  |  |
| --- | --- |
| **User** | Doctors/Nurses/Radiographers |
| **Show or provide individual patient’s history** | Displays details for users |
| **Show or provide patient’s treatment procedure** | Displays details for users |
| **Hardware** | Dell computers/ Mac Desktops |
| **Software** | Windows 7 and higher |
| **Communications** | Download/e-mail .jpg file |

Non-Functional Requirements

|  |  |
| --- | --- |
| **Performance** | Image processed in reasonable time |
| **Safety** | Image processed is accurate |
| **Look and feel requirements** | System is astatically pleasing |
| **Ease of use and Navigation** | Easily understood and can be used |
| **Security** | System provides only legitimate user  Exceptions or error handling routines |
| **Quality** | Image processed is better than previous system |

From here I needed to added more functions to the system. Firstly a login UI to make the system secure and a way to download, so users can email the images. I also updated the state chart, so it would be easy for Harry to understand (see Appendix D.)

**UI Design**

I decided to create two UIs one for logging in and the second for performing all the necessary image processing. In this main body of work I will add the UIs I wish to use, while in the appendix I will add some of the rejected ones (see Appendix E). For the login UI I have only produced one design, as it does not need dwelling on.

Finalised UI – Log in GUI

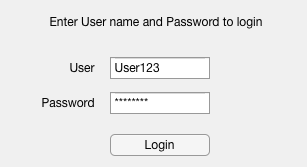


fig. 4

Finalised UI – Image processing UI

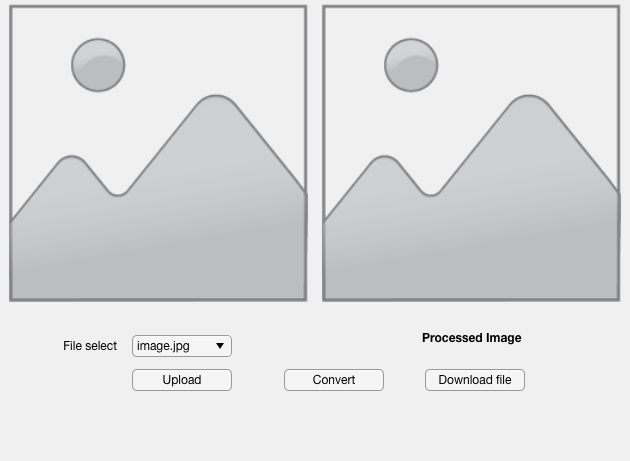


fig. 5 This is how the application will look and has not been coded yet

These are the two UIs that Harry and me have finalised on. Out of all the designs I have sent him he preferred this the most. Now that UI has been designed, each element can be coded through Matlab to preform the desired functions. These designs where done by using the “design app” in Matlab, which too their merit is a whole application on its own. Using this “app” you can drag and drop the desired buttons, image placeholders and text, amongst other objects.

You are also able to look into the applications source code:

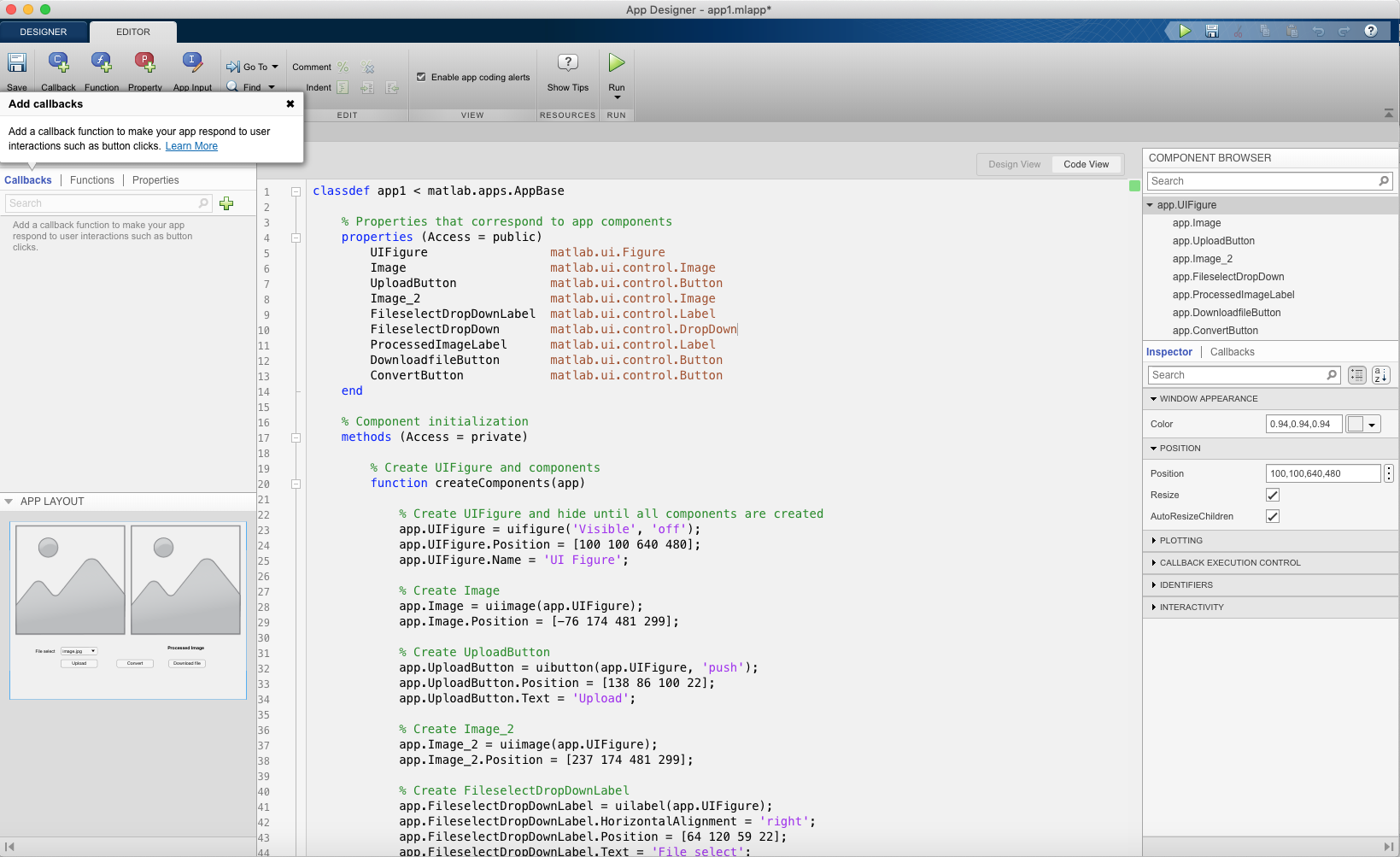


fig. 6

**Coding**

I started by writing a script for the steps I wanted the application to take. This was a straightforward code (or so I thought), where it found the tumour and highlighted a boundary around it. I wanted to the add the algorithmic functions to the script (Mog, Gmm), however it was taking a longer then I thought just to create this current script. Nevertheless, now I feel I have the ‘backbone’ structure to add upon in the next iteration of this project life cycle. I have added screenshots of the code in the appendix (see appendix F.) However, I will explain step by step what the script does. I have also added the two MRI images I used while creating the script. One image is clear and the other contains noise (see Appendix G)

I first uploaded the image using the ‘imread()’ function. The image was then turned into grey scale using the ‘im2bw()’ and setting the threshold to ‘0.7’. Anything below that threshold would be white and anything above would be black. We then label each of the black and white parts using the ‘bwlabel()’ functions. This is because all I want is the white area, which contains the tumour.

Code line 1 to line 4

*mri = imread(‘mri1.jpg’);*

*grey = im2bw(mri, 0.7);*

*label = bwlabel(grey);*

The solidity of different part of the brain varies, however the solidity of the tumour is virtually constant. This makes highlighting it in different images simple. The function ‘riegionprops()’ measures the labels in the image this includes the area and the solidity. The function is set to stats.

Code line 6

*stats = regionprops(label, ‘Solidity’, ‘Area’);*

I then set the solidity and area of the label to their own corresponding variables.

Code line 8 to line 9

*density = (stats.Solidity);*

*area = (stats.Area);*

I then want the program to recognise a high dense area. I do this by establishing anything above ‘0.5’(50%) is high density. This would also establish that the tumour is in the late stage. We then set the maximum area to the ‘denseArea’, which is anything above ‘0.5’. I then use the ‘find()’ to compare the two variables ‘area’ and ‘maxArea’ to find the tumour and setting it to another variable ‘tumourLabel’. This is done by using the Boolean function ‘==’, which returns a value of True or False. ‘Ismember()’ is a function that detects members of a set. This will help me create a set of arrays where I can add the variable label and tumour label and set them to ‘tumour’.

Code line 11 to line 14

*denseArea = density > 0.5;*

*maxArea = max(area(denseArea));*

*tumourLabel = find(area == max\_area);*

*tumour = ismember(label, tumourLabel);*

I then performed a common morphological operation, which checks that the tumour that we found and highlighted all has the same value. In this case a white colour. This is done by using the ‘strel()’ function and then setting the tumour variable to it.

Code line 16 to 17

*se = strel(‘square’, 5);*

*tumour = imidlate(tumour, se);*

From here I start plotting the image we have processed. This includes the brain and the tumour alone. The functions used here are ‘sublot()’ and ‘imshow()’ and ‘title()’. The first two images are the original MRI and the highlighted tumour alone.

Code line 21 to line 27

*subplot(1,3,1)*

*imshow(mri, [])*

*title(‘MRI Scan’)*

*subplot(1,3,2)*

*imshow(tumor, [])*

*title(‘Segmented Area’)*

I now want to create a third image, which draws a boundary around the tumour. This is done with the function ‘bwboundries()’ inside the brackets I specified that I want to draw around the tumour and I can also specify that I do not want holes. I first must show the original image. From here I used a ‘for loop’ to draw around the tumour. This code loops until it creates the desired effect, in this case drawing a boundary around the tumour. It starts by setting the index ‘i’ to the length of the largest array dimension. It then cycles through plotting around it using the variables ‘y’ and ‘linewidth’ and setting that to a thickness of ‘1.45’ once it has reached where it had started, the ‘for loop’ ends.

