Can entropy-based image alignment metrics offer improved image aggregation of tissue density for mammographic risk assessment?

Final Report for CS39440 Major Project

Author: Laura Collins (lac32@aber.ac.uk)
Supervisor: Dr. Neil Mac Parthaláin (ncm@aber.ac.uk)

4th March 2016
Version: 1.0 (Draft)

This report was submitted as partial fulfilment of a BSc degree in Artificial Intelligence and Robotics (inc Integrated Industrial and Professional Training) (GH7P)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work I understand and agree to abide by the University's regulations governing these issues.

Name ....

Date ....

Consent to share this work

By including my name below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name ....

Date ....

Abstract

Include an abstract for your project. This should be no more than 300 words.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Description

This project is concerned with the alignment of multiple mammographic images using an image-alignment technique called Congealing [42]. The aim will be to implement image-alignment application which allows the user to not only choose standard Shannon entropy to align the images as in [42], but also 2 different light-weight Fuzzy Entropy metrics for alignment - Non-Probabilistic [24] and Hybrid entropy [62]. The user will be able to generate 3 mean images of the input set, 1 for each metric. By utilising different alignment metrics on the same sets of input images the result should be a range of varying average output images, which further may be used to ascertain the most useful entropy algorithm for the alignment of mammographic images.

Each input set of images must belong to the same tissue density category (as covered in Subsubsection 2.1.1.3), but from different patients, to allow the resulting mean image to be an accurate depiction of the average breast structure within that category. Once a mean image is constructed of each category, this should aid radiographers in their qualitative categorisation of a new patients' scans.

Simple and accurate categorisation is important due to the increased risk factors associated with more dense tissue breasts. Therefore if a radiographer can be confident in their categorisation of a patient's breast tissue, should the patient fall within the higher risk category they can receive more frequent, specialised scans, along with more targeted care to detect any abnormalities more quickly should they arise.

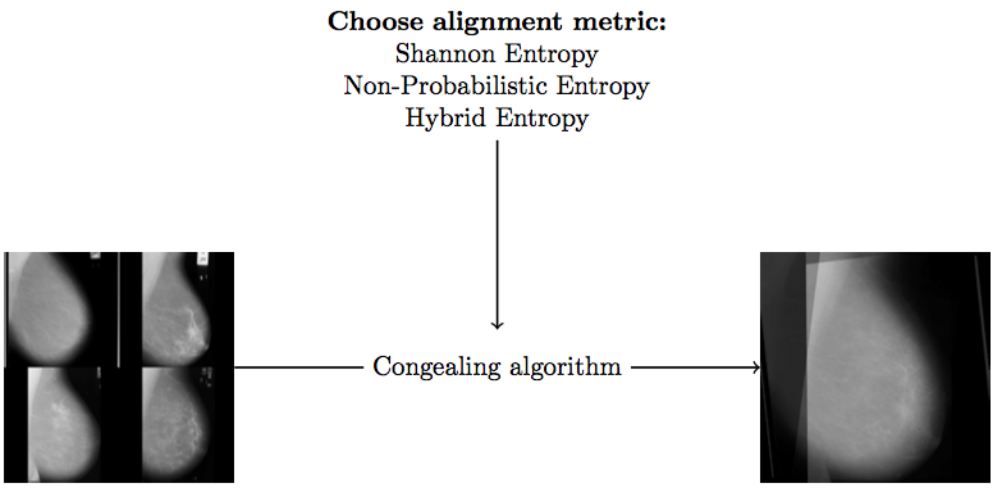


Figure 1.1: Graphical outline of the project

Figure 1.1 outlines the major processes the User would take to align their input images. This includes:

- Selecting the appropriate input data for all of the same BI-RADS classification
- Selecting which entropy alignment metric they would like to be utilise
- These are both fed into the Congealing algorithm
- An end average image is displayed

1.2 Project Structure

This section will give a brief overview of the structure of the project.

1.2.1 Research

The main piece of research to be undertaken in this project will be evaluating which of the fuzzy entropy algorithms will be light-weight and simple enough to be run quickly on a radiographer's own laptop. Typically, research implementations of fuzzy entropy algorithms tend to be complex, and therefore computationally expensive, something not ideal when a patient has a short time-slot with a radiographer.

1.2.2 Software Implementation

In order to assess the usefulness of basic fuzzy entropy algorithms in the alignment of mammographic images, a tool must be built to handle the input images and all the output data. This tool will be created using MATLAB [58] and it's Fuzzy Logic [50] and Image Processing [51] toolboxes.

The main functions of the tool will be:

- Allow the user to input a large image containing all the images they wish to align
- Allow the user to remove any medical markers, or any other artifacts, as they see fit
- Allow the user to choose a particular alignment metric and number of iterations to run on the input images
- Output the final mean image, the adjusted input images (how they look after alignment) and the entropy value of the final image set

1.2.3 Testing

The testing to be undertaken during this project will include scientific and software.

1.2.3.1 Scientific testing

This will be testing the output after the Congealing process has been run using a fuzzy entropy alignment metric. One way to measure the result will be to evaluate the entropy value at the end of the alignment process - as the lower the entropy, the more aligned the images are. Another way in which to test the output of the experiments will be to visually inspect the final mean images produced to see how well aligned the input images are.

1.2.3.2 Software testing

Some software testing will be necessary to ensure the proper working of the tool developed for experimentation. Both Unit testing of the application's functionality and Acceptance testing of the User Stories defined throughout the project will be carried out.

1.3 Objectives

The Objectives for this project are are follows:

- **Align images using fuzzy entropy algorithms.** Images should be passed in, and it should be clear that an aligned version of the input images is calculated and output.
- **Answer any other relevant research questions associated with this project.** These are covered in Subsection 2.3.2
- **Create a tool to streamline inputting images and viewing the output.** As this project uses light-weight, simpler fuzzy entropy algorithms to hopefully speed up processing time *(see next objective)*, then the tool in which you run them should reflect this.
- **Create a quick tool which can be used on anyone's laptop or PC.** Not many people outside of the research community use tools such as MATLAB, so to be able to run a simple executable program is important.
- **Research and implement a solution to remove medical markers from mammogram scans.** As the Congealing algorithm looks to align the scans using grey-level pixel values, then the white medical markers in many mammograms create an issue as these will also try to align.

Chapter 2

Background

2.1 Background Research

In Europe, breast cancer is the leading cause of death through cancer for women, with 1 in 6 women dying from cancer having it in the glandular breast tissue [21]. The UK is contained within the higher mortality band which runs across the EU, sitting alongside countries such as the Netherlands, North-West France and Western Germany (see Figure 2.1). However the reason behind why these countries have a higher breast cancer mortality rate than their neighbours to the north and south is unknown.

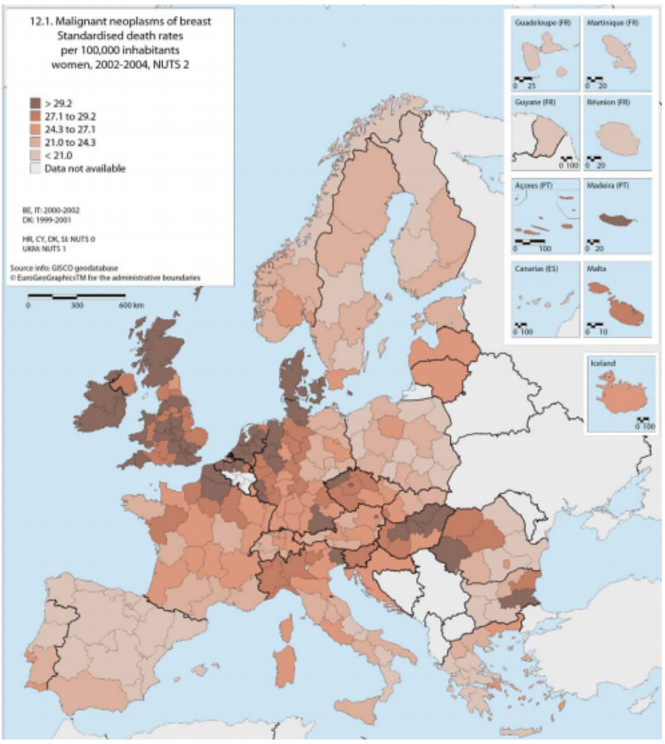


Figure 2.1: Breast tissue composition. *Image Source: EU Commission: Atlas on Mortality [21]*

2.1.1 Tissue density classification

The internal breast structure consists of different kinds of tissue and glands [3]:

- Fatty and connective tissue: protects the lobules and ducts, gives shape to the breasts
- Lobules - milk-production glands
- Ducts - carry milk from Lobules to Nipple

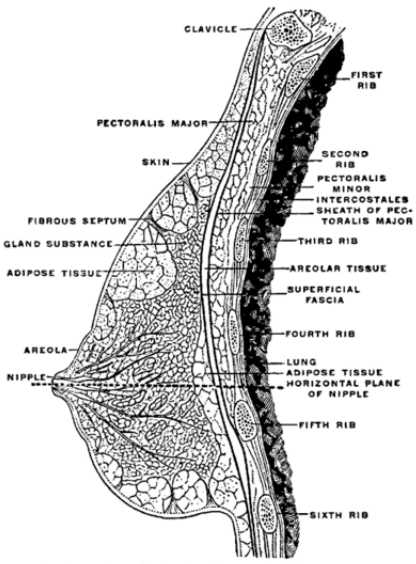


Fig. 1108.—Right breast in sagittal section, inner surface of outer segment. (Testut.)

Figure 2.2: Make up of breast structure. *Image Source: Gray's Anatomy [34]*

Fatty and connective tissue density can vary widely between women. Extensive research is ongoing into the links between a higher proportion of fibrous/glandular tissue versus fatty tissue and a higher risk of breast cancer, however it is currently widely accepted there is a strong link between dense tissue and breast cancer [17]. Therefore, simple classification of denser tissue is vital for both radiographers and patients alike.

There exists several methods for classifying the density of breast tissue, as outlined in the following Subsections.

2.1.1.1 Wolfe classification

Wolfe described the first qualitative means in which to classify breast tissue density in 1976 [83].

- **N1:** consisting mainly of fat (lowest risk)
- **P1:** fat plus linear densities occupying no more than 25% of the breast (low risk)
- **P2:** linear densities occupying >25% of breast (high risk)
- **DY:** dense (highest risk)

2.1.1.2 Boyd classification

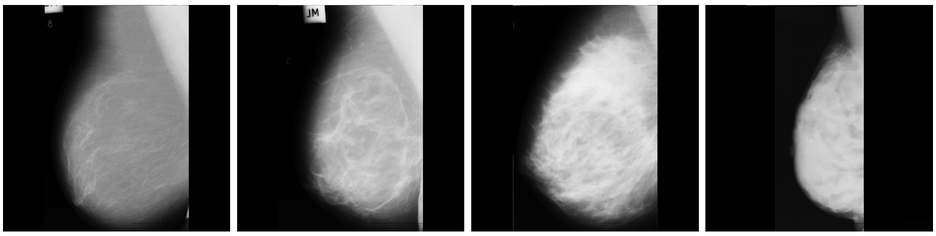
Boyd and colleagues proposed a quantitative means to categorising breast tissue density, based on a percentage of 'dense' tissue assigned by a radiographer [17].

- **A:** 0%
- **B:** >0% - 10%
- **C:** >10% - 25%
- **D:** >25% - 50%
- **E:** >50% - 75%
- **F:** >75%

2.1.1.3 BI-RADS classification

A widely accepted quantitative tool for the classification and risk analysis of mammography and ultrasounds is BI-RADS (Breast Imaging-Reporting and Data System) system, defined by the American College of Radiology [73].

- **a:** almost entirely fatty (Figure 2.3a)
- **b:** scattered areas of fibroglandular density (Figure 2.3b)
- **c:** heterogeneously dense, which may obscure small masses (Figure 2.3c)
- **d:** extremely dense, which lowers the sensitivity of mammography (Figure 2.3d)



(a) (b) (c) (d)

Figure 2.3: Comparison of the 4 BI-RADS classification

This is the classification of choice for this project due to it's wide-spread acceptance and usage in the industry.

2.1.1.4 Tabár classification

This technique is somewhat different from the previous 3 by utilising anatomic-mammographic correlations, as developed by Tabár [33].

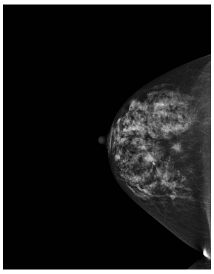
- **I:** balanced proportion of all components of breast tissue with a slight predominance of fibrous tissue
- **II:** predominance of fat tissue (fat breast)
- **III:** predominance of fat tissue with retroareolar residual fibrous tissue
- **IV:** predominantly nodular densities
- **V:** predominantly fibrous tissue (dense breast)

2.1.2 Mammograms

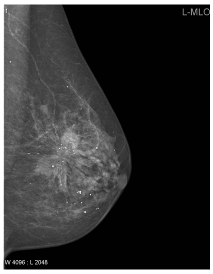
Quite simply, a Mammogram is an X-Ray of the breast tissue pressed between 2 plates from a number of different angles. Below are a selection of the most common angles [66] [13]:

- Cranial-Caudal (CC) - taken from above (Figure 2.4a)
- Medio-Lateral Oblique (MLO) - from the side, at an angle (usually 45$\deg$) (Figure 2.4b)
- Medio-Lateral (ML) - from the centre outwards (Figure 2.4d)
- Latero-Medial (LM) - from the side, into the centre (Figure 2.4c)

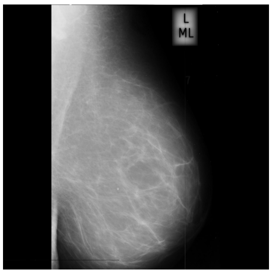
CC and MLO are generally standard practice angles, with ML and LM adding more information for the radiographer to assess.



(a) Cranial-Caudal: Case courtesy of Dr Garth Kruger, Radiopaedia.org, rID: 18580



(b) Medio-Lateral Oblique: Case courtesy of A.Prof Frank Gaillard, Radiopaedia.org, rID: 12608



(c) Medio-Lateral: Case courtesy of Mini-MIAS dataset [76]

(d) Latero-Medial: Case courtesy of Dr Paresh K Desai , Radiopaedia.org, rID: 5873

Figure 2.4: Comparison of the 4 mammogram angles typically used

Organisations such as Breast Test Wales invite women between the ages of 50 and 70 to attend a scan every 3 years [14]. However women with higher-density breasts, which is ascertained during a mammogram, could be called back for more regular screening, to ensure to catch any abnormalities sooner.

2.1.2.1 Alternatives to Mammograms

Although the input data of choice for this project will be Mammographic scans, it is important to remember that for some women, and under some circumstances, it may be more appropriate to use a different method of diagnosis.

Ultrasound

Women under 35 are often offered an ultrasound scan over a mammogram, due to their breasts being of a higher density naturally which makes obtaining a clear mammogram more difficult. Ultrasounds can also show the if the breast lump is a cyst, or if it is solid internally [79].

Biopsy

A Biopsy is usually a secondary step after diagnosis of a breast lump via mammogram or ultrasound. It can take a number of forms including:

- Needle biopsy
- Vacuum biopsy

- Needle aspiration
- Punch biopsy
- Wire guided biopsy

2.1.3 Existing Computer Systems

A lot of the current computer systems focus on mammography computer aided diagnosis due to the ease in which a Radiographer can misdiagnose cancer from a mammogram. Mammograms are difficult to read, or technological issues may occur and as a result, radiographers can fail to detect between 10-15% of breast cancer cases [18].

However this is not the focus of this project. This project aims to focus upon healthy tissue, and the first steps towards computers classifying breast tissue into the correct density category. So what steps have been taken currently, in Industry and in research, towards this end goal?

Paper by Mohamed Abu ElSoud; Ahmed M. Anter - Automatic mammogram segmentation and computer aided diagnoses for breast tissue density according to BIRADS dictionary

Not sure if this section is needed? Cannot yet get ahold of above paper.

2.2 Research Method

For this project, a literature review was undertaken to assess the work completed by researchers in the fields of Entropy, Fuzzy Entropy and image alignment methods to help better understand what has been investigated, and to gain an understanding of the background.

2.2.1 Entropy

In terms of Information Theory, the Merriam-Webster Dictionary defines Entropy to be [1]:

Entropy (noun): the degree of disorder or uncertainty in a system

Shannon entropy, derived by Claude Shannon [70] can be mathematically defined as :

$$H(X) = -\sum_{i=0}^{N} p_i \log_2 p_i \tag{1}$$

Where p_i is the set of probabilities for all the variables in X.

Let us consider a fair coin toss. The probability of heads is exactly $\frac{1}{2}$, therefore, the entropy of landing on heads is:

$$\begin{aligned} H(heads) &= -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2}) \\ &= 1.0 \end{aligned} \tag{2}$$

On the other side, if a system outputs solely the letter "M", then the entropy of receiving the letter "M" is exactly 0. This is because when either the positive or the negative outcome is 100%, then both sides equal "0" when fed into the entropy equation.

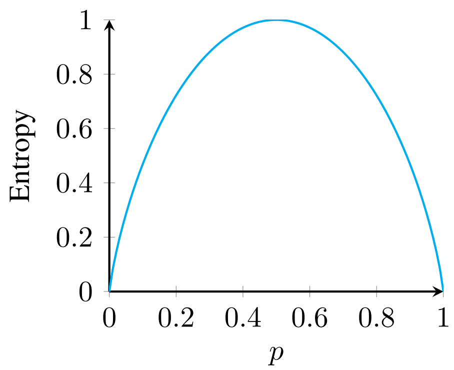


Figure 2.5: Entropy mapped against probability (p) of occurrence.

It follows that entropy can only ever take a value between 0 and 1, with an entropy of 1 have a 50% probability, and an entropy of 0 being 100% certain.

2.2.2 Uncertainty

However real life is not 100% certain - a small amount of uncertainty in life is to be expected and sometimes desired. A surprise party for many is the nice kind, however uncertainty associated with risk - i.e. "Will I lose my job in the recession?" - is uncertainty with a negative impact. Modeling uncertainty is especially important to researchers so they can understand it, and then continue to use it to our advantage in techniques such as fuzzy entropy.

There exists several types of uncertainty in this world that can be modeled by mathematical means:

2.2.2.1 Probabilistic Uncertainty

By definition:

> *Probability: the chance that something will happen [8]*

Probabilistic distribution is a widely accepted and used technique for representing expert judgements of uncertainty [60]. Early work carried out by DeGroot (1970) [26], built upon that of Savage (1954) [69], gave a simple layman's explanation:

> *For instance, if the person prefers decision A to B and B to C then they must also prefer A to C.*

2.2.2.2 Possibilistic Uncertainty

By definition:

> *Possibility: a chance that something might exist, happen, or be true : the state or fact of being possible [7]*

Possibilistic uncertainty (closely related to "fuzziness") indicates the lack of information we hold about the possible outcome values from a system - a sort of ambiguity. Possibilistic uncertainty models the possible outcomes from a system, as estimated by a decision maker because it is possibly impossible to determine beforehand [80].

2.2.2.3 Indiscernibility Uncertainty

By definition:

> *Indiscernibility: the quality or state of being indiscernible [5]*
> *Indiscernible: impossible to see, hear, or know clearly [6]*

Find an explanation of Indiscernibility

2.2.3 Fuzzy Entropy

Fuzzy entropy stems from combining standard Shannon entropy with the practices of Fuzzy Set Theory, discovered by Zadeh in 1965 [84]. This introduces the idea of "Membership" to a category, where an object can belong to more than one category to a certain degree.

One common example of this is listing someone as 'Short', 'Average' or 'Tall' in height. If a tall person is someone over 6 feet in height, would a person who measured 5foot 11inches not be classified as tall? Given crisp sets, then they would be classified as 'Average'. In fuzzy set theory, they would be be a certain degree of tall, and a certain degree of average, with the highest membership likely to win out when categorising their height. Another example of this can be seen in Figure 2.6

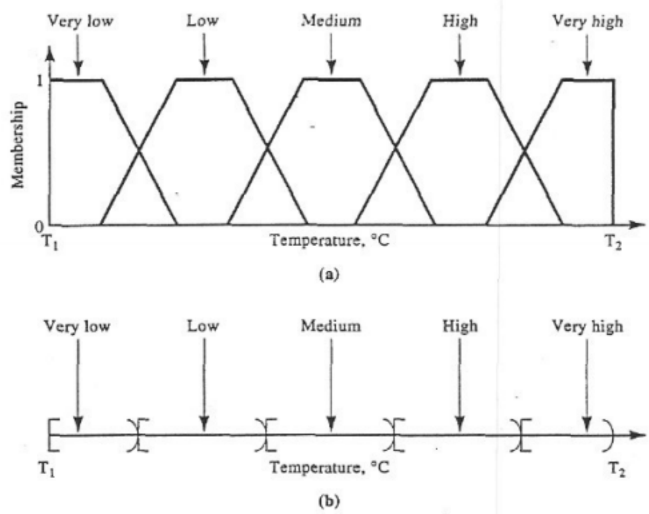


Figure 1.4 Temperature in the range $[T_1, T_2]$ conceived as: (a) a fuzzy variable; (b) a traditional (crisp) variable.

Figure 2.6: A comparison between Fuzzy Sets and Crisp sets. *Image Source: Fuzzy Sets and Fuzzy Logic: Theory and Applications [28]*

After combining Fuzzy Set Theory with Entropy, then the amount of fuzzy information gained from the fuzzy set(s) is known as fuzzy entropy.

2.2.3.1 Non-Probabilistic Entropy - 1972

De Luca and Termini are considered to be the first to have taken Shannon Entropy and extended it to include fuzziness [24]. They also defined properties which a fuzzy entropy must follow, in order to be classed as true.

Their non-probabilistic fuzzy entropy equation is as given:

$$H_A = -K \sum_{i=1}^{n} \{\mu_i \log(\mu_i) + (1 - \mu_i) \log(1 - \mu_i)\} \tag{3}$$

Where μ is the maximum membership across all the fuzzy sets.

The entropy given by equation (3) satisfies all four of De Luca and Termini's defined properties:

P-1 $H_A = 0$ iff A is a crisp set ($\mu_i = 0$ o r$1 \forall x_i \in A$) (4a)

P-2 H_A is maximum iff $\mu_i = 0.5 \forall x_i \in A$ (4b)

P-3 $H \geq H^*$ where H^* is the entropy of A, a sharpened version of A (4c)

P-4 $H = \overline{H}$ where $\overline{H}$ is the entropy of the complement set $\overline{A}$ (4d)

2.2.3.2 Fuzzy Shannon Entropy - 1989

Sander [68] presented a characterisation of a fuzzy entropy some time after De Luca and Termini's work was published. His implementation of Shannon fuzzy entropy is laid out in equation (5) below:

$$H(f) = -c \sum_{i=1}^{n} f(x_i) ln f(x_i), c > 0 \tag{5}$$

Where the power of a fuzzy set is defined as:

$$P(f) = \sum_{i=1}^{n} f(x_i) \tag{6}$$

Sander further went on to propose some properties, which must be imposed on a fuzzy entropy d to ensure that $d(f) = H(f)$:

1. Sharpness: $d(f) = 0 \Leftrightarrow f(X) \subset 0, 1, f \in [0, 1]^X$ (7a)

2. Valuation: $d(f \wedge g) + d(f \vee g) = d(f) + d(g), f, g \in [0, 1]^X$ (7b)

3. Generalised additivity: There exists two mappings s,t: $[0, \infty) \to [0, \infty)$ such that $d(fxg) = d(f)t(P(g)) + s(P(f))d(g)$ for all $f \in [0, 1]^X, g \in [0, 1]^Y$, where X and Y are finite sets. (7c)

2.2.3.3 Object-background segmentation using new definitions of entropy - 1989

Pal & Pal outlined their first fuzzy entropy algorithm in 1989 [61], which satisfies all four of De Luca and Termini's conditions (outlined in Equations(4)). It as as follows:

$$H = -k \sum_{i=1}^{n} \{\mu_i exp(1 - \mu_i) + (1 - \mu_i) exp(\mu_i)\} \tag{8}$$

2.2.3.4 Higher Order Fuzzy Entropy & Hybrid Entropy - 1992

In Pal & Pal's paper "Higher order fuzzy entropy and hybrid entropy of a set" [62], they not only prove some of De Luca & Termini‘s work to be flawed, but also defined two new fuzzy entropy algorithms, and a new set of definitions.

Higher Order Fuzzy Entropy

As defined by Pal & Pal:

- $P =$ Fuzzy property set
- $\mu =$ the degree to which x_i possesses the property P
- $n =$ number of elements, with $r =$ a combination of elements from group n
- $S_i^r =$ denotes the ith element of such a combination
- $\mu(S_i^r) =$ the degree to which the combination S' as a whole possesses P
- There are $\left[\binom{n}{r}\right]$ such combinations

The entropy of order r of the fuzzy set A is defined as:

$$H' = \left(\frac{I}{\binom{n}{r}}\right) \sum_{i=1}^{\binom{n}{r}} \{\mu(S_i^r)exp(1-\mu(S_i^r))\} + \{1-\mu(S_i^r)\}log\{\mu(S_i^r)\} \tag{9}$$

If $r = 1$, then (9) reduces to Equations (8) and (3)

Hybrid Entropy

Another fuzzy entropy implementation outlined in Pal & Pal‘s paper was Hybrid Entropy. This algorithm is particularly useful as it combines Probabilistic and Possibilistic (fuzziness) uncertainty and if fuzziness is removed or not present, it returns to that of a classical set.

Let us define Hybrid Entropy.

- Let p_0 and p_1 be the probabilities of receiving 0 and 1 symbols over a noisy digital communication line respectively.
- Let μ denote the membership functions of the fuzzy set "Symbol close to 1"
- Both E_1 is a monotonically increasing function of μ - E_0 can be perceived as the likelihood (possibility) of receiving a "1" symbol
  - as μ increases from 0 to 1, then E_1 also increases
  - e.g. with an incoming "0" symbol, if μ increases, than the difficulty of correct interpretation also *increases* - a wrong interpretation of a "0" becomes likely
  - e.g. for an incoming "1" symbol, if μ increases, then the difficulty of correct interpretation *decreases* - improving likelihood of correct classification

- At the same time, E_0 can be perceived as the likelihood (possibility) of receiving the "0" symbol for the same reasoning

E_0 and E_1 can be defined as:

$$E_0 = \frac{1}{n} \sum_{i=1}^{n} (1 - \mu_i) exp(\mu_i) \tag{10a}$$

$$E_1 = \frac{1}{n} \sum_{i=1}^{n} \mu_i exp(1 - \mu_i) \tag{10b}$$

Therefore, the hybrid entropy of fuzzy set A can be defined as:

$$H_{hy} = -p_0 \log(1 - E_0) - p_1 \log(E_1) \tag{11}$$

2.2.3.5 Fuzzy Entropy: a Brief Survey - 2001

Due to the older nature of some of the papers listed above, some were difficult to locate online. So when implementing the chosen algorithms (Non-Probalistic Entropy and Hybrid Entropy), Alsharhan et al's paper "Fuzzy Entropy: a Brief Survey" [15] was a useful tool.