Code line 29 to line 37

*[B,L] = bwboundries(tumour, ‘noholes’);*

*autoplot(1,2,3)*

*imshow(mri,[])*

*hold on*

*for i = 1:length(B)*

*plot(B(i)(:,2), B(i)(:,1), ‘y’, ‘linewidth’, 1.45*

*end*

*title(‘Image Processed’)*

*hold off*

Here are the results on both MRI images:

Image with noise

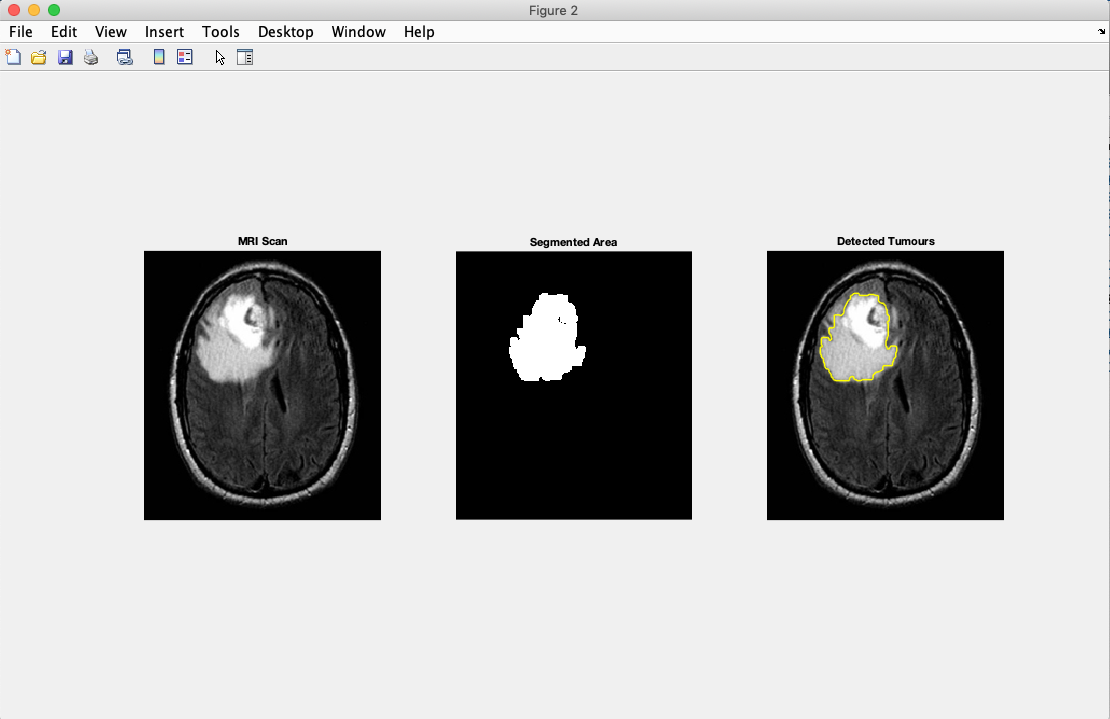


fig. 7

Image without noise

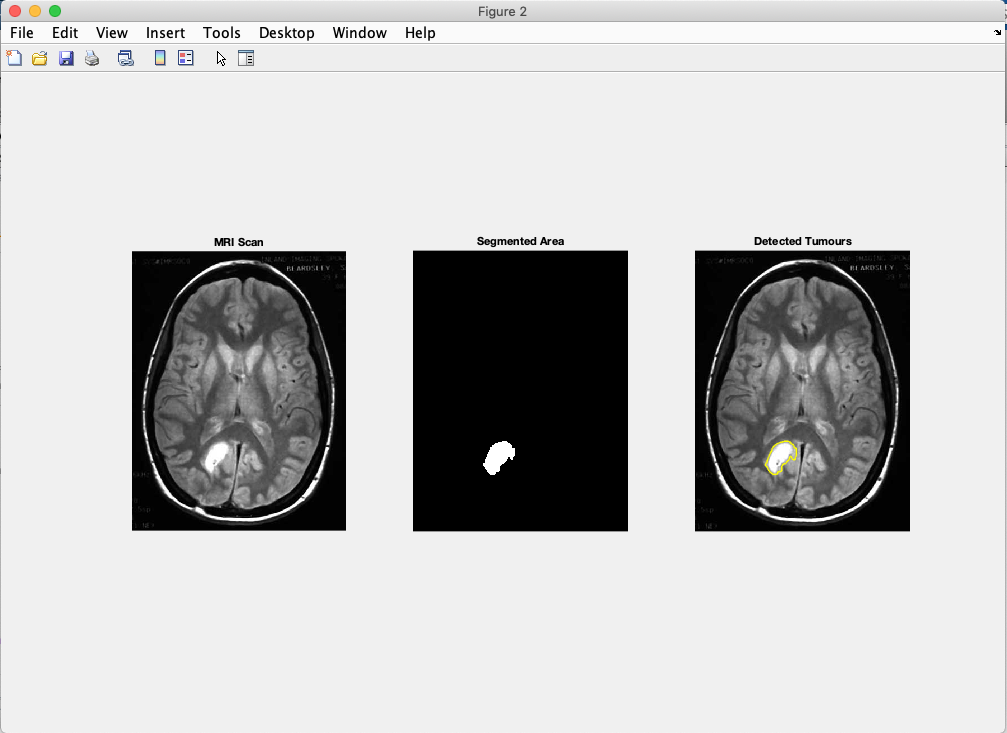


fig. 8

**Unit Tests**

The issue with running a systems test at the moments is some of the features that require this sort of test are still not done. This includes the upload, image processing and download buttons. In the weeks where I planned to fixes bugs I have decided to resolve these issues. Nevertheless I have written up unit test examples of how to link a test script to the main script.

For example I would have a ‘my\_tests.m’ file containing three test functions:

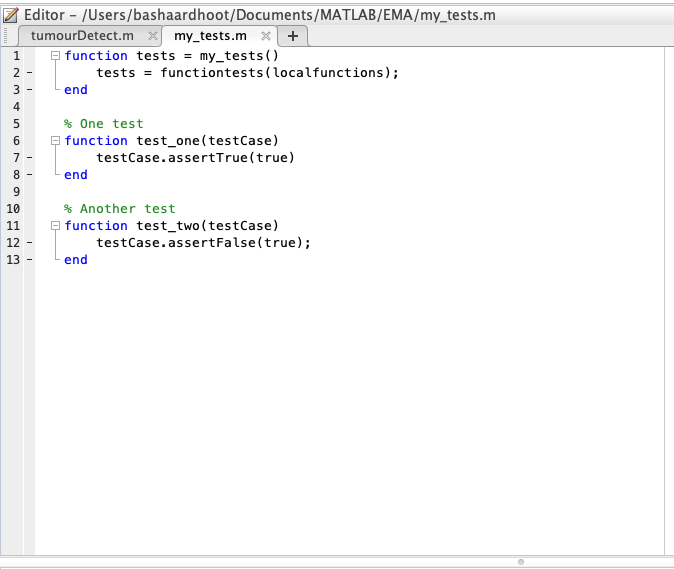


fig. 9

Each function would test one aspect of the code and then on the main script I would run the code with the line:

*runtests(‘my\_tests.m’)*

**Systems test**

While testing the two MRI images and running them through the script obvious requirements and functions are missing. These include UI experience for the user and most importantly algorithmic functions, which would help reduce noise and detect a tumour in the early stages. The system can detect a tumour, which I have shown figure 5 and figure 6. However, in figure 5 the system cannot distinguish between the tumour and noise. At this point and time the system does not resemble what was stated at the requirement stage.

**Beta 1**

For the moment I have produced a script that performs the image processing necessary to find a tumour in the early stages. This is not what was promised to the user, however it can be built upon in later iterations. With the aid of my computer and the user (Harry Stewart-smith) I was able to discuss what was achieved and what we need to accomplish to achieve the software that was stated in the requirement stage. I have asked him five questions, which I aim to ask in the next iteration. These questions where:

* What do you think of the application at this stage?
* What is missing from the application that needs to be added?
* How useful is the application at the present time?
* Would this be useable in your work environment?
* What improvements would you suggest?

While talking to the user, he gave his opinion on the application and answered my question as best as he could. He told me what I have achieved was impressive, however not what was promised. He added the their where important features missing from the application. This included an interface, a way to upload an image and a way to download one. This led him to say that at the present time the application was not useful. He also added that if he where to use it in a work environment it would be for teaching purposes and not for practical medical use. He finally added the major improvement he would like to see in later iteration would not only be the features he had already stated, but a way to highlight the tumour in the early stages.

**Bug Fixes**

At this stage I decided to sort out the UI so the code can be ready to build upon on the next iterations. In the app designer and editor suite you are able to access the source code for each of the buttons and (axes) image display. This is simply done by right clicking on the object and picking the callback selection. For the ‘image processing’ button I just added the original script so the image from the first panel get processed and displayed in the second panel. The main problem I had was solving the upload button. I was not sure how to call the file search prompt. Nevertheless this was solved. Below is the code for that:

*[File\_Name,Path] = uigetfile({'\*.jpg';'\*.png\*';},'Image Selector');*

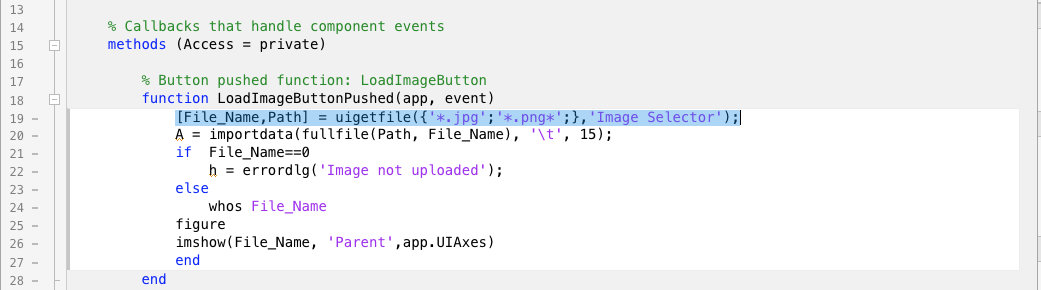


fig. 10

Figure 8 shows the coded ‘Load Image’ button. Line 19 open the file upload prompt so the user can choose his or her file. Line 21 is where the ‘if’ statement starts. The parameter starts by asking if there is a file. ‘File\_Name == 0’ means that there is no file, which would return with an error ‘Image not uploaded’. The ‘else’ statement is when the user finds the file the wanted. This would return with the file uploaded to the ‘UIAxes’.