Its concise nature, and chronological listing ensured a strong understanding of the basic principles, before introducing the more complex algorithms (such as Higher Order Fuzzy Entropy). The paper also highlights advantages and flaws to each solution.

2.2.4 Joint Image Alignment

Joint image alignment, occasionally otherwise known as groupwise image alignment, focuses on the alignment of several images, into one average image. This research area has been particularly prevalent in areas such as medical and facial imagery [77] [22]. Cootes et. al. leverage a groupwise registration algorithm to choose one base image with control points to align (typically a standard mesh frame), analyse each following image in turn estimating the movement needed to align corresponding control points, then iteratively warp each image to fit the reference frame, adjusting the texture model as they‘re aligned. This type of alignment will not be considered for the project due to the over-complexity needed for the input image, along with computational limitations due to aligning one image at a time with the base image.

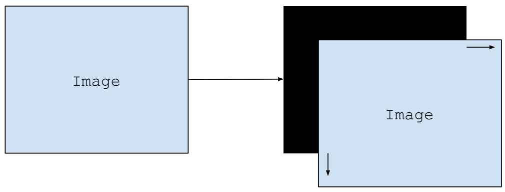
This Subsection will look into a couple of the techniques which will be suitable for this project.

2.2.4.1 Learned-Miller‘s Congealing

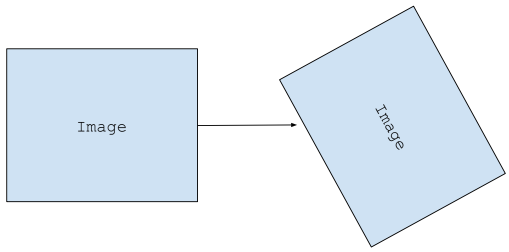
Learned-Miller's Congealing [42] is often cited as being one of the first to truly align simple sets of data (which must have minimal noise, no occlusions and illumination variation) [85] [63]. Many more robust image alignment techniques have been developed off of the basis of this work, however with more computational-expense.

This algorithm works by iteratively reducing the pixel-wise entropy over the input images, using a set of standard image transformations, in a non-deterministic manner, such as:

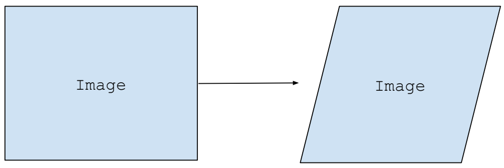
- x & y translations (Figure 2.7a)
- rotation (Figure 2.7b)
- x & y shear (Figures 2.7c & 2.7d respectively)
- x & y scale (Figure 2.7e)



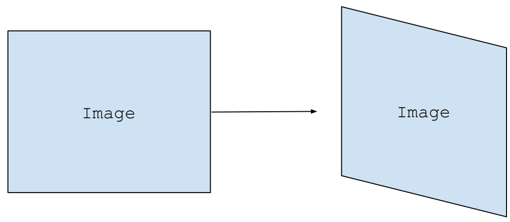
(a) Image translation in the x & y axes.



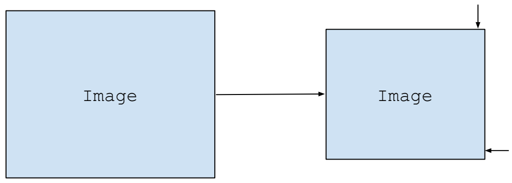
(b) Image rotation about the origin.



(c) Image shear in the x axis.



(d) Image shear in the y axis.



(e) Image scale in the x & y axes.

Figure 2.7: Image transformations executed by the Congealing algorithm

The entropy is calculated by assessing each individual set of pixel-locations in the 'Pixel Stack' (see Figure 2.8), and by calculating the entropy of the empirical distribution of values in the Pixel Stack.

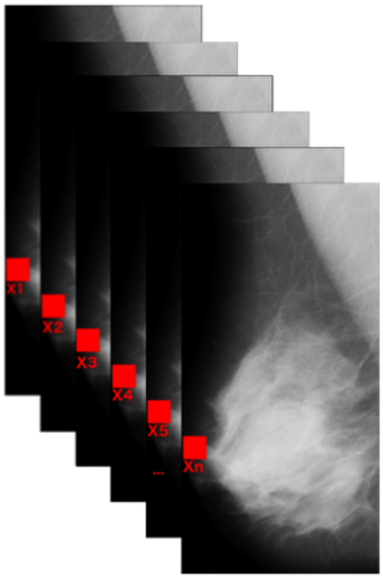


Figure 2.8: Each pixel from the same location throughout the set creates a 'Pixel Stack'

2.2.4.2 Least squares Congealing for unsupervised alignment of images

Further work was done upon the Congealing algorithm proposed by Learned-Miller by Cox et al. in 2008 [23]. They set out to address any performance issues and to remove the need for a pre-defined step size. It proposes to mitigate these issues by implementing an alternative method for aligning the images - utilising the Lucas & Kanade algorithm for aligning a single image to another using a gradient descent approach [46].

2.2.4.3 Unsupervised Joint Alignment of Complex Images

Huang and a team (notably including Learned-Miller) further extended the Congealing algorithm to be usable upon complex images - such as faces and cars at different orientations [38]. This method removes the need to hand-label the input data and improves the performance of face recognition systems, by ensuring the objects are properly oriented prior to recognition.

2.2.5 Image Alignment using Fuzzy Entropy

Research has been undertaken in the past to investigate image alignment using fuzzy entropy metrics, however typically they were found to be computationally costly, and therefore slow to run on a conventional PC or laptop. This project will be investigating whether there are simpler, more light-weight fuzzy entropy metrics which could be implemented, for an increased chance that image alignment techniques can be leveraged in everyday-life. It will also be investigated if, and further how, the outputs of these alignments differ per each fuzzy entropy metric.

Some of this work which has implemented a more computationally-costly Congealing algorithm is that presented by Mac Parthaláin and Strange in their 2013 paper "Fuzzy-entropy based image congealing" [47]. Their implementation included dynamically-calculated fuzzy sets and a fuzzy similarity relation matrix - allowing a comparison of all the objects simultaneously to each other.

Further work upon Mac Parthaláin and Strange's technique was published in 2014, by Li a visiting-student to Aberystwyth University [44]. Li's work, which was supported by Mac Parthaláin and other Aberystwyth staff, considered the application of similarity relation fuzzy entropy and image congealing in order to carry out chinese character recognition. One of Li's final conclusions was that similarity relation fuzzy entropy was too computationally costly for the application, and that another algorithm from Al-Sharhan et. al's fuzzy entropy paper [?] may have been more appropriate.

2.3 Analysis

2.3.1 Task composition

After both the background research and literature review were completed, a list of main “Tasks” to be undertaken in order to complete this project was composed. These are outlined in the following subsections.

1: Decide how best to implement Membership functions

Fuzzy entropy requires Membership functions, in which the image data can be classified. This task would be to decide whether to dynamically calculate the Membership functions, or whether to statically define them within the back-end of the system. The take would also be to decide upon which shape of Membership function the application would use. If to be statically defined in the backend of the system, what values would the bases and shoulders of the sets take?

2: Research and choose which fuzzy entropy algorithms would be best suited to this project

This project stresses the importance of running image alignment techniques on a standard laptop or PC. With this in mind, each fuzzy entropy algorithm would need to be analysed based on their simplicity of calculations, in order to run quickly.

3: Research and decide upon which image alignment technique would be best

Given this project is about aligning images using entropy techniques, this may indeed be an easy task to complete, as options are limited. However, other options should indeed be considered, especially if they are accurate and have low computing cost.

5: Decide which programming language to use

A lot of image processing can be run using programs such as Python, however the demo code given by Learned-Miller [42] has been compiled in MATLAB. Research would have to be undertaken to see if Learned-Miller‘s approach has been implemented in any other languages if this is the image alignment approach decided upon.

6: Decide how best to represent the input and output data

This most likely would be in the form of a GUI as the target user - Doctors - wouldn‘t have access to tools like MATLAB, nor be well-versed in using the Command Line or Terminal.

7: Determine how well to best assess the output

This could possibly involve a professional medical opinion? Or it could be based of the amount the entropy declined over time? Or it could simply be a visual inspection of how well the scans aligned.

8: Research different ways in which to build the GUI

Which programming language would this be done in? If MATLAB were to be chosen, it is capable of linking in which many other languages, such as Java, C and C++. However if the image alignment was done in another programming language, it may be possible to stick to just one language.

9: Determine which dataset to use

Given the data of choice - Mammograms - there exists an ethical issue of utilising people’s medical scans. Research would have to be undertaken to determine whether there are freely-available, license-free Mammographic scans available for use in this project.

2.3.2 Research questions

The research questions are tightly interwoven with the Objectives of this project, outlined in Section 1.3

2.3.2.1 Does the use of fuzzy entropy alignment metrics improve the alignment of mammograms?

Through background research it would follow that there would be no issue in aligning images using fuzzy entropy techniques. However the implementation might be somewhat difficult or computationally-costly, which would undermine the objective of a quick application.

And if so, could one fuzzy entropy metric be more useful than another? As the uncertainty in fuzzy entropy will help model different types of tissue, the way in which they assess uncertainty will affect the output image.

2.3.2.2 Do clinicians / radiographers / mammographers find the output at all useful?

Does the output show a Doctor something useful? Does one of the fuzzy entropy alignment metrics show something different to one of the others? These are questions which would help to validate the success of this research question.

This would be a step towards making a tool that one day could be utilised by Doctors worldwide to aid in their classification of a patient‘s breast tissue, by giving a second opinion based on image analysis.

2.3.2.3 What advantages / disadvantages does each fuzzy entropy alignment metric entail?

One algorithm may be slower, but produce better results, so it is important to weigh up the speed versus the quality of the output. This may mean that one of the wishes for the project: speed and accuracy, may be forfeited for the other.

Chapter 3

Experiment Methods

3.1 Overview

In order to test the main hypothesis of "Does the use of Fuzzy Entropy alignment metrics improve the alignment of mammograms?" some kind of application was needed to portray the visual output. It would be built to take a set of input images, allow the user to select the alignment metric plus select the amount of iterations desired and output the final congealed image. Details of the decreasing entropy would be a key output, along with the average image after each iteration completed, for a full picture of improvements.

However, the decisions about how to implement membership, which fuzzy entropy algorithms to use and which image alignment techniques remain.

3.1.1 Pixel Membership

From the analysis of the planned Fuzzy Entropy algorithms, one major task to be undertaken would be to calculate the membership of each pixel. Membership stems from Fuzzy set theory, as outlined in Subsection 2.2.3.

There are two common methods to modeling degrees of membership. The first is to manually define the category boundaries, so in the case of trapezium functions, the two bases and the two shoulders. The other solution would be to iterate over the values you have and to computationally build an even distribution throughout your membership functions, as in [47]. Whilst this is the preferred method for being dynamical in it's calculations, it is also more computationally expensive as pre-processing of the image would have to be completed before the Congealing algorithm could be run.

Taking the computational-expense into account, for grey-level pixel values, ranging from 0 (black) to 255 (white), two or three trapezium functions would be sufficient, therefore modeling 'Low', 'Medium' and 'High' grey-level values. The bases and shoulders would be statically defined, as in Figure 3.1 and Table 3.1. For Non-Probabilistic entropy the highest membership for each pixel from each of the three trapezia would be taken as the membership degree. Hybrid entropy would take a slightly different approach, which will be covered later.

<table>
<tr><td colspan="4">Low</td><td colspan="5"></td></tr>
<tr><td colspan="2"></td><td colspan="5">Medium</td><td colspan="2"></td></tr>
<tr><td colspan="5"></td><td colspan="4">High</td></tr>
<tr><td>0</td><td>50</td><td>60</td><td>70.4</td><td>85</td><td>170</td><td>195</td><td>205</td><td>255</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Table 3.1: Intepretation of the fuzzy sets across greyscale values

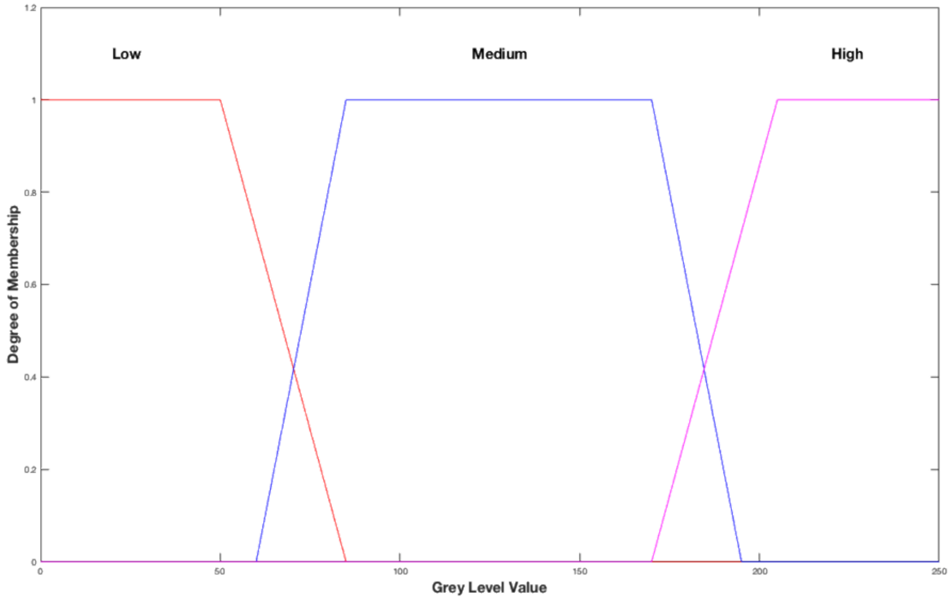


Figure 3.1: 3 trapezium-shaped membership sets

3.1.2 Fuzzy Entropy choices

Chosen algorithms:

- Non-Probabilistic Entropy
- Hybrid Entropy

Given the simplistic nature of Non-Probabilistic entropy, this was one of the chosen Fuzzy Entropy algorithms to be implemented in the project.

Hybrid entropy was chosen for implementation in this project due to it's hybrid nature (implementing both Probabilistic and Possibilistic uncertainty) and for it's simplification nature - in the absence of fuzziness, then E_0 and E_1 reduce to p_0 and p_1 respectively, therefore classical Shannon entropy. This is especially useful in image processing, and other such areas which deal with a lot of noise.

Additionally, Shannon Entropy has already been implemented by Learned-Miller's Congealing algorithm, so this offers a non-fuzzy alternative to image alignment.

Discarded algorithms:

- Fuzzy Shannon Entropy
- Higher Order Entropy

The initial plan was to implement the Fuzzy Shannon entropy algorithm in the project - however after further investigation which revealed that the algorithm does not model Probabilistic uncertainty - it was decided that it was to be excluded.

This project does not implement Higher Order Fuzzy Entropy due to the computational-overhead needed to run - especially on images with as much detail as a mammogram.

3.1.3 Image Alignment choice

Image Alignment choice:

- Congealing algorithm

As this project will be working with mammograms, something with little variation nor inconsistency, Congealing is the perfect, light-weight image alignment algorithm to which to build upon, especially as the demonstration code available for research has a Shannon entropy implementation already developed.

Discarded Image Alignment choice:

- Least squares Congealing
- Joint Alignment of Complex Images

Least squares Congealing algorithm was disregarded for this project due to the preference to focus upon entropy-based alignment algorithms and the computational costs that the authors themselves regard to be a drawback of their algorithm.

The Complex Images implementation of Congealing was quickly identified as overly complex for this project. The original Congealing algorithm was more appropriate for grey-scale mammograms, with a consistent canonical pose.

3.2 Implementation tools

This section will going into detail about the tools used to implement the application built to support the hypotheses.

3.2.1 MATLAB

MATLAB [58] was chosen as the main implementation tool for the project as it is specifically designed to aid in scientific research. Furthermore, MATLAB was the ideal choice as the original Congealing algorithm was implemented in MATLAB. Other alternative languages, as outlined below, were ruled out:

- Java: after contacting Learned-Miller directly, it was concluded that there was no GlsCongealing algorithm demo code programmed in Java. The author did not want to further increase the workload to create a Java implementation as this could put the project at risk of non-completion.
- C++: the author decided not to persue using a C++ implementation of the Congealing algorithm, as in [25] due to lack of experience in the language.

MATLAB offers a lot of built-in packages designed to alleviate the more mundane implementation tasks, such as reading in images (function `imread`) and applying functions to every item in a matrix (function `bsxfun`). This also leads to quicker run-times as MATLAB relies heavily on vectorisation of code (as outlined later in the document - Section 3.6), which reduces the time spent running `for` loops.

However, to use MATLAB as a student, a license must be purchased. This costs £29 + VAT as a stand-alone product, with additional Toolboxes costing an extra £16 each.

3.2.2 Version Control

Version control is an important tool in modern day software creation. It records changes to files (such as code or written documents), and allows the user to update versions, or rollback to a previous one. In teams this is vital due to developers often working on the same, or similar, pieces of code simultaneously.

The tools utilised for version control in this project were Git [32], Github online [30] and Github desktop tool [31].

Git

Git [32] is one of the most popular in terms of Software Configuration Management (SCM) tools. It is a command-line tool which allows you to work on sections of code completely independent of each other and later merge them back into one complete article. Due to being written in C, Git is extremely quick compared to its rivals, being up to 325 times faster than Apache Subversion (SVN) [29].

Github online

Github [30] - `github.com` - is an online hosting service for Git repositories. It allows the user to clearly see all the files in the repository, make minor changes via an online editor, and to easily track features such as Issues and Feature requests. By offering both public and private repositories, Github allows developers to work on new, incomplete projects that only they, or their team, can see, or to open-source their software and make it freely available to the world.

Github also allows visualisations of the commit history to the repository, as demonstrated in Figure 3.2.

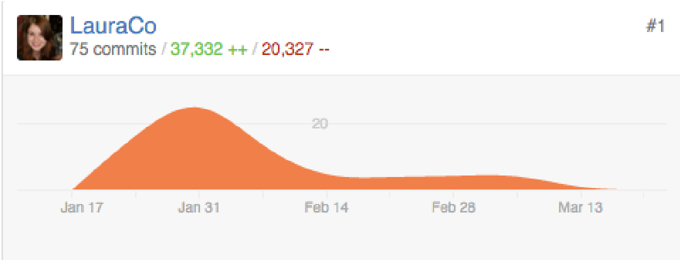


Figure 3.2: Graph outlining git commit history in one branch during the project.

Github desktop

Github desktop [31] is a simple tool which leverages the power of both Git and Github. It allows the user to run commands usually executed via the command line in a GUI and provides a graphical representation of the repository being worked in. This reduces the complexity of using branches, as the user can simply compare their current branch against others, and pull, push, merge and create pull requests as desired.

Due to it being created by Github, instead of a third party, this ensures that the latest version of Git is always implemented, and it seemlessly links in with the online project repository.

3.3 Dataset

Whilst mammographic images are the dataset of choice, this project could be applied to any greyscale image set, such as other medical imagery. The choice to use mammographic images was a personal one due to family history and an interest in aiding medicine via computer science techniques.

There exists several open mammogram datasets, so there was no issues in obtaining data without ethical concerns. The ethics form completed for the University can be found in Appendix B. This section will outline the main open datasets considered for this project.

3.3.1 Mammographic Image Analysis Society (MIAS) database

The chosen dataset is a version of the Mammographic Image Analysis Society (MIAS)'s database, as it is commonly used within the research field and compiled for the sole purpose of trying to better understand mammograms. The original MIAS database has been refined to create the Mini-MIAS database which contains the same data, however with a size of 1024x1024 pixels. This size is preferable over the original MIAS data, as it is a lot quicker to process.

Examples of Mini-MIAS scans can be found throughout the document, however for reference, an example can be seen in Figure 3.3

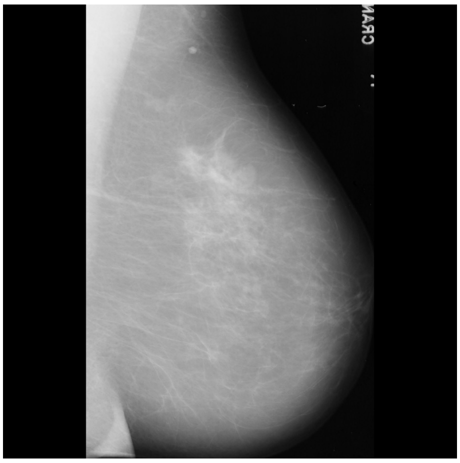


Figure 3.3: Example Mini-MIAS scan, scaled down for inclusion in document.

3.3.2 Other datasets

Digital Database for Screening Mammography (DDSM)

The DDSM database was created after a collobration between Massachusetts General Hospital, Sandia National Laboratories and the University of South Florida Computer Science and Engineering Department and contains around 2,500 scans [36] [37]. However, like the original MIAS database, the images available via the DDSM are extremely large, and therefore unsuitable for this project due to the time it would take to process.

Mammographic Mass Data Set

As the name suggests, this dataset contains 961 instances of scans containing masses - both benign and malignant [27]. This project aims to work with solely healthy tissue, therefore this dataset is not beneficial.

3.4 Algorithm Implementation

This section will provide an outline to the implementation of both of the fuzzy entropy algorithms including the fuzzy-set membership implementation and a brief outline of the Shannon entropy implementation provided in the demo code by Learned-Miller [42].

More information on the new functions created for this project, along with functions modified and utilised from the existing code base can be found in Appendix C.

3.4.1 Membership Implementation

As covered in Section 3.1.1, the Membership trapezia would be defined within the system, rather than dynamically being calculated. This made use of the Fuzzy Logic Toolbox [50] by leveraging the following functions:

- **trapmf:** trapezium-shaped membership function
- **evalmf:** a generic membership evaluation function

Having pre-defined functions for creating and evaluating the membership trapezia ensured that the function was a quick implementation. However it did reduce the number of Unit Tests that could be written to test the creation of such elements.

In this function, an image is read in, the membership of each pixel is evaluated against each of the two or three membership trapezia, adding the outcome to a corresponding array (e.g. one array for the membership of all the pixels against the low-trapezium etc). The two or three arrays (depending on the number of trapezia) are then compared and the element at position x,y with the greatest membership value is added to another array which collates all of the highest values.

For example:

```
pixel[1,1]'s membership in trapezium low = 0.8
pixel[1,1]'s membership in trapezium medium = 0.4
pixel[1,1]'s membership in trapezium high = 0

Therefore as 0.8 is the highest membership value, this is added to
  ↪ the image membership array
```

The three/four arrays (one for each trapezium and one for the highest values) are then passed out of the function, and utilised in both of the fuzzy entropy algorithm functions.

3.4.2 Shannon Entropy

Learned-Miller's demo code came with an implementation of Shannon Entropy [42]. Whilst originally the predominant dataset was MNIST handwriting data [43], which is a binary dataset (pixels were either black or white), this was not useful for this project as there is no variation in greyscale, even when a mean is taken of multiple images.

However, Learned-Miller had included extra code to handle the processing of greyscale and colour images, such as in MRI images. This ensured that no function needed to be created to handle greyscale images, greatly reducing the pre-programming needed. The only image handling that was encountered was to do with the mammograms, and the creation of a large pgm file to pass into the Congealing algorithm.

As outlined in 2.2.1, Shannon entropy is defined as:

$$H(X) = -\sum_{i=0}^{N} p_i \log_2 p_i \tag{1}$$

3.4.2.1 MATLAB implementation

This can be computed very quickly using a lookup table containing all the possible values of p. This makes it likely that the Shannon entropy algorithm will be the quickest on each iteration. However as it does not take any type of uncertainty into consideration when aligning the scans, the outcome could be quite dramatically different from that of a Fuzzy entropy nature.

Learned-Miller's Shannon entropy implementation has been retained, and can be found in the function `fastEntLookup.m` which is how it was originally named. Where possible, when original functions have been used, their original file names have been kept the same.

The function itself needed no changes to fit in with this project, however the way in which it is called by both `binaryCongeal.m` and `incrTrans.m` has been adapted so the user can choose which alignment metric to use when congealing their images.

3.4.3 Non-Probabilistic Entropy

As outlined in Section 2.2.3.1, De Luca & Termini‘s Non-Probabilistic entropy can be defined as:

$$H_A = -K \sum_{i=1}^{n} \{\mu_i \log(\mu_i) + (1 - \mu_i) \log(1 - \mu_i)\} \tag{2}$$

We will assume $-K$, the positive constant, is defined as $\frac{1}{n}$ as outlined in [24].

3.4.3.1 MATLAB implementation

After some research into current implementations of Fuzzy Entropy algorithms in MATLAB, it was concluded the best approach would be to implement De-Luca & Termini’s algorithm from scratch. This entailed creating a membership function, which computes the grey-level membership of each pixel in the mean image (calculated from a set of input images).

This array of pixel memberships is fed into a `nonProbabilistic.m` function where it is iteratively passed into latter part of equation 2 $\left(\text{after} \sum\right)$. The output array is then summed and multiplied by $\frac{1}{n}$ as defined in Equation 2 and Subsection 2.2.3.1. The final mean pixel entropy is calculated by taking the image entropy and dividing by the number of pixels in the image.

3.4.3.2 Technical challenges

The main technical challenge for this implementation is ensuring maximum optimisation to keep running times to a minimum. Leveraging MATLAB’s own Toolbox for calculating the membership saves a lot of time and lines of code, however it was important to check what they call from within. One membership function was redrawing the trapezia every time it was called, significantly slowing down the process - reducing the amount of times the initial function was called helped reduced the run-time by over 60seconds. This challenge of optimisation is covered in-depth in the latter Subsection 3.6.4.

3.4.4 Hybrid Entropy

As mentioned in Section 2.2.3.4, the Hybrid Entropy equation is as follows:

$$H_{hy} = -p_0 \log(1 - E_0) - p_1 \log(E_1) \tag{3}$$

Where E_0 and E_1 can be defined as:

$$E_0 = \frac{1}{n} \sum_{i=1}^{n} (1 - \mu_i) exp(\mu_i) \tag{4a}$$

$$E_1 = \frac{1}{n} \sum_{i=1}^{n} \mu_i exp(1 - \mu_i) \tag{4b}$$

And p_0 and p_1 are the probabilities of receiving 0 and 1 symbols respectively.

3.4.4.1 MATLAB implementation

Due to reasons covered in Subsection 3.4.4.2, Hybrid Entropy membership was implemented using 2 trapezia covering 2 fuzzy sets, as seen in Figure 3.4.