Here are the results from the UI:

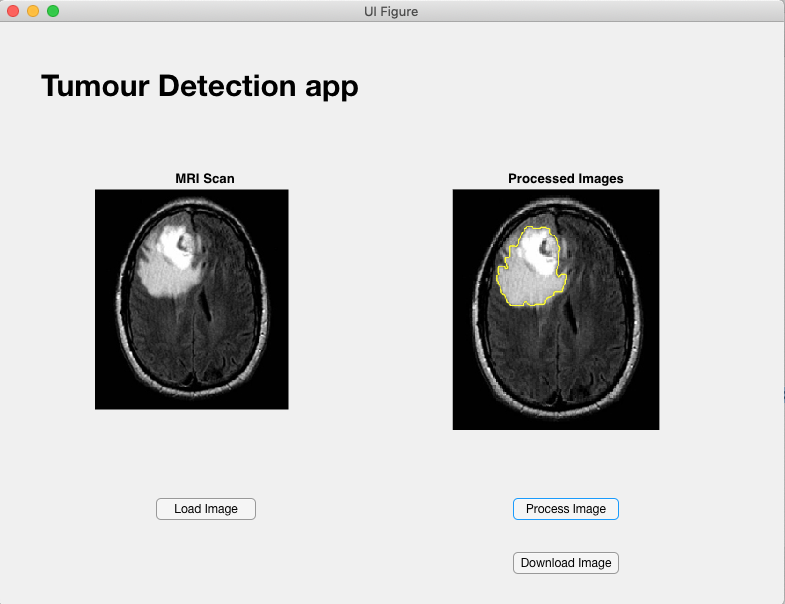


fig. 11

**Improvements**

At this stage of the project I only managed to improve one aspect to make the system more streamed lined. Purely because of time management issues and while creating the UI more bugs started appearing.

The improvement I made was to let the user only upload ‘.jpg’ and ‘.png’ files, as they where the only ones that should be used. In figure 8 line 19 there is a function called ‘uigetfile()’. I found this in the documentation section of the ‘MathWorks’ website. On the site the documentation recommended to add a ‘\*’ symbol between the brackets. However that would let any type of file acceptable. To solve this I simply specified the ‘.jpg’ and ‘.png’ extension in the brackets.

**Final Testing**

In the previous TMA I stated the outcome of this section I would ‘Successfully have an application that was error free’. Unfortunaly at the time of writing this it is false.

List of functions that need to be added in the next iteration:

* Download button needs to be coded
* Algorithmic functions need to be added
* Several more image panels added to the UI to show these algorithmic functions.

List of current bugs:

* Processed image panel resizes when image is processed (see figure 9).
* Blank window pops up when image is uploaded.

**Release version**

I stated here that I would have a successful system by the first iteration. Unfortunately that is not the case. Nevertheless I whole-heartedly believe that this is an excellent state to start the next iteration and deliver the intended system. I have a functioning script that recognises a tumour and a UI that can run it. All that is needed is algorithmic functions to rid the images of noise and detect the edges. Also a bug free UI to make it user friendly.

**Second Cycle**

For this cycle, before the EMA deadline I have chosen and added eleven MRI images to use in this iteration. They include the previous two images that I have processed. I have also processed these new images and added them to the appendix. (Please see Appendix F)

**Research**

While researching online I managed to find how to create the algorithmic function I need to build upon my program.

*Edge detection, 2-D Gaussian filtering and Background subtraction* is what I needed for this project. Using the ‘MathWorks’ I have found how to perform each step.

*Background subtraction* – this is a technique, which allows an images foreground to be extracted. Using the syntax ‘imsubtract’ can do this for us.

An example of the code can look like this:

Read a grayscale image into the workspace.

I = imread('mri.png');

Estimate the background.

background = imopen(I,strel('disk',15));

Subtract the background from the image.

J = imsubtract(I,background);

Display the original image and the processed image.

imshow(I)

*2-D Gaussian filtering* – The purpose of this function is to smooth the image Using the syntax ‘imgaussfult’.

An example of the code can look like this:

Read image to be filtered.

I = imread('mri.jpg');

Filter the image with a Gaussian filter with standard deviation of 2.

Iblur = imgaussfilt(I,2);

Display the original and filtered image in a montage.

montage({I,Iblur})

title('Original Image (Left) Vs. Gaussian Filtered Image (Right)')

*Edge detection* – The purpose of this function is to find the edges of an image. The syntax ‘edge(I)’ can help us do this.

An example of the code can look like this:

Find edges using the Canny method.

BW1 = edge(I,'Canny');

Find edges using the Prewitt method.

BW2 = edge(I,'Prewitt');

Display both results side-by-side.

imshowpair(BW1,BW2,'montage')

**Gathering requirements**

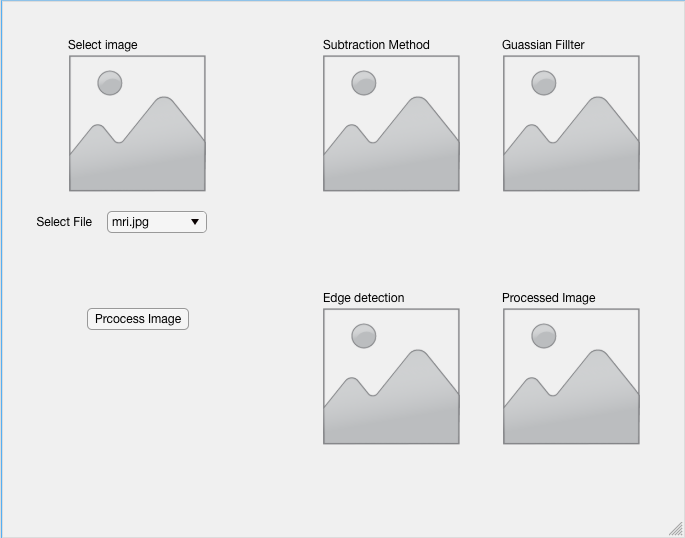
I was lucky to meet up with Harry Stewart-Smith again and show him what I fixed during the first iteration. I also wanted to show him the extra MRI images I added for processing, as they ranged from large tumours to smaller ones. As for the requirements he only asked that the app would show the different stages of the image processing steps.

Questionnaire

* What do you think of the application at this stage?
* What is missing from the application that needs to be added?
* How useful is the application at the present time?
* Would this be useable in your work environment?
* What improvements would you suggest?

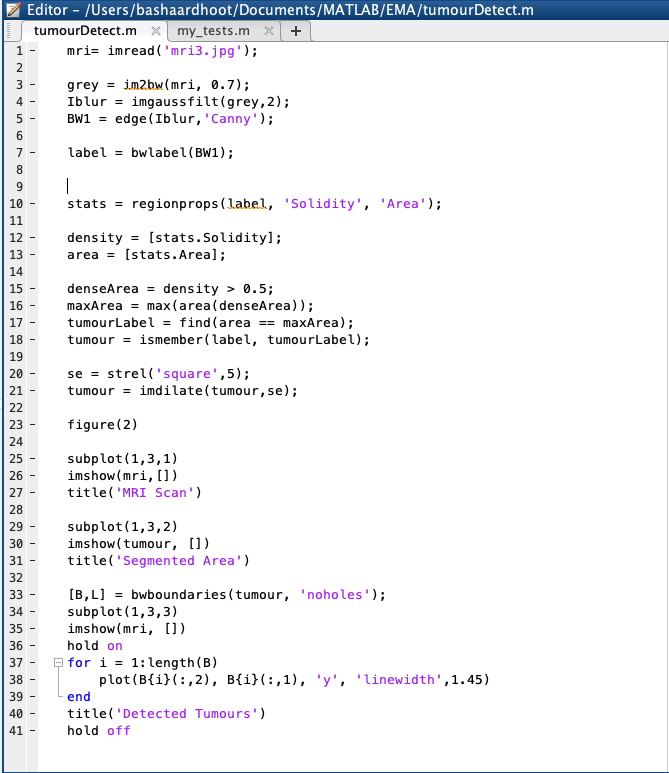
Harry’s response this time was that he was happy at the different stages the applications was able to detect the tumour. What he felt that was missing was a way to show the different steps the application took to reach its goal. He added that the application was useful in its current state as, it was able to detect tumour of different sizes. However the application was not bug free, so it would not be ideal for the work environment.

**New UI**



This is a ‘wireframe’ of what the purposed new UI will look like. Each panel on the right would show how each algorithmic function would affect the original MRI image.

**Coding**

****

At this stage of the project I was having problems incorporating the new algorithmic functions. I was able to use them separately, however incorporating to the code that I had was unsuccessful. I kept on getting error messages. At this stage I moved on to writing up my EMA, as I was not able to get in contact with Harry Stewart-Smith.

**Review of current stage of project work**

**The Ethical and Legal Issues of Medical Software**

I believe there are four main issues that need to be addressed through this projects lifecycle. They are privacy and confidentiality, security breaches, system implementation and data inaccuracies.

1. Privacy and confidentiality – This is all about an individual granting me permission to use their MRI images and that I can agree that their data cannot be uniquely identified.

I believe that this is an issue most if not all individuals view as critical. The MRI images should not show any patient medical information. This includes name, D.O.B, address and patient ID. This information should only be viewable to those who are given permission (i.e. GP’s, nurses, and radiographers). Taking this even further these individuals like nurses might only need to know penitent ID and name, while radiographers might only need to know patient ID. At the time of talking to my user he told me each patient is given an ID and this is usually in the name of the ‘.jpg’ and ‘.png’ file. Patient ID is a set of numbers, which references that individual. All three users would need access to this. I believe most people would not mind if this information is passed around. I believe that a patient ID can be written above the image panel on the UI. This would not compromise Privacy and confidentiality. At the time of writing, my application does not have this feature, but I will look at ways to implement it.

2. Security breaches – This is when sensitive, protected or confidential data stolen or transmitted to unauthorised persons. Most data breaches occur through overexposed unstructured data.

This issue for my application deals with making sure access is only given to individuals that require it. These MRI images are sensitive information and I must make sure that no ordinary individual can access it. I can only secure my system and anything done after (i.e. downloading the image and sending it) is down to that system and the individual. My data structure requires a UI to login in to the program. At the time of writing this I do not have a running login UI, however this will be amended in this upcoming iteration.

3. System implementation – This is the process of defining how to build and store information, ensuring the system meets excellent standards.

This issue I feel does not pertain to my project as much as the other issues. All information that is uploaded and downloaded will be saved on the users computer. I believe to solve this issue my application would just need to wipe the image between each use, as this would not let other medical employees view the images.

4. Data inaccuracies – This process is the most important. It deals with reducing error in healthcare, which would reduce health disparities and improve the public health.

For me this is the most important issue. This is because any data inaccuracies can cause major damage to the patient. For example if the application does not find a tumour when there is one this can cause the patient to believe he or she is fine. People usually believe what medical technology tells them. This is why there is a need for my application to always deliver accurate information. A way of making sure it can do this would be to test multiple MRI images and noting down the results.

**Review of project management**

The lifecycle that has been used in this project is an iterative one. Unfortunately I did not deliver what I had stated at the beginning of this module. Nevertheless the amount of progress that has been achieved from using Matlab has been great. I have only been programing with it since October and being able to produce an application of this kind has been encouraging. I do not believe I would have been able to get this far in the project with out using an iterative life cycle, as it has given me the chance to repeat, practice and refine the work at hand.

However there are several factors that need to be considered when reviewing project management.

These include:

* Is the project currently delivering to schedule?
* Have risks been controlled and mitigated?
* Were issues identified and resolved?
* Were changes properly managed?
* Is the project on track?

At this point I do not believe I have delivered to the schedule I stated in this EMA. I do however believe that the risks have been controlled and mitigated, as there was the possibility that the project would fail. The main issues, which are the algorithmic functions, have been identified, but at this point have not been resolved. Changes have been properly managed, as I have tried my best to update designs and code to the request of the users. I do believe the project is on track if more iterative cycles are taken.

**Review of personal development**

This project has been a great and challenging experience. It has taught me a lot about my capabilities. For example, like how to manage a project by myself. It has taught me how to create timetables and Gantt charts and keeping to them. Taking on this project has also shown me the different ways in which I am able to gather information. Whether it being from a library book or just using the Open University library resource. The latter has been unquestionably helpful, not only for its vast amounts of information, but also for how accessible it is. Most importantly it has shown me that the best way for me to work is practicing and repeating. With out constantly working and testing parameters in Matlab I do not believe I would have reached this far in the project. I do however believe that I need to learn and practice more in the Matlab environment, as it would extend my current knowledge and skill and deliver an even better application.

There are several key skills that I have learnt and improved upon. These include time management, Matlab skills, and image processing knowledge and even some basic MRI and tumour knowledge. Some of these skills I had minor or no knowledge on them at all, however now I feel confident to pursue a project within these fields, like a Matlab project using their image processing toolbox.

I believe I would need more knowledge about image processing, how to incorporate algorithms and knowledge about tumours to take this project to the next level. Finally I would like to comment about how I learn and work most effectively. It has to be with a deadline with certain goals that have to be achieved in between.

Word count [10733]

**Refrences**

*Almost half of cancer patients diagnosed too late, Denis Campbell, health correspondent, Mon 22 Sep 2014, published Mon 22 Sep 2014*

*Balafar, M.A., Gaussian mixture model based segmentation methodsfor brain MRI images, Springer Science+Business Media B.V. 2012, Published online: 6 March 2012*

*Bentley, J. (2018). Programing Pearls. 2nd ed. New Jersey: Pearson Education*

*Cancer Research UK:* [*www.cancerresearchuk.org*](http://www.cancerresearchuk.org)

*Darve, E., Le H., (2009). MATLAB WORKBOOK[online]. CME 102 Matlab Workbook 2008-2009. Available from:* [*https://mc.stanford.edu/cgi-bin/images/e/e3/Darve\_cme102\_matlab.pdf*](https://mc.stanford.edu/cgi-bin/images/e/e3/Darve_cme102_matlab.pdf)

*Eaton, M. L; Kennedy, D. (2007) Innovation in Medical Technology : Ethical Issues and Challenges. Baltimore: Johns Hopkins University Press, pp. 1–155.*

*Eddins, S. L. (2009) ‘Automated Software Testing for Matlab’, Computing in Science & Engineering. IEEE, 11(6), pp. 48–55*

*Elgammal, A. (2014). Background Subtraction: Theory and Practice[online]. Morgan & Claypool, Synthesis Lectures on Computer Vision, Page 8. Available from:* [*https://ieeexplore-ieee-org.libezproxy.open.ac.uk/document/7005380?bknumber=7005380*](https://ieeexplore-ieee-org.libezproxy.open.ac.uk/document/7005380?bknumber=7005380)

*Ilnumerics Team. Ilumerics.net [online]. 2019. Available from:* [*https://ilnumerics.net/computing-engine.html*](https://ilnumerics.net/computing-engine.html)

*Kawagoe, S; Hirotomo, T; Junichi, H; Seiichiro, A; Saburo, T. (2016) ‘Ultra-low field MRI food inspection system prototype’, Physica C: Superconductivity and its applications. Elsevier B.V., 530, pp. 104–108. Available from Elsevier 1995 volume: 241 issue: 1*

*Khanian, M Feizi, A; Davari, A;. (2014). An Optimal Partial Differential Equations-based Stopping Criterion for Medical Image Denoising[online]. J Med Signals Sens. 2014 Jan-Mar; 4: 72–83. Available from:* [*https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3967457/*](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3967457/)

*MartinsEmail, I; Carvalho, P; Corte-Real, L; Alba-Castro, J, BMOG: boosted Gaussian Mixture Model with controlled complexity for background subtraction, August 2018, Volume 21, Issue 3, pp 641–654*

*MathWorks Segmentation methods in image processing and analysis, uk.mathworks.com, viewed: 15/02/2019*

*MathWorks Team. uk.mathworks.com[online]. MathWorks.INC. 1994-2019. Available from:* [*https://uk.mathworks.com/help/images/getting-started-with-image-processing-toolbox.html*](https://uk.mathworks.com/help/images/getting-started-with-image-processing-toolbox.html)

*Mordvinstev, A., ; Abid K. RevisionImage, Segmentation with Watershed Algorithm, opencv-python-tutroals.readthedocs.io, Published: 2013*

*Narayanan, A; Vallor, S; Why software engineering courses should include ethics coverage[online]. Communications of the ACM. Volume 57 Issue 3, March 2014 Pages 23-25. Available from:* [*https://dl-acm-org.libezproxy.open.ac.uk/citation.cfm?doid=2566590.2566966*](https://dl-acm-org.libezproxy.open.ac.uk/citation.cfm?doid=2566590.2566966)

*Oge, M. (2011). Image Segmentation, Crucial Tasks In Image Processing ‐ And Computer Vision, Low‐Level Image Processing And Image Analysis Transition[online]. Practical Image and Video Processing Using MATLAB®, Image Processing, Chapter 15, p.365-386. Available from:* [*https://ebookcentral.proquest.com/lib/open/detail.action?docID=697467*](https://ebookcentral.proquest.com/lib/open/detail.action?docID=697467)

*Özdemii̇r, C. (2012) ‘Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms’, Hoboken, NJ, USA: John Wiley & Sons, Inc.*

*Patra, K.C.,Panigrahi, M. ; Sushil Kumar Mahapatra ; Minu Samantaray, An Enhanced BE-GGMM-EI Algorithm for Medical Image Denoising, Publisher: IEEE, Published: 11-11 Jan. 2016*

*Ratan, R; Sharma, S; Sharma, S.K. Brain Tumor Detection based on Multi-parameter MRI Image Analysis. CGST-GVIP Journal, ISSN 1687-398X, Volume (9), Issue (III), June 2009.*

*Reitsma, A. M. ‘The ethics of medical progress’ (2007) Nature Medicine. Nature Publishing Group, 13(6), pp. 665*

*Saini, P.KR; Singh, M. International Research Journal of Engineering and Technology, Volume: 02 Issue: 02 | May-2015*

*Sullivan, J. R; Dawson, B. M; Rabbani, M. (1992) ‘Image processing algorithms and techniques III : 10-13 February 1992, San Jose, California /’, in. Bellingham, WA: The Society.*

*TM354 Software engineering*

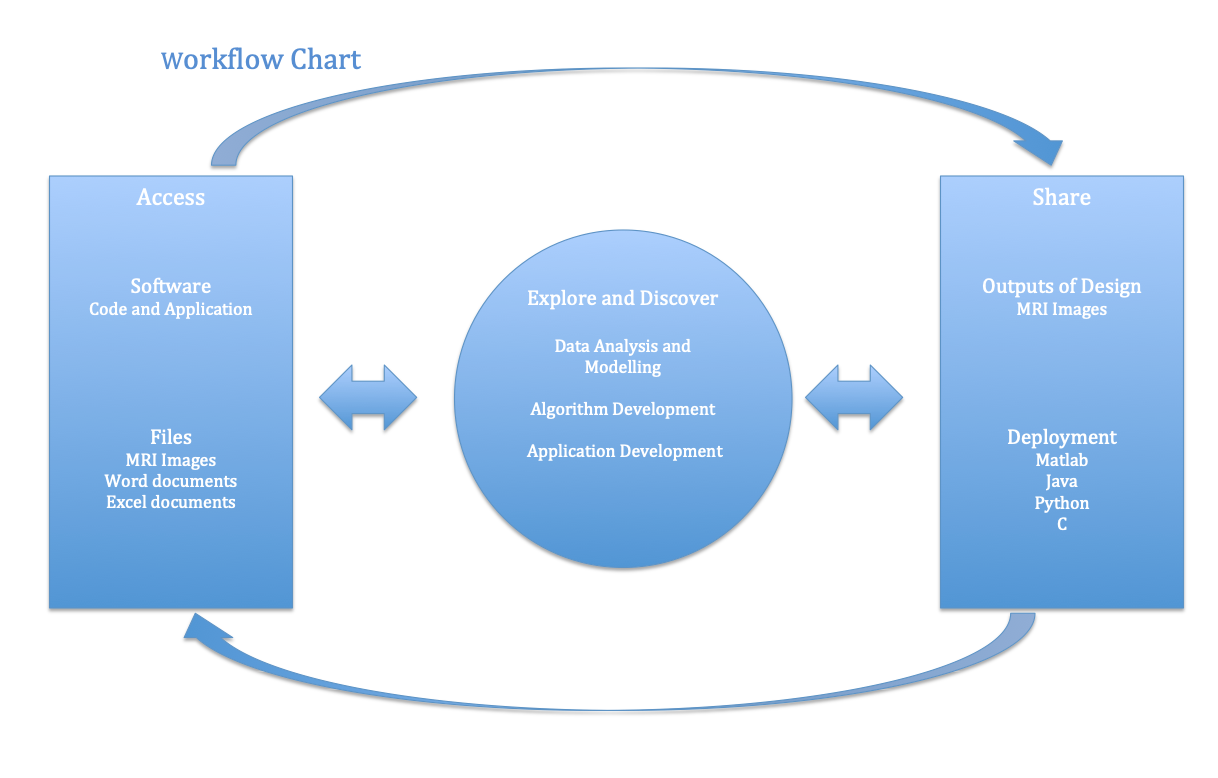
*TM355 Communications technology*

*Venkataramana, B; Padmasree, L; Srinivasa Rao, M; Ganesan, G; Rama Krishna, K. (2017) ‘Implementation of Clustering Algorithms for real datasets in Medical Diagnostics using MATLAB’, Journal of Soft Computing and Applications. International Scientific Publications and Consulting Services (ISPACS), 2017(1), pp. 53–66.*