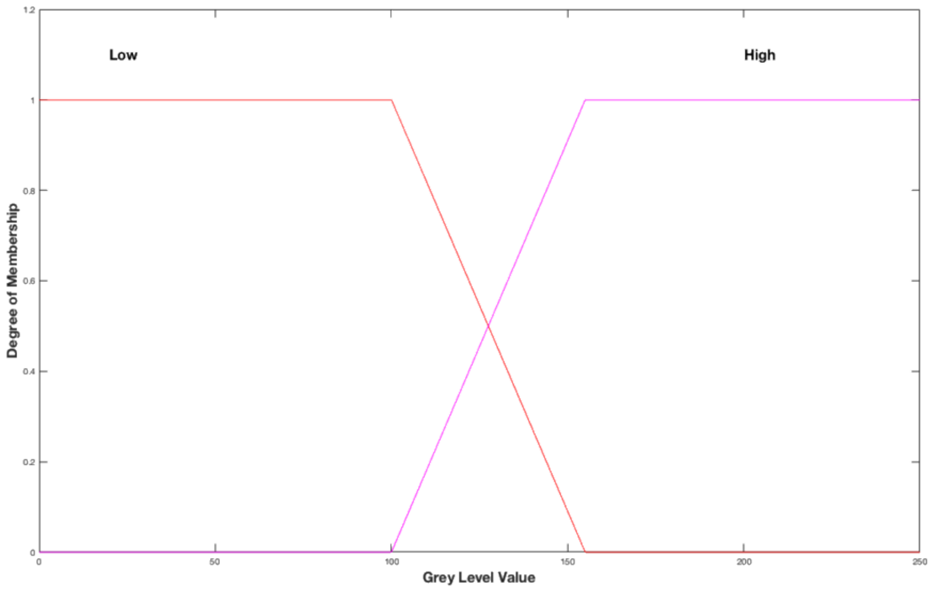


Figure 3.4: Two membership trapezia for Hybrid Entropy - Low and High grey-level values.

Two arrays are then fed into the Hybrid Entropy function - one listing all the pixel membership values from the low trapezium, and the other from the high trapezium. The final entropy is taken as a comparison between the low and high fuzzy sets.

3.4.4.2 Technical challenges

Whilst Hybrid Entropy utilises a membership function, much like Non-Probabilistic entropy, it was derived to work with binary entropy, not the ternary membership modeled for Non-Probabilistic. Because of the binary nature, the equation uses 'inversion' to depict if not this fuzzy set, then must belong to the other.

Experimentation was done as to whether the equation could be adapted in such a way to continue using three separate membership trapezia - low, medium and high grey-level values. Logic would dictate that if the comparison of two fuzzy sets works, then to compare the low fuzzy set to the medium, the medium to the high and the high to the medium should work.

In theory, calculating E_0 and E_1 for each trapezium, calculating the hybrid entropy for each, and then combining them, should work:

$$E_0 = \frac{1}{\text{No. of pixels in low trapezium}} \sum_{i=1}^{n} (1 - Low\mu_i)exp(Low\mu_i) \tag{5}$$

$$E_1 = \frac{1}{\text{No. of pixels in low trapezium}} \sum_{i=1}^{n} Low\mu_i exp(1 - Low\mu_i) \tag{6}$$

Where $Low\mu$ is the membership of the pixels in the low fuzzy set.

$$H_{hy} = -p_0 \log_{10}(1 - E_0) - p_1 \log_{10}(E_1) \tag{7}$$

Where

$p_0 = \frac{\text{No. of pixels in low trapezium}}{\text{No. of pixels in low trapezium + med trapezium}}$

and

$p_1 = \frac{\text{No. of pixels in med trapezium}}{\text{No. of pixels in low trapezium + med trapezium}}$

This was done for all 3 trapezia, then combined and divided by 3 (for the mean entropy). As the result for each trapezium should be between 0 and 1 (as each is an entropy value), then combining them should be no issue. However this was not the case.

First of all, the hybrid equation output was deemed to be 'NaN' - something which generally occurs when attempting to divide by 0. Anomalous outputs from the high trapezium was to be expected, as there are very few pixels which fall within the range nearer the white end of the grey-level scale. This was mitigated by setting any 'Nan' output equal to 0, in effect ignoring that particular output from the highest fuzzy set.

After this mitigation, the third and fourth iteration had suitable entropy values, however the fifth entropy value was a negative, something which is not possible in terms of entropy, as it must be between 0 and 1 - see Figure 3.5.

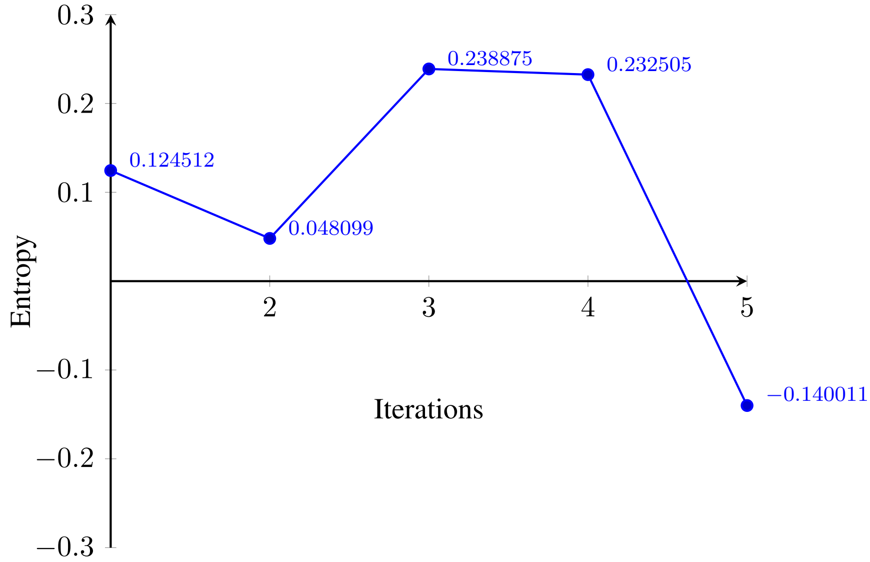


Figure 3.5: Graph showing the entropy output after 5 iterations

It was concluded that the implementation of three fuzzy sets within Hybrid Entropy would not be realistic within the remaining time-frame of the project, and the membership for Hybrid Entropy was redefined to the concept of 2 fuzzy sets, as derived by Pal and Pal. This would mean one trapezium for pixel grey-level values with low values, overlapping with a high grey-level value trapezium at approximately 128, as seen in Figure 3.4.

3.5 Software Implementation

3.5.1 Methodology

In the past, software projects followed a strict-plan driven approach, such as the Waterfall method, however more recently, Agile practices have become widely accepted, allowing the developer more freedom. This features an iterative development approach, with short "iterations" or "sprints" defined in which the developer should complete a block of work, typically a "story" or "feature" given the Agile methodology chosen.

The Agile Methodology has a manifesto [9], which encompasses all the values it strives to achieve:

- ***Individuals and interactions*** *over processes and tools*
- ***Working software*** *over comprehensive documentation*
- ***Customer collaboration*** *over contract negotiation*
- ***Responding to change*** *over following a plan*

> *That is, while there is value in the items on the right, we value the items on the left more.*

Scrum is one of the most popular interpretations of an Agile Methodology, due to it's simplicity [2]. Scrum is *not* an agile methodology, however is a framework, to which agile practices such as Pair Programming and Test Driven Development (TDD) can be aligned.

An adapted Scrum methodology has been undertaken for this project. The flexible nature is particularly useful given the research nature of this project, as the requirements were not fully defined at the start of the process, and changed as time went on given the outcome of experimentation with mathematical concepts for image alignment.

Additionally, eXtreme Programming (XP) [81] dates back to 1996, and is one of the most recognisable Agile Methodologies used in the software industry currently. XP claims to create successful software projects by following 5 key principles:

- **Communication:** constantly communicate with their customers and fellow programmers
- **Simplicity:** keep the design simple and clean
- **Feedback:** testing the software starting on day one
- **Respect:** every small success deepens their respect for the unique contributions of each and every team member
- **Courage:** deliver the system to the customers as early as possible and implement changes as suggested

Given that this project is a single-person project, neither framework/methodology would work well on it's own, so for this project, it was decided that Scrum would be the main framework, with elements of XP to help strengthen areas such as design and testing.

3.5.1.1 Tools to manage methodology

This project has been chiefly supported by the application `taiga.io` - a beta web app [11], which aims to promote the use of Scrum and Kanban [64].

Having an online application to organise User Stories, Tasks, Issues and to track progress using a Burndown chart was extremely important in this single-person project, where work was carried out across several different devices and platforms. It also ensures a historical record of what was completed, and when, as is evident from Subsubsection 3.5.1.2.

3.5.1.2 User stories

User Stories are a bid to shift away from talking in technical jargon, and to shift towards talking in plain english about project requirements. When working with a customer, this is obviously useful, as occasionally they are non-technical, so this helps promote an open-dialogue between customer and developer, and a clear understanding of the customer's needs.

User Stories typically follow a template for consistency, usually something similar to:

As a <type of user>, I want <some goal> so that <some reason>. [19]

User Stories also have associated Story Points, which is a typically a numbering system leveraged to indicate the effort needed to implement the Story. Due to the uncertain nature of programming, it is not always an accurate reflection of effort, however through the Agile community it is generally accepted that to be consistent in your assignment of points is more useful than being accurate [72]. During the early stages of the project, it is often the case in which estimation is a little off what it should be, however as the project progresses, and the developer gains a better understanding of the tasks, and how to implement them, then estimation tends to become more accurate.

Table 3.2 outlines the User Stories used during this project, along with when they were worked upon (during which Sprint) and how many Story Points were associated with it.

Reference	**User Story**	**Milestone**	**Story Points**	**Additional Comments**
1	Clinicians can upload a set of images (MATLAB Command Window) so they can control what images are input into the Congealing Algorithm	Sprint 0	5	Initially this was hard-programmed
2	Developer will implement membership of a pixel so that Fuzzy Entropy can be calculated	Sprint 1	10	
3	Clinicians can align scans using Non-Probabilistic Entropy so it can be used in the Congealing Algorithm	Sprint 2 & 3	20	Due to complexity of the implementation, this was spread over 2 sprints

Reference	User Story	Milestone	Story Points	Additional Comments
4	Clinicians can select an alignment metric (MATLAB Command Window) so they can select which to align the images using	Sprint 4	5	
5	Developer will make standard GUI with no functionality so that this can be demonstrated as a proof of concept	Sprint 4	5	
6	Clinician can choose number of iterations (MATLAB Command Window) so they can run as many as they want to	Sprint 4	3	
7	Developer will implement basic mammogram upload so that they can be aligned	Sprint 4	8	
8	Clinicians can align scans using Shannon entropy so it can be used in the Congealing Algorithm	Sprint 5	8	
9	Clinicians can upload a set of images - GUI	Sprint 5	10	
10	Developer will optimise membership function so to improve performance	Sprint 5	2	Promoted from an Issue
11	Developer will optimise Non-Probabilistic function so to improve performance	Sprint 5	2	Promoted from an Issue
12	Clinicians can clear an input image so that they can reselect an input image - GUI	Sprint 5	3	
13	Clinicians can align scans using Hybrid Entropy so it can be used in the Congealing Algorithm	Sprint 6	20	
14	Clinicians can select an alignment metric from a drop-down menu so it is easy to choose which alignment metric to use - GUI	Sprint 6	5	
15	Clinicians can select the number of iterations to be run using an alignment metric (GUI) so it is easy to select how many iterations to run	Sprint 6	5	

Reference	User Story	Milestone	Story Points	Additional Comments
16	Clinicians can see metadata about the input image so they can see if the uploaded image is the correct one - GUI	Sprint 6	2	
17	Clinicians can see each iteration mean image so they can compare the improvement over each iteration - GUI	Sprint 6	3	
18	Clinicians can see adjusted input images on final iteration so they can see how the input images have changed by the final iteration - GUI	Sprint 6	3	
19	Developer wants to know why Scans are rotated 90 to left as this is aesthetically displeasing	Sprint 7	8	Promoted from an Issue
20	Developer will research and implement removal of Medical Markers as this causes alignment issues	Sprint 7	5	
21	Clinicians can discard (clear) an alignment so they can start a new alignment - GUI	Sprint 8	5	
22	Clinicians can click on average image to view it bigger so they can see the detail easier - GUI	Sprint 8	2	
23	Clinicians can save the final mean image with a sensible name so they can easily find it again - GUI	Sprint 8	3	
24	Clinicians can see the iteration details so they can understand more about the improvement - GUI	Sprint 8	8	
25	Clinicians can see Congealing is running so they know it's in progress - GUI	Sprint 8	3	

Table 3.2: User stories defined during the project timeline.

3.5.1.3 Burndown chart

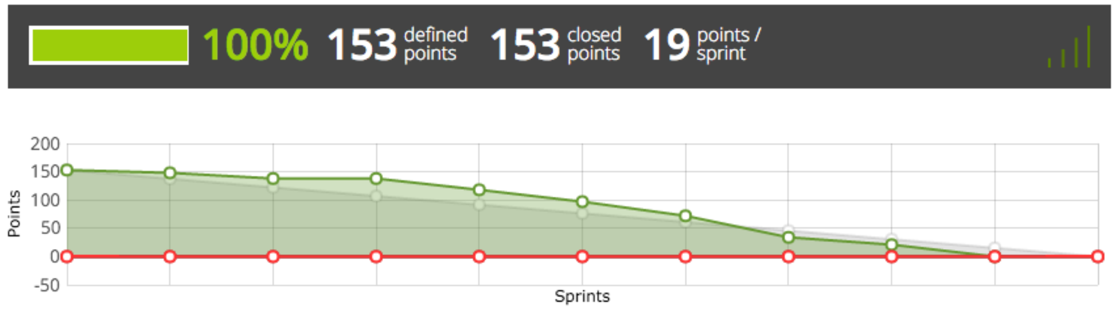


Figure 3.6: Project Burndown chart.

Figure 3.6 is a graphical representation of progress per week, as utilised in Scrum, called a Burndown chart. It allows the developer(s) a quick reference as to the progress of the project, and works by subtracting completed Story points as they're completed. Taiga includes the trend line which sets a target for completion per Sprint.

In Taiga, it was also possible to have a weekly burndown chart, so as to track progress throughout the week, rather than just the entire project. This ensured a steady rate of development through the week between Supervisor meetings, rather than over-working at the beginning of the week and having nothing left to do at the end, or vice-versa.