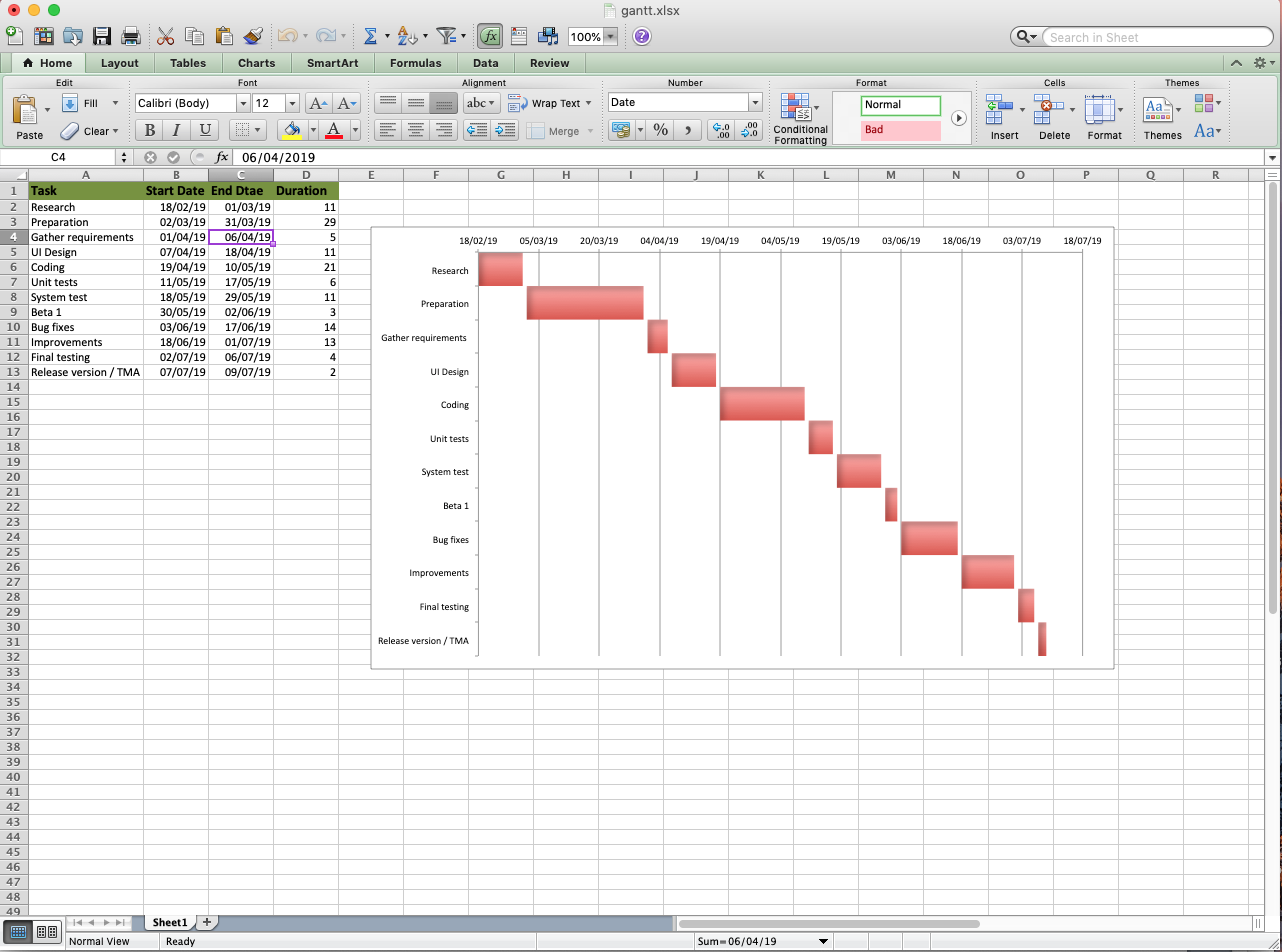
*Zhu, X; Ai, S; Fang, D; Liu, B; Lu, X; (2014). A novel modelling approach of aluminium foam based on MATLAB image processing [online]. Computational Materials Science, Volume 82, Pages 451-456. Available from: https://www-sciencedirect-com.libezproxy.open.ac.uk/search/advanced?docId=10.1016/j.commatsci.2013.10.020*

**Appendices**

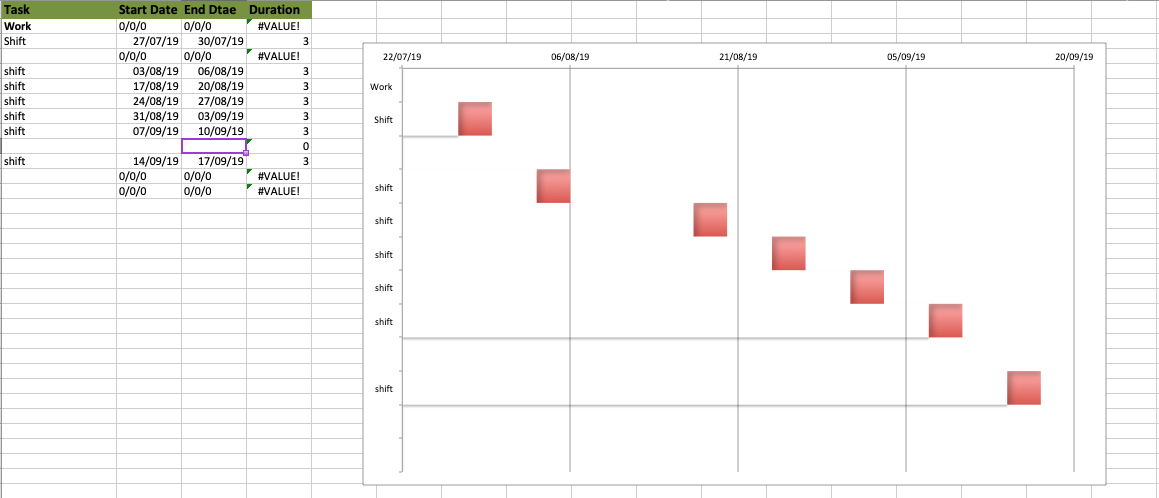
Appendix A

****

Appendix B

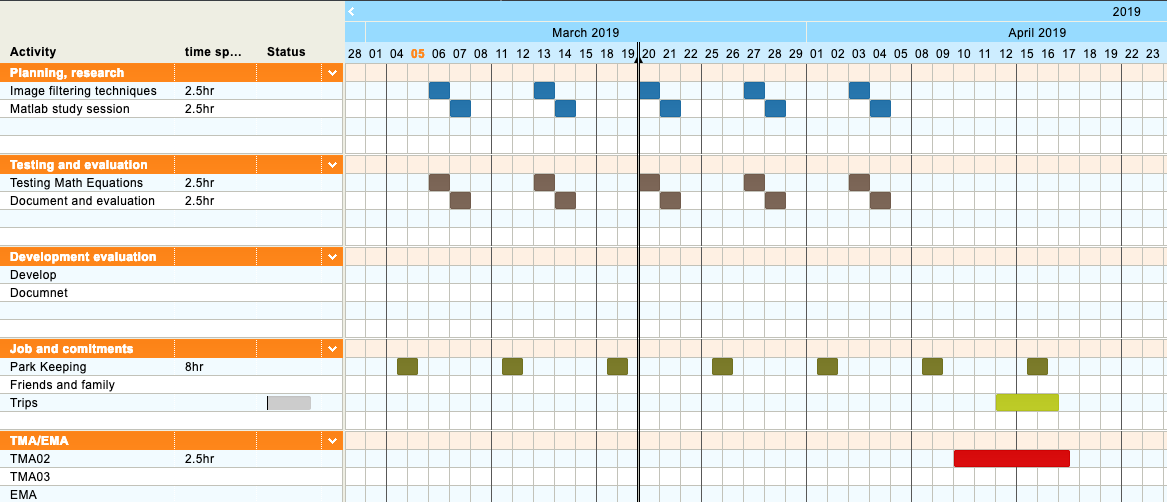


Gantt chart for work shifts

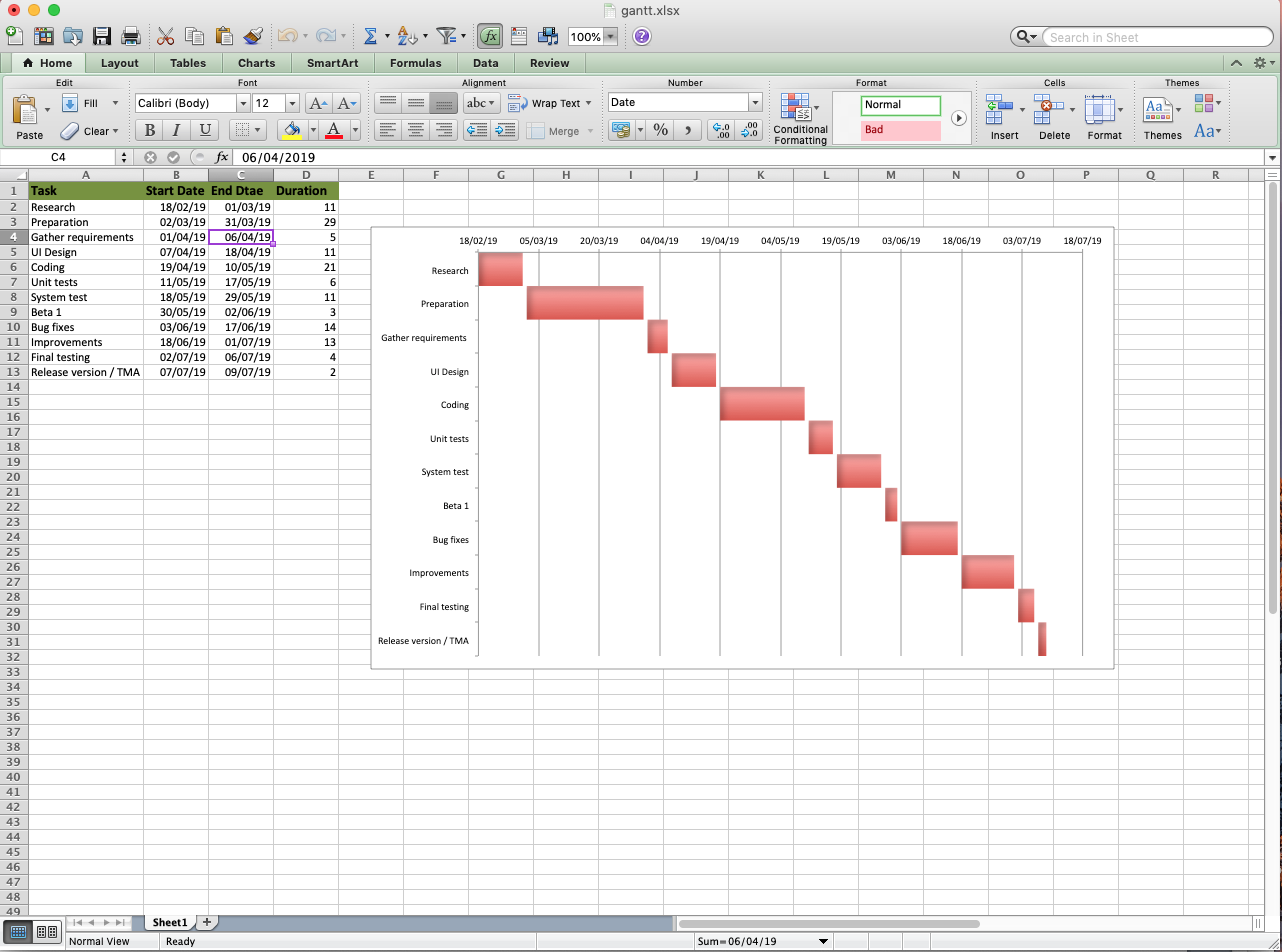


Pervious Gantt chart for TMA 2 and TMA 3

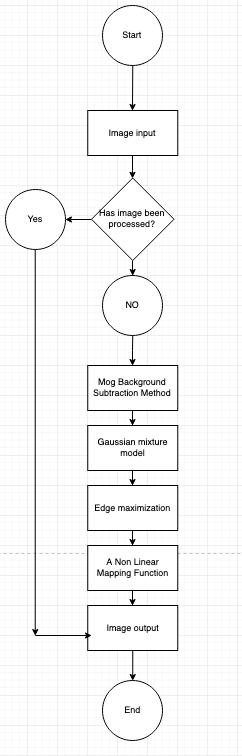
Gantt chart for TMA02 completion



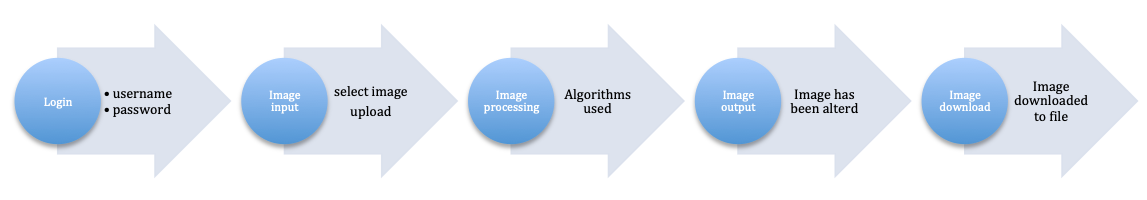
Gantt chart for TMA03 completion



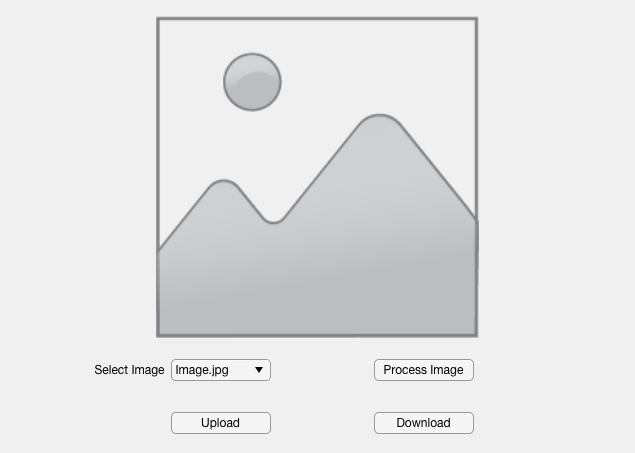
Appendix C



Appendix D



Appendix E

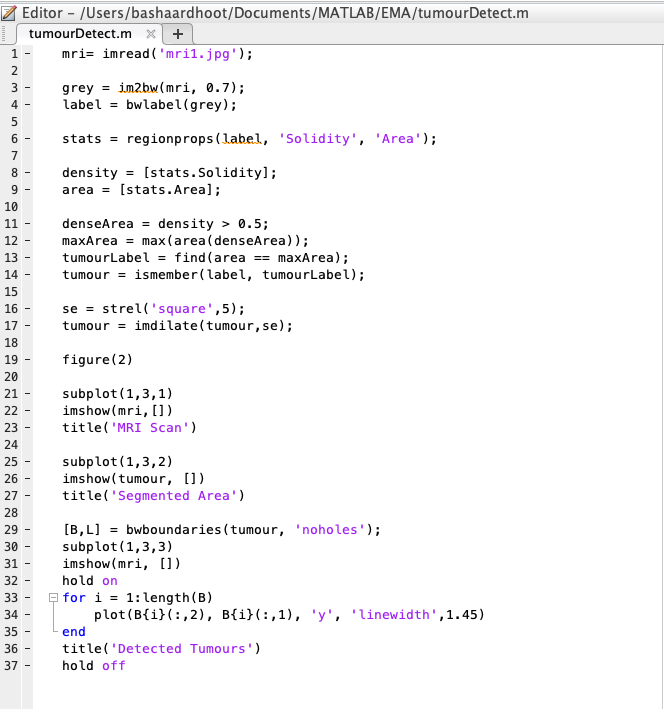


This was one of the fist designs created. The idea was to make the app less cluttered, where I removed one of the image placeholders. The image would be uploaded and processed in one placeholder. This was rejected because Harry would of liked to compare the before and after image.



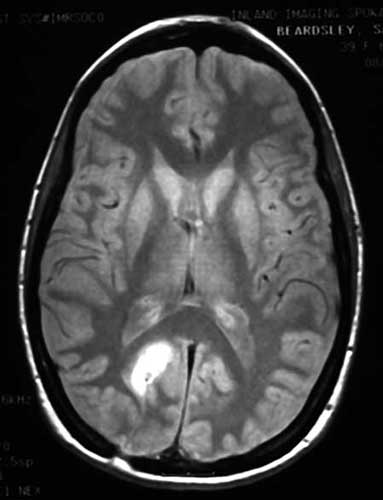
This was one of my preferred UI, as I felt it is much easier for your eyes to move up and down rather than side to side to compare images. However it was rejected for the style and the large amount of space not used. In hindsight I could have had both image placeholders justified and the buttons either side.

Appendix F

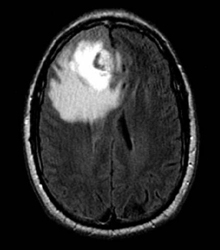


Appendix E

Clear MRI scan showing tumour

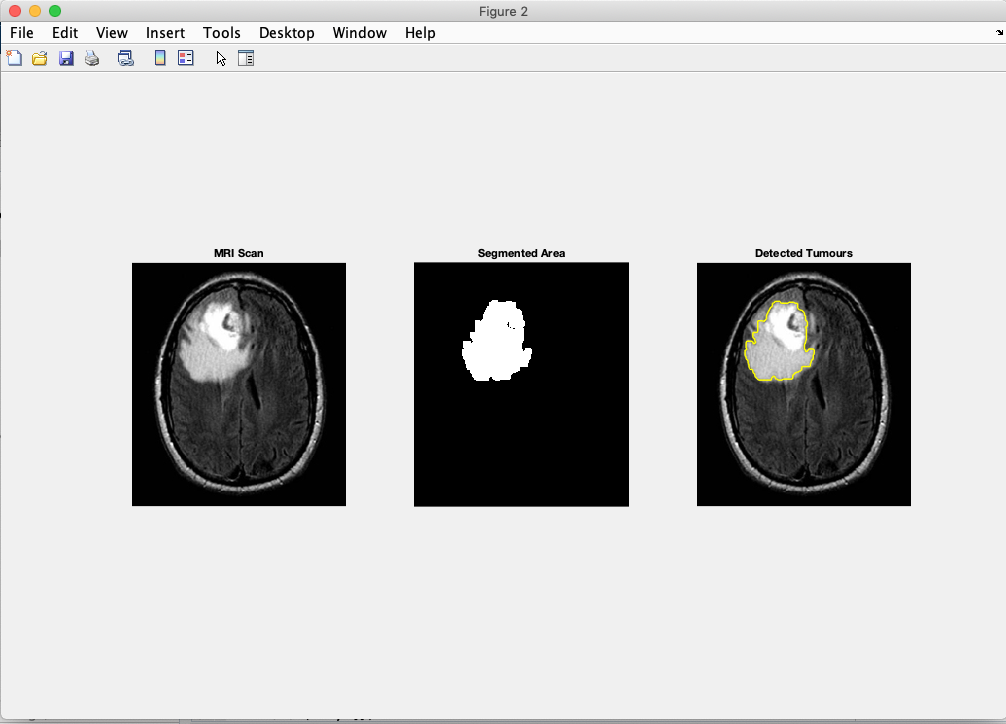


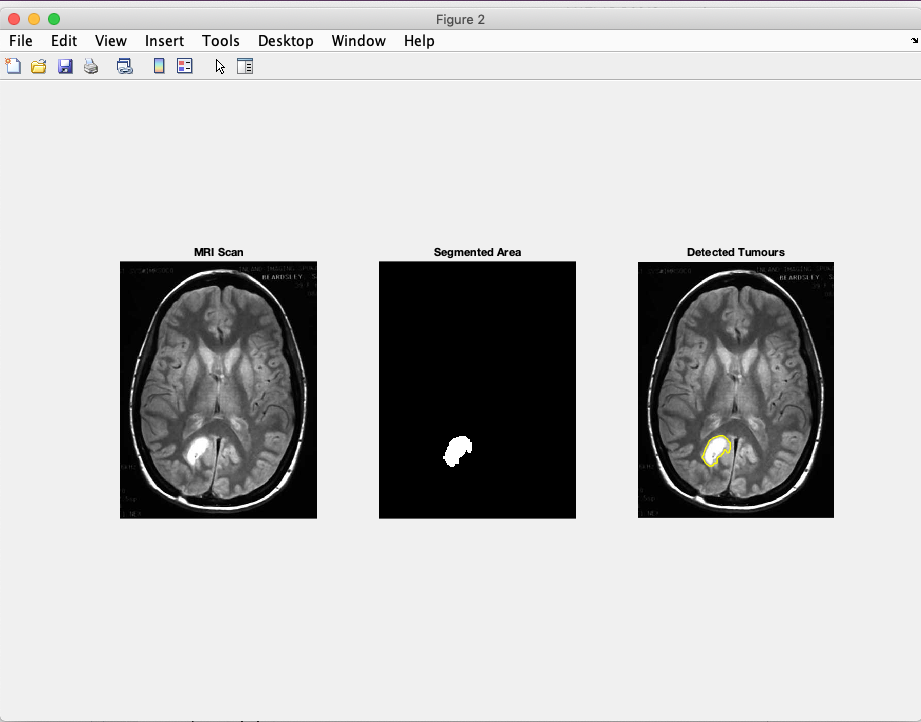
MRI image with noise, where edges tumour are unclear