3.5.1.4 Sprint Review & Retrospective

Sprint Review meetings are held at the end of each Sprint, to assess what work was done during the week, and does the end product match the Sprint Goal set out at the start of the week. In this project, Sprint Goals, Sprint Reviews and Retrospectives took shape in the form of an informal online blog. Sprints were defined as a week long in this project, running between supervisor meetings (Monday - Sunday). Weekly posts would outline what had been completed that week, how things went (good and bad) and what was to be completed during the following week. Whilst less structured than the conventional approach to Reviews, it works well within a single-person project, and was a good reflection of what had been accomplished.

In Agile Methodologies, Retrospectives are typically at the end of each Sprint, so the team can assess:

- What works well
- What doesn't work well
- What should they start doing

This ensured good practices were carried forward into the next Sprint, whereas unhelpful ideas were left in the previous week.

3.5.1.5 Daily Standup

Daily standups are a vital part of Scrum‘s teamwork ethos. Each morning (or during a set allotted time), the team would meet to discuss what was accomplished the day before, what are the plans for the day ahead, and what road-blocks are in their way. This provides the developer (and further the team) a clear picture of what has yet to be done, and allows fellow team-mates to offer expertise to help overcome obstacles. Whilst this project is not being developed by a team, the benefit of daily standups to productivity, organisation and planning still stands, along with the crowd-sourcing element of expertise.

Throughout the project, stand ups have been held with peers, who’re also working upon their Major Projects. Whilst not daily, they tended to fall every two days, and it gave the developer a chance to hone skills in explaining the project to people not well-versed in the subject. It was also a good breeding-ground for new ideas, and an open forum for discussion into the pros and cons of certain approaches.

3.5.2 Design

In traditional plan-driven methodologies, such as the Waterfall method, Design would take shape in the form of a design document where all the requirements would be outlined and written up in detail. As mentioned previously, this would be impractical for such a fluid, experimental project so practices were leveraged from eXtreme Programming (XP) to ensure that the system design was not compromised by the lack of early, solid requirements.

3.5.2.1 CRC Cards

In XP, Class, Responsibilities, and Collaboration (CRC) Cards are a useful task in which the entire team can collaborate on the system design. Whilst there is no team in this project, they still play a vital role in structuring the system, can be iteratively updated and are easily discardable should the need arise.

Typically CRC cards would represent Objects, with the class written at the top, the responsibilities down the left and the collaborating classes down the right-hand side. However as mentioned in Section 3.2.1, MATLAB is built around a scripting language, and all the “Classes” in this project are replaced by Functions and Scripts. Therefore, each CRC card represents a function or a script, and it’s corresponding responsibilities and collaborations as normal - see Figure 3.7 for more detail and Appendix D for the CRC card iterations throughout the project.

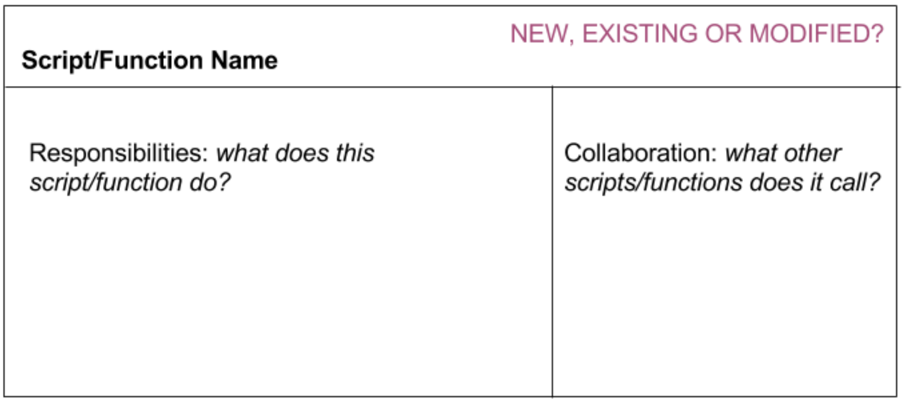


Figure 3.7: Example CRC Card as used in this project

3.5.2.2 GUI Design

The name "Enantiomorph" was chosen as the application name as a more concise, recognisable alternative to the project title.

> *Enantiomorph: either of a pair of crystals (as of quartz) that are structural mirror images [4]*

This section will look at the design evolution of the application GUI.

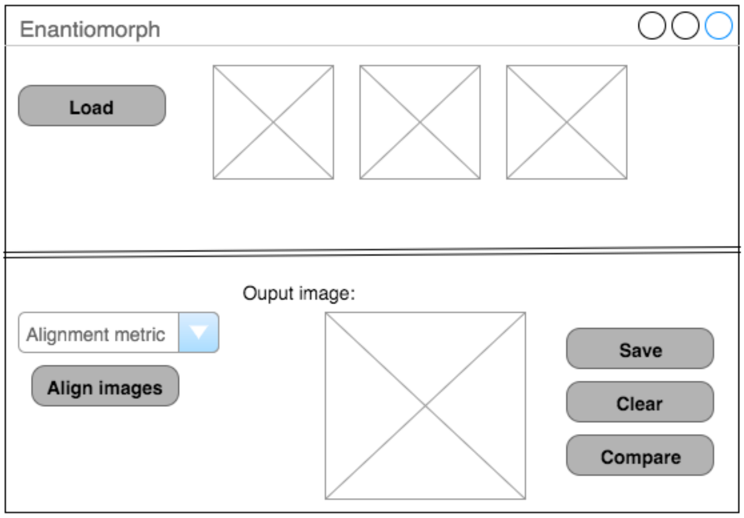


Figure 3.8: Initial wireframe design for GUI

The initial design, as represented in Figure 3.8, was designed to incorporate the first set of project requirements.

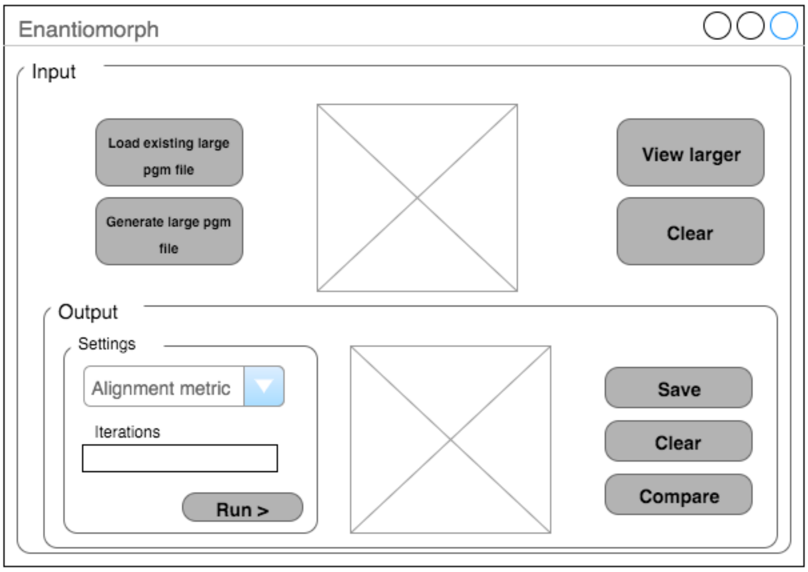


Figure 3.9: Second wireframe design for GUI

Figure 3.9 represents the changes in requirements as the project progressed. It became clear that a load button which allowed the user to *both* generate a large pgm file from a folder of mammograms, or upload a large pgm file that already exists would be difficult to implement. Therefore the button was split into two buttons, with appropriate text above the buttons to help the user decide which to use.

By the second GUI iteration, it became apparent that implementing a way in which to stop the Congealing algorithm automatically would be too time-consuming for the time left in the project. Therefore the user would have to specify how many iterations they would like to run. This meant a textbox with numerical validation had to be incorporated into the GUI and the extra iteration information fed into the back-end.

During the second application iteration, the outputs of each iteration mean and the adjusted input images were implemented for the user to see.

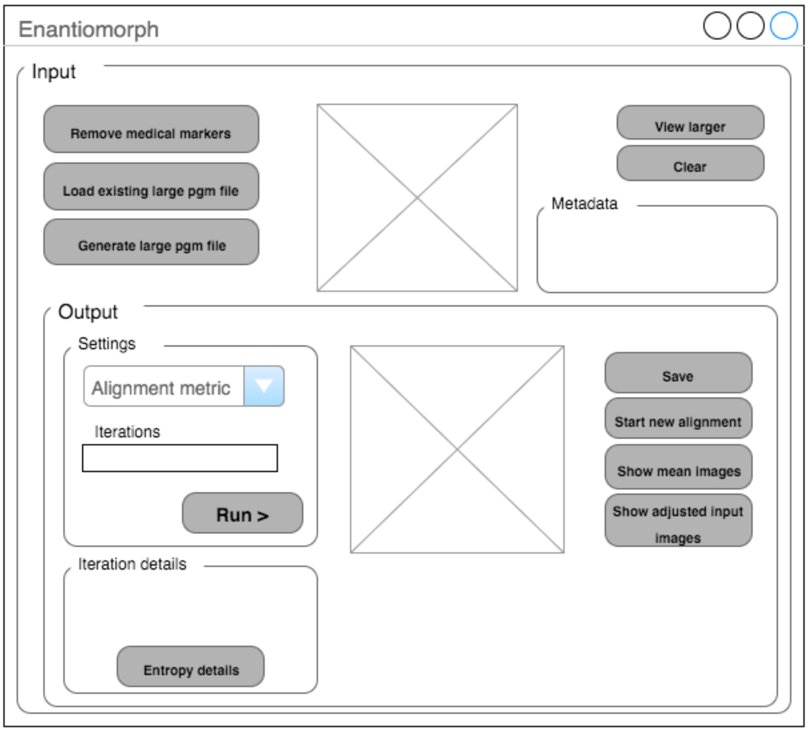


Figure 3.10: Third wireframe design for GUI

The final wireframe created is outlined in Figure 3.10. Additional information about the Congealing process can be accessed via the button in the bottom left corner and metadata about the input image displayed in the top section. Users can also clear the entire GUI to start a new alignment - this could be useful should they wish to compare the outputs from the 3 different entropy alignment techniques.

The final GUI can be found in Subsection 3.5.3, Figure 3.11.

3.5.3 GUI Implementation

Initially the Graphical User Interface (GUI) was to be implemented using JavaFX [59], and any MATLAB additions would be linked in, however it became apparent that it is possible to create GUIs easily within MATLAB itself. GUIDE [52] is MATLAB‘s Application development environment where you can either build using purely drag-and-drop techniques, program the application like normal in the editor, or both.

The combination of both the drag-and-drop environment, and manually programming via the editor was undertaken during this project, to allow a greater amount of freedom and flexibility. Drag-and-drop was leveraged to style the GUI and the editor was used to program the functionality in the back-end and link in the Congealing and Fuzzy Entropy algorithms.

The design process, including wireframes, of the GUI can be seen in the earlier Subsection 3.5.2.2.

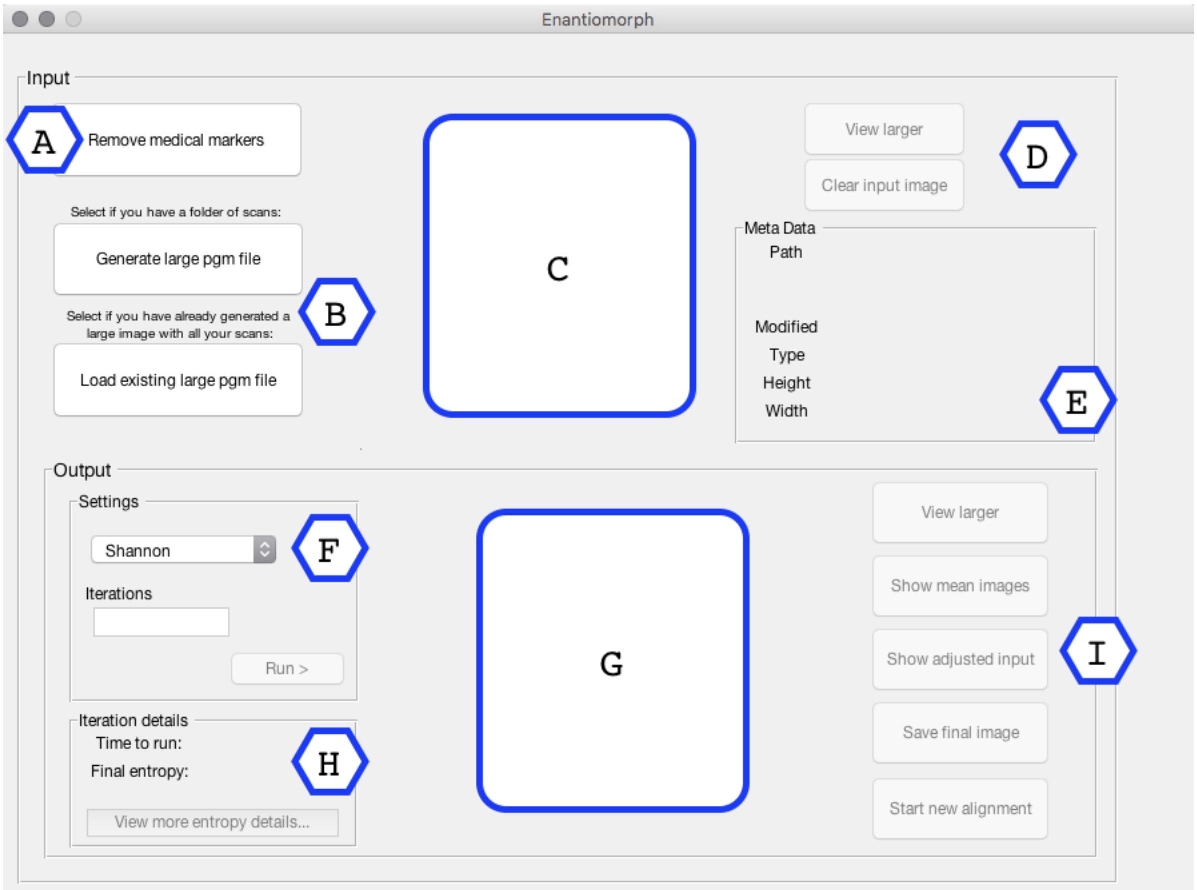


Figure 3.11: The main Graphical User Interface (GUI)

Figure 3.11 gives a snapshot of the final application implemented for users to align their images.

GUI breakdown:

A: Users can remove medical markers (or any other artifacts) from their mammographic images prior to creating the image to be congealed. This in itself is a separate GUI, which will be covered later in the Section.

B: After removing medical markers, the user can go on to 'generate' a large pgm file, which will contain all the mammographic images they wish to align. This will then proceed to load the large pgm into the application. If the user has already generated one of these images, they can simply choose to load it instead.

C: Once the image is loaded in, it will be displayed here.

D: Should they wish, the user can view their input image larger, or clear the image should they upload the wrong file.

E: Metadata about the image is displayed here.

F: The user can select the image alignment metric from the drop-down menu, and enter the number of iterations they wish to perform. Pressing the 'Run' button will start the Congealing algorithm, and the user will see an egg-timer/pinwheel to signify it is running.

G: Once the images have been aligned, the final average image will be displayed here.

H: Information about how long the congealing process took, and the final entropy value will be displayed here. Users can choose to view a graph detailing the reduction in entropy over each iteration using the 'View more entropy details...' button should they wish.

I: These buttons allow the user to choose what to do next. They can either view the final mean image larger, view the mean images for each iteration run, see how the input images have adjusted to fit in the final mean image, save the final mean image or clear the application ready to run a new alignment.

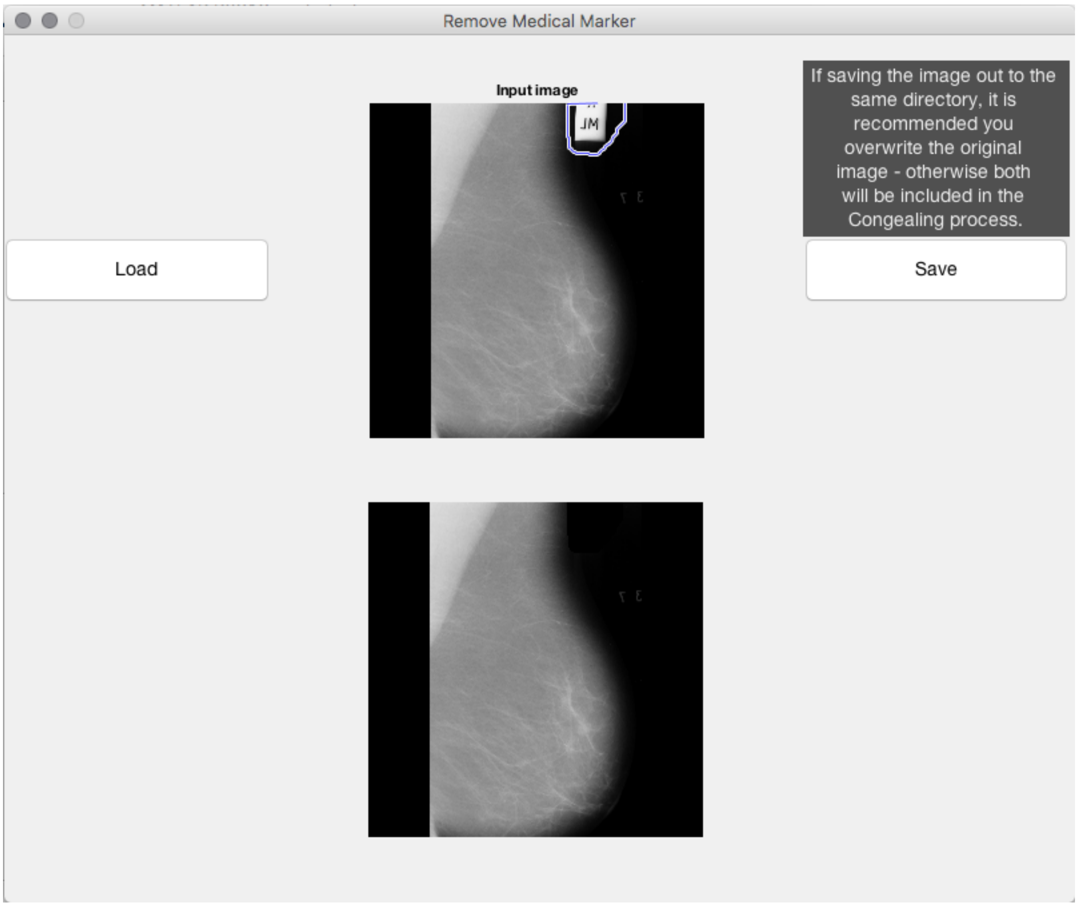


Figure 3.12: The Graphical User Interface (GUI) for removing medical markers

Figure 3.12 details the GUI in which the user can remove medical markers/other artifacts. This simple user interface allows them to simply:

- Load in the image of their choice
- A pop-up (not pictured) gives them instructions on how to draw on the image
- The user then draws around the area they would like to remove
- The final image is displayed in the second image
- The user can save the image back out, overwriting the original if they wish

Once all the unnecessary markers or artifacts have been removed, the user can close the GUI and be returned to the main application.

3.5.4 Function calls

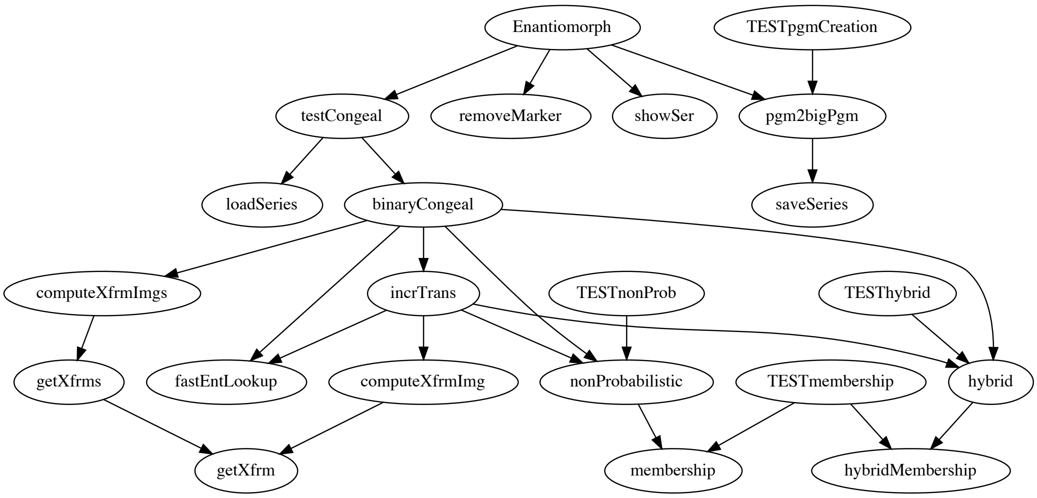


Figure 3.13: Function calls through the application.

Figure 3.13 shows which Function/Script calls which through the application, all the way up to `Enantiomorph`, the GUI. This diagram shows the integration of new, modified and existing functions/scripts in the code-base, as listed in Appendix C.

Diagrams such as this were helpful especially at the beginning of the project, due to working with an existing code-base. It was advantageous to see which functions were directly being called, and where the new functions for fuzzy entropy and membership would fit in.

3.5.5 Testing

Prior to the project beginning, it was outlined that this project would follow a TDD practice. In TDD, the Developer first writes a test, which will fail due to the lack of corresponding functionality. They would then go on to implement the functionality desired by the test. Finally, any refactoring of the initial test and/or code would take place.

However due to the nature of the project, it became increasingly more difficult to follow given the research which had to be undertaken alongside development. This led to a change from TDD to Retrospective Testing, all tests would be written post-functional-implementation. This is a more traditional approach to testing, and still catches the same errors which might occur during TDD.

3.5.5.1 Unit Tests

Unit Tests were completed using MATLAB's Unit Testing Framework [74], which covers all the ways in which you can program in MATLAB:

- Script-Based Unit Tests
- Function-Based Unit Tests
- Class-Based Unit Tests

The majority of my work in MATLAB was function-based, so this was the style followed for unit tests. As Figure 3.14 demonstrates, all Unit tests passed.

1 Name	2 Passed	3 Failed	4 Incomplete	5 Duration
'TESTnonProb/testNonProbEntropyLessThanOne'	1	0	0	1.8517
'TESTnonProb/testNonProbEntropyLargerThanZero'	1	0	0	1.8428
'TESThybrid/testHybridEntropyLessThanOne'	1	0	0	0.1140
'TESThybrid/testHybridEntropyLargerThanZero'	1	0	0	0.1190
'TESTpgmCreation/testImageType'	1	0	0	0.6913
'TESTpgmCreation/testCommentInsertion'	1	0	0	0.5880
'TESTpgmCreation/testImageSize'	1	0	0	0.6043
'TESTpgmCreation/testGreyScale'	1	0	0	0.5877
'TESTmembership/test3TrapeziumMembershipDegreeNotEmpty'	1	0	0	0.0828
'TESTmembership/testCalulatedMembershipDegreeForAllPixels3Trapeziums'	1	0	0	0.0841
'TESTmembership/test2TrapeziumMembershipDegreeNotEmpty'	1	0	0	0.0731
'TESTmembership/testCalulatedMembershipDegreeForAllPixels2Trapeziums'	1	0	0	0.0748
'TESTmembership/test3TrapeziumMembershipDegreeLessThanOne'	1	0	0	0.0810
'TESTmembership/test2TrapeziumMembershipDegreeLessThanOne'	1	0	0	0.0799
'TESTmembership/test3TrapeziumMembershipDegreeGreaterThanZero'	1	0	0	0.0798
'TESTmembership/test2TrapeziumMembershipDegreeGreaterThanZero'	1	0	0	0.0734

Figure 3.14: Results from MATLAB Unit Tests.

3.5.5.2 Acceptance Tests

Acceptance tests are when each 'requirement' is assessed in turn to assure its completion. In this project, as there was no firm requirements at the beginning of the process, the User Stories which were derived over the project duration will be assessed against. In eXtreme Programming (XP), a User Story is not considered to be complete until the time in which it passes it's acceptance test, as stating on the XP website [82].

Table 3.3 outlines the results from the Acceptance Tests run. The left-most column "User Story Reference" aligns with Table 3.2, where more detail on each Story can be found.

User Story Reference	**Expected Outcome**	**Actual Outcome**	**Pass/Fail**
1	Image is loaded into the system	As expected	Pass
2	Membership array is passed out of the membership function and is usable in other functions	As expected	Pass
3	Images are aligned using Non-Probabilistic entropy and the output & entropy outputs are realistic	As expected	Pass
4	Images are aligned using the metric the user has selected when running the function	As expected	Pass

User Story Reference	**Expected Outcome**	**Actual Outcome**	**Pass/Fail**
6	The number of iterations is run as specified by the User, then function stops	As expected	Pass
8	Images are aligned using Shannon entropy and the output & entropy outputs are realistic	As expected	Pass
9	Image(s) selected to be loaded into the GUI is displayed	As expected	Pass
12	Image box where input image appears goes blank after Clear button is selected	As expected	Pass
13	Images are aligned using Hybrid entropy and the output & entropy outputs are realistic	As expected	Pass
14	Images are aligned using the chosen alignment metric	As expected	Pass
15	The number of iterations is run as specified by the User, then function stops	As expected	Pass
16	When an input image is loaded in, Metadata is displayed in the GUI about the image	As expected	Pass
17	After Congealing, the user can press the “See all Mean images” button and a new Figure displays the mean image after each iteration	As expected	Pass
18	After Congealing, the user can press the “See Adjusted Inputs” button and a new Figure displays the adjusted input images after the final iteration	As expected	Pass
21	Image box where output image appears goes blank after Clear button is selected along with all other fields	As expected	Pass
22	Image is displayed larger in a new Figure	As expected	Pass
23	Save file dialog appears with a sensible name suggested (i.e. final image - alignment-chosen - number of iterations)	As expected	Pass
24	When “Entropy details” button is selected, a new Figure appears with a graph showing entropy decrease. Final Entropy & time taken also displays in the main GUI	As expected	Pass
25	Egg-timer appears when Congealing Algorithm is running	As expected	Pass

Table 3.3: Acceptance Test results

3.6 Technical Difficulties

3.6.1 Image Rotation

One issue which was faced when creating the large .pgm file containing all the input images, was that they were rotated 90°to the right, as demonstrated in Figure 3.15.

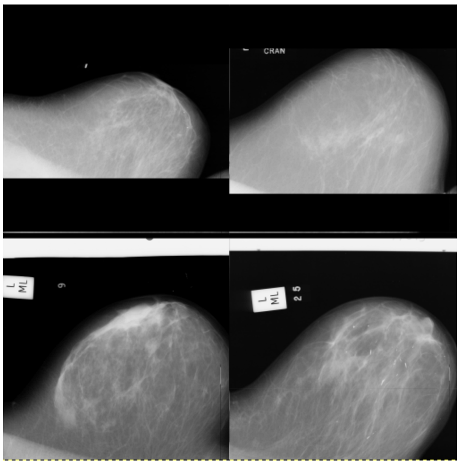


Figure 3.15: 4 rotated input images concatenated into one larger image.

It quickly became apparent that the order in which the image array was being written to file to create this larger pgm file was incorrect, however due to MATLAB's use of vectorisation, it was difficult to diagnose where the issue lay. After some investigation, it was revealed that the function `fwrite` by MATLAB [48], used for writing binary data to file, wrote each line out column-by-column, rather than the customary row-by-row approach.