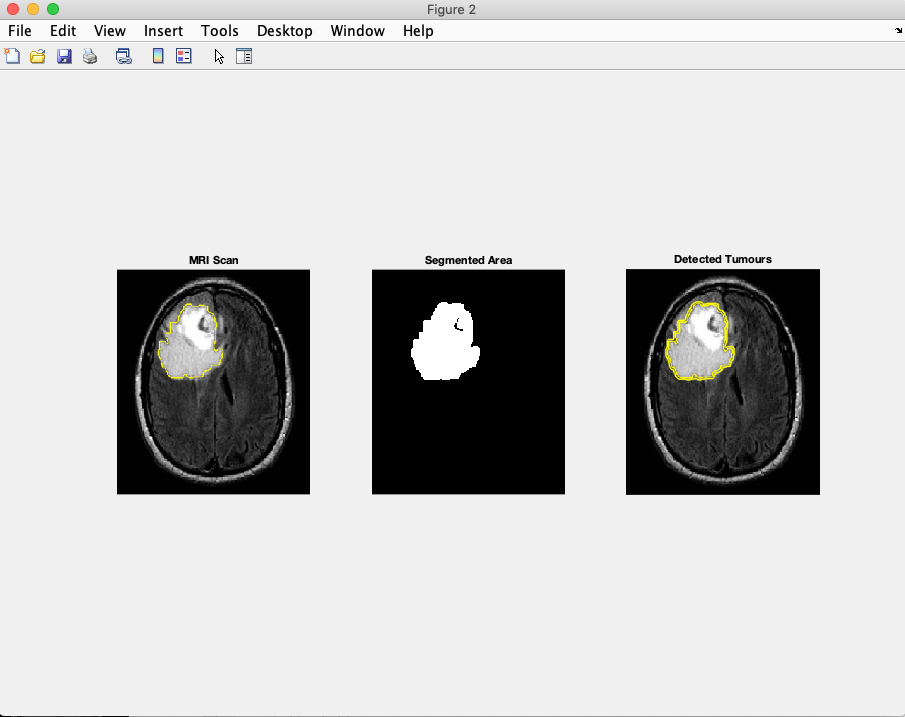
Appendix F

MRI 1

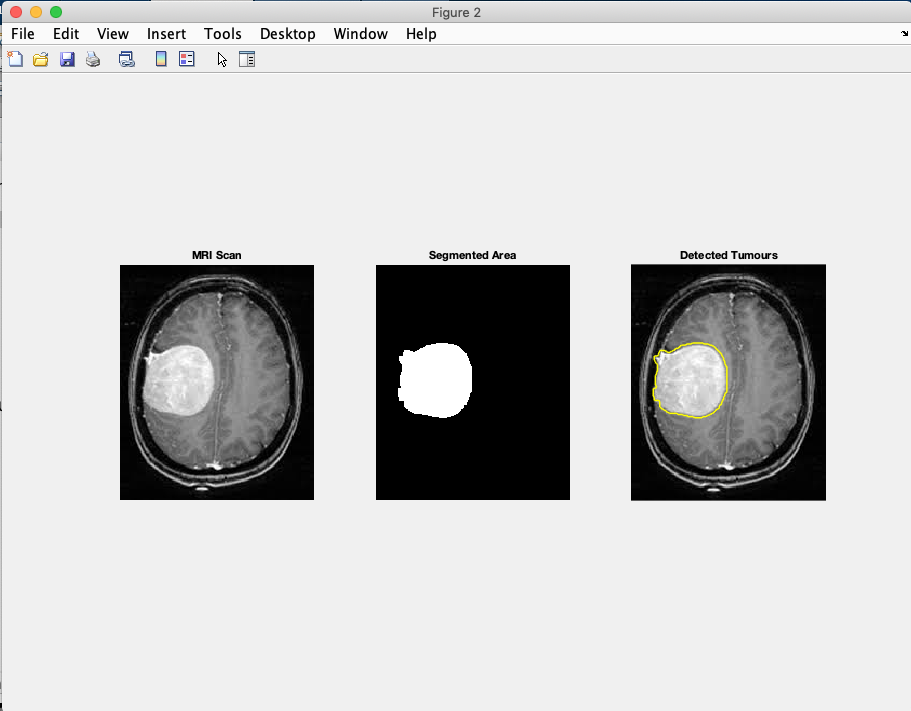
****

MRI 2****

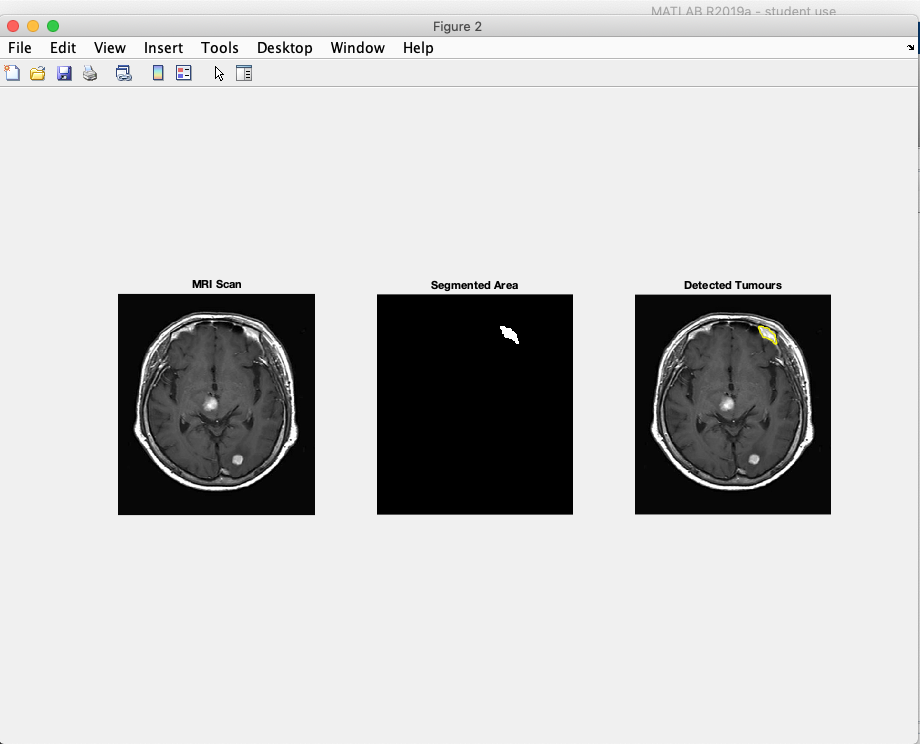
MRI 3



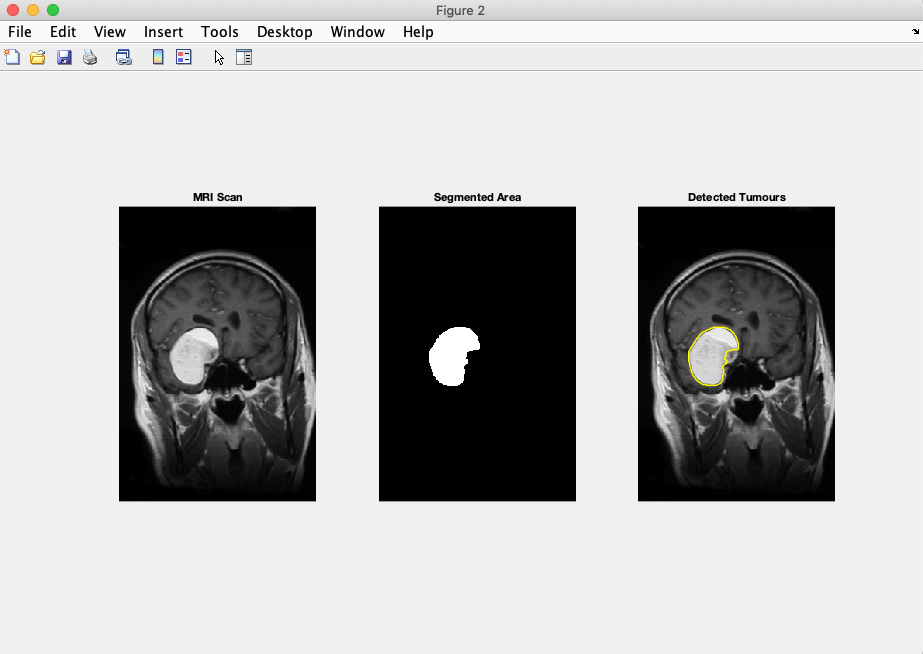
MRI 4



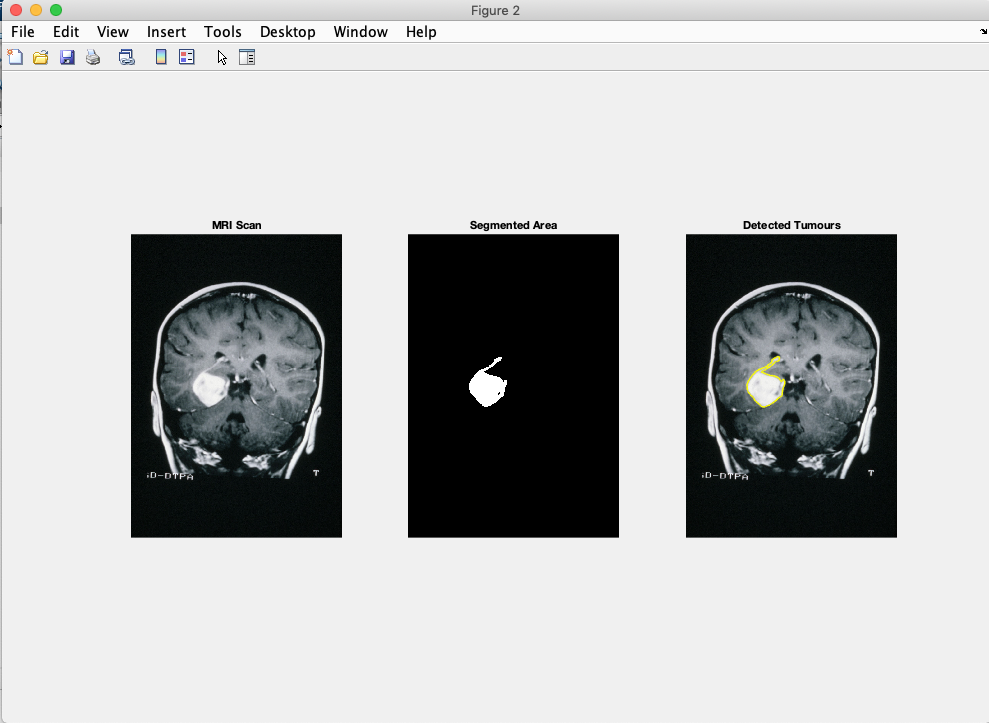
MRI 5



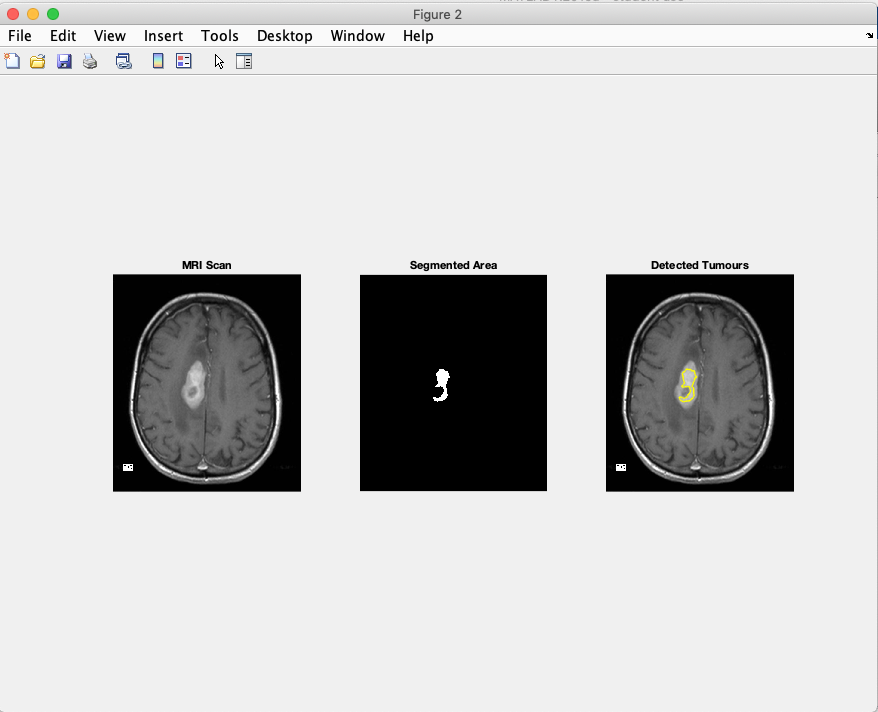
MRI 6



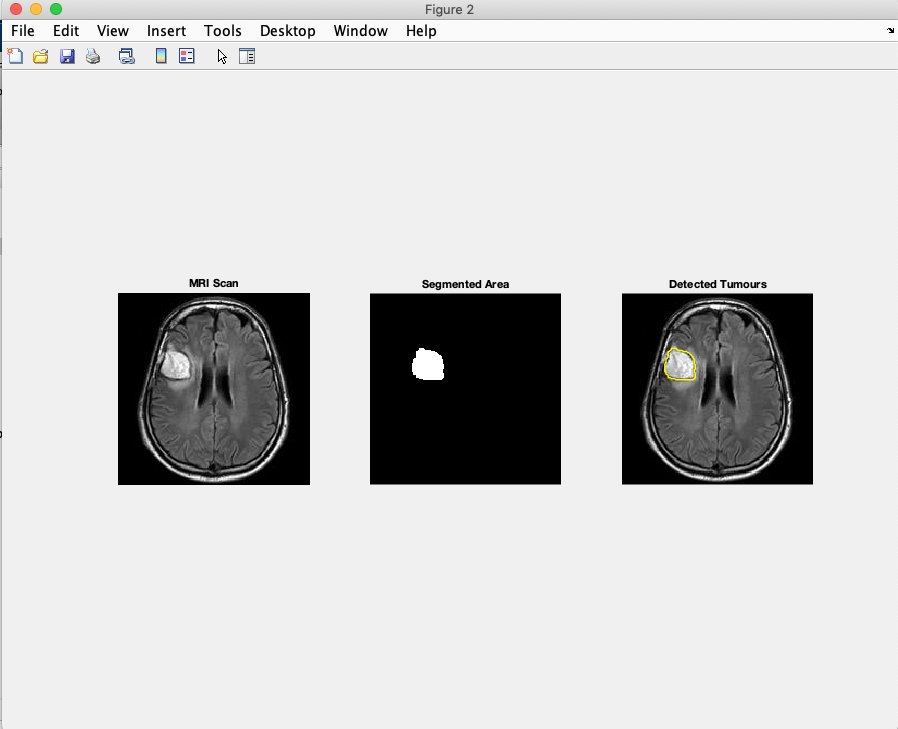
MRI 7



MRI 8



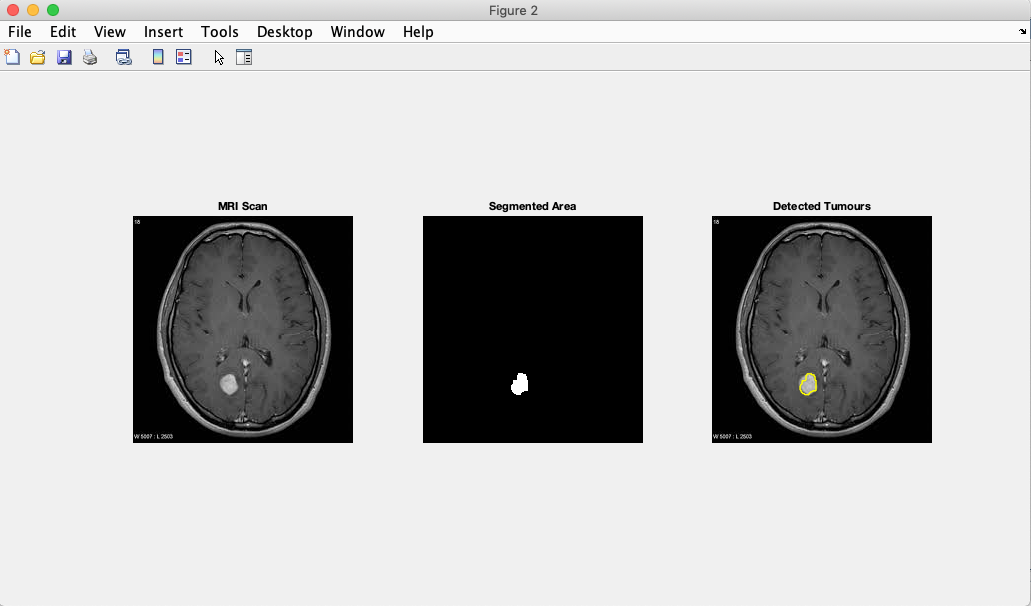
MRI 9



MRI 10



MRI 11



**Journal**

**28/4/2019 – 23/7/2019**

28/4/2019 – 5/5/2019

Gathering requirements: met up with my friend Harry who is a graduate of Kings College. Talked briefly about the design of the UI. Also said I can communicate with him through e-mail to arrange meetings.

\* Started project late due to stress at work. Really have catch up and put more hours.

6/5/2019 –12/5/2019

UI design: started to create multiple UI designs. Emailed User some of the designs. Wilted down to two.

Read up how to code UI.

\* Information on how to code UI out-dated. Seems new syntax and parameters must be used. (i.e. ‘parent, app.Axes’ to call-back image panel.)

13/5/2019 – 19/5/2019

Coding: Designed simple script to turn image black and white. Struggling with the documentation section of the ‘Mathsworks’ website. Starting reading through it again.

Found out how to label and distinguish between blacks and white. Now capable of just highlighting the tumour.

\* Have to take another week to read documentation. Still need find out about more functions

20/5/2019 – 26/5/2019

Coding: very delayed. Have to concentrate on creating a application that just finds tumour.

Found lots of functions in the documentation sections. Must put them in glossary with explanation of what they do.

Created the script, however does not have algorithmic functions and not connected to the UI.

\* must continue to next steps, so atleast I will have something substantial to hand in.

First cycle

27/5/2019 – 2/6/2019

At this stage I should be at the systems test stage. I seem to be one step behind.

Unit test: cannot perform a unit test. Do not have a function UI. Did more research into Unit tests and how to perform and link them to main script.

\* Had to get extra shifts, due to bills. Did 6 day a week.

3/6/2019 – 9/6/2019

Sat down to see what was missing and made a list. Decided leave the script as is and concentrate on UI.

Systems test: Ran more MRI Scans through the system. It can find the tumour but struggled with noise and interference.

\* Meeting with Harry soon, wrote up questions I wanted to ask him

10/6/2019 – 16/6/2019

Beta 1: met up with harry showed him what I achieved. Explained what had happened and guaranteed that they will be fixed. Asked him the questions I wanted to answer. Got a response, which was very helpful. It also added to the requirements.

\*Decided to work on UI, so at least I had a system of some sorts.

17/6/2019 – 23/6/2019

Bug fixes: every tutorial about the UI coding in Matlab out-dated. Slight worry about completion.

Spent hours of trail and error trying to figure out how to code the buttons. In the end managed to complete coding of upload and image processing button.

\* Download button still needs coding

24/6/2019 – 30/6/2019

Improvements/Final testing: my slight improvement for this week was to figure out how to only make sure ‘.jpg’ and ‘.png’ files where uploaded.

\* Need to start typing TMA03

1/7/2019

Might need extension, have a lot to type up!

Extension granted need to continue working.

Need to ask for days off work.

Second cycle