To mitigate this issue, the array passed into fwrite would have to be transposed prior or during being passed into the function. There are two ways in which MATLAB permits the transposition of arrays:

3.6.1.1 Simple 2D array transposition

MATLAB has a "Transpose" function [56] which simply flips two elements in a 2D array as utilised in:

```
fwrite(output,handles.finalImg.','uchar');
```

Where handles.finalImg is a GUI holder for a 2D array of pixel values. This example was taken from the `removeMarker.m` function - where the user can remove Medical Markers and save the output back to the original file.

3.6.1.2 3D+ array transposition

For arrays with more than 2 dimensions, simply swapping the values around will not work, so the MATLAB function `permute` [54] must be used.

```
sers=zeros(squareImageSize(1),squareImageSize(2),noOfScans); %set size of array

for i = 1:noOfScans

  scan = fopen(strcat(pathname,'/',scanDirectory(i).name)); %open each input image individually
  im=(fread(scan,[squareImageSize(1),squareImageSize(2)],'uchar'));
  sers(:,:,i) = im; %add each input image to a 3D array which compiles all the input images into one

end

outfname=sprintf('%s/big_scan.pgm', pathname);
s=sers(:,:,:);
saveSeries(outfname,permute(s,[2,1,3])); %use the saveSeries demo function to write the final image arrays out to a file
```

This example was taken from the `pgm2bigPgm.m` function - where a set of input images are passed in, and a large pgm image containing all the input images is outputted (as in Figure 3.16). This image is then passed into the Congealing algorithm for alignment.

3.6.1.3 Final Outcome

After transposing all arrays which are to be saved out to file, whether directly through `fwrite` or via the `saveSeries` function, all images are saved in the correct orientation.

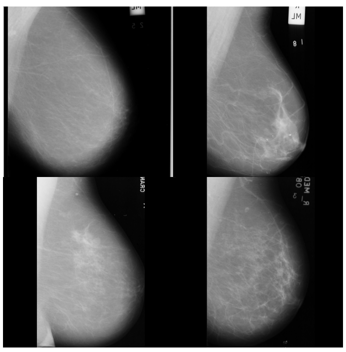


Figure 3.16: Final output image.

3.6.2 Medical Marker Removal

This subsection has been formalised from a blog post written by the author on 28th March 2016 [20].

As the images are aligned using a comparison of the pixel-value, the Medical Markers included on mammograms cause an issue. This is because if more than one scan contains these white patches (left by the metal clip during scanning), then they will try and align with each other during the Congealing process.

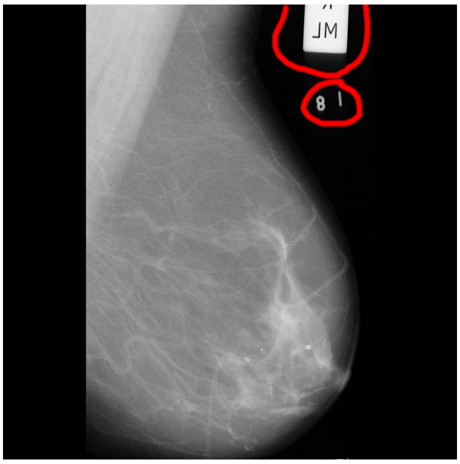
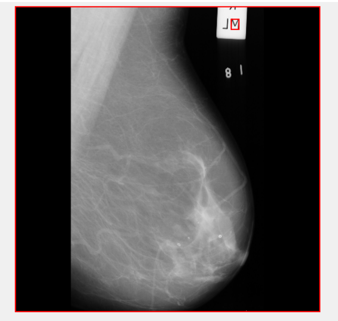


Figure 3.17: Image containing Medical Markers

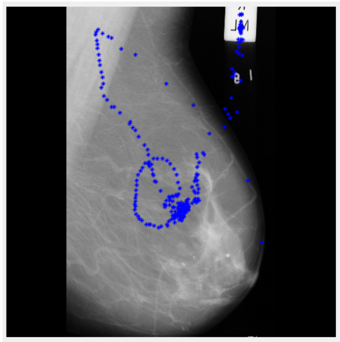
Two options were available for the avoidance of Medical Markers:

- Ask the User not to use scans containing Medical Markers
  - This is extremely restrictive
  - This could massively reduce their number of usable scans
- Find a computer vision and/or image processing technique to remove these clips
  - Preferably automatically
  - Manually removing would work for small input data sets

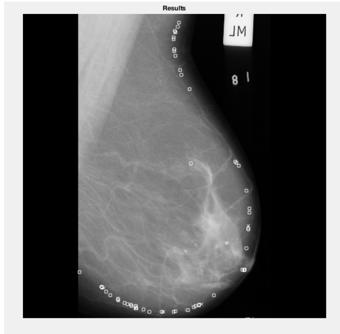
3.6.2.1 Discarded ideas



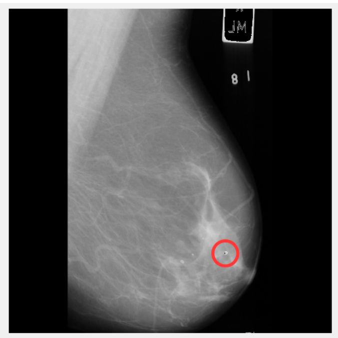
(a) Squares detected in a mammogram using morphological operations.



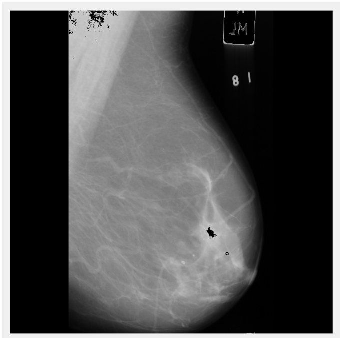
(b) Regionprops detecting areas and superimposing the area information on a mammogram.



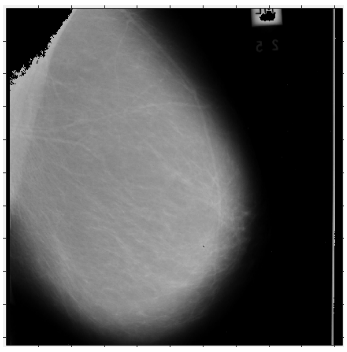
(c) Shape recognition picking out the rough boundary of breast tissue.



(d) Removing areas of grey-level value above 220.



(e) Lower grey-level removal threshold to 200.



(f) Applying removal over 220-value to another scan.

Figure 3.18: Output of discarded methods of marker removal

Morphological operation - remove squares from image

Utilising a morphological operation, such as the one demonstrated by Chandra Kurniawan in the thread [40]. However, as can be seen in Figure 3.18a, not only is the Marker itself not perfectly square, but because the image is square, it detects that instead. So removal would be made more difficult by the fact you would have to specify a maximum size square to remove, that smaller than the image size.

This idea was discarded due to the marker unlikely to ever be perfectly square in the scan.

MATLAB function regionprops

Another candidate function for removing Medical Markers was the MATLAB function `regionprops` [55]. The idea behind using this function would be to measure the area of the squares in the image, so then they could be removed. However, the output, as seen in Figure 3.18b, was not something desired, and without spending an inordinate amount of time tweaking the function, it is not useful to the detection of the markers.

Shape recognition demo

On the Mathworks File Exchange site, a community run to help MATLAB users, there was a demo created by Ahmed Samieh to aid in the recognition of certain shapes [67]. It classifies the shape by properties such as roundness, ratios of dimensions and centroids.

Modifying this demo slightly to make it compatible with the grey-scale mammograms, the output is somewhat promising, as seen in Figure 3.18c.

However, due to the slightly inaccurate identification of the tissue boundary, this is likely to remove data which is useful to the Congealing algorithm. Unless this can become a near perfect outline around the breast tissue, it is unlikely to be useful for selecting and focusing in the object of interest.

Removing white objects over a specified grey-level value

Returning to the Mathworks forum, there is a thread about removing white glare from a jewellery photo [12]. This was adapted to detect the medical marker by specifying to find and remove patches over 220 grey-level value. As seen in Figure 3.18d, most of the marker has been removed, however it also removes a small bit of breast tissue.

To see if the entire marker could be removed, if you lower the grey-level threshold for removal to anything over 200 value, then the output is as in Figure 3.18e. Unfortunately it does not remove the entire marker, and some of the vital breast tissue is lost.

Further to that, by running the white removal at grey-level value at 220 (the suitable choice for my first test scan) on another test scan and absolutely nothing is removed. Lower the threshold to begin removing white areas (down to grey-level value of 180) the results are less desirable, as demonstrated in Figure 3.18f.

3.6.2.2 Chosen method

Another demo on the MATLAB forum outlined a way in which a user can draw an area to remove, then a mask can be applied over the top to hide any problem areas [10].

After reading through the demo given as an answer by "Image Analyst" on the forum, the author rewrote and refactored some of the functionality in the provided code in order to fit the removal criteria. The User can utilise the MATLAB function `imfreehand` [53] to draw over the input image in order to indicate the area to be removed. This area is then filled in with the darkest grey-level value found in the drawn area (typically 0 for black, however may differ between scans).

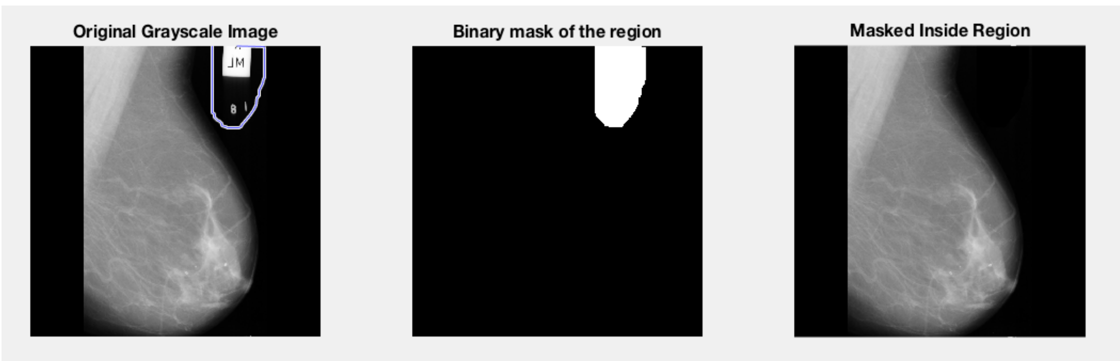


Figure 3.19: Image depicting the steps taken to remove Medical Markers from a scan.

As shown in Figure 3.19 this has been shown to be extremely successful and therefore was utilised in the project.

3.6.3 Creating the Mammogram image data

As this project was building upon the work done by Learned-Miller [42], it was useful to utilise the load function that was already in place. However, the nature in which the image files were loaded into the system caused some unexpected hurdles.

In order to understand how the demo data was uploaded, and therefore implement a function to compile a large set of mammogram scans in the correct format, research was carried out into the nature of PGM files, and the function which comments play in the headers.

3.6.3.1 PGM file format

Portable Gray Map (PGM) file format is part of a package called Netpbm, which contains 220 separate programs for dealing with files such as PGM, pbm and pnm. As the name suggests, it is a lightweight-greyscale image format, which is simple for use in programs, making it ideal for this project.

The structure of PGM files is very specific and is defined as [65]:

```
A 'magic number' which identifies the file type. A pgm image's magic number is the two characters 'P5'.
Whitespace (in the format of tab, space etc)
A width, formatted as ASCII characters in decimal.
Whitespace (in the format of tab, space etc)
A height (in the same format as width)
Whitespace (in the format of tab, space etc)
Maximum Grey Value (Maxval) - usually 255
Single whitespace character (typically new line)
A raster of Height rows, in order from top to bottom. Each row consists of Width gray values, in order from left to right. Each gray value is a number from 0 through Maxval, with 0 being black and Maxval being white. Each gray value is represented in pure binary by either 1 or 2 bytes. If the Maxval is less than 256, it is 1 byte. Otherwise, it is 2 bytes. The most significant byte is first.
```

Figure 3.20: PGM header rules

A comment in PGM is proceeded by the # symbol, and is not counted in the formatting as defined in Figure 3.20.

3.6.3.2 Specific file format for Congealing

When investigating the Congealing demo code, it became apparent that comments were utilised in the reading-in of image information.

```
P5
# 28 28 6742
2324 2324
255
```

Figure 3.21: Example MNIST PGM file header

Listing 3.21 above shows the first 5 lines of the PGM MNIST data which was included in the Congealing demo. The second line, proceeded by a #, therefore a comment, includes information on height and width of each individual MNIST number (28 and 28), and how many of these numbers are included in the large file (6742).

This information is then used to set the number of images per row and to set an array to the appropriate height, width and number of included images in the `loadSeries.m` function.

3.6.3.3 Creating an appropriate save function

The next step was to write a function which would appropriately concatenate the MINI-MIAS dataset [76] to create a large PGM input image for Congealing. This led to the function `pgm2bigPgm.m`, which is a pre-processing funciton before the original `saveSeries.m` demo function, which will:

- read in the number of images in the chosen directory
- identifies the dimensions of each scan in the directory (with MINI-MIAS, they are all the same dimensions)
- creates a string containing all the suitable information needed for reading (as outlined in Subsubsection 3.6.3.2)
- creates a file called "big_scan.pgm" and saves all the images out to the one file (after transposition, as in Subsection 3.6.1)

3.6.3.4 Final outcome

An example of the final mammogram image data can be found in Figure 3.16. When the user specifies an odd number of images, or the number of images do not create a square, extra black padding is created around the image, as can be observed in Appendix E Figure **??**.

3.6.4 Vectorisation

Vectorisation is the process to replace loop-based code with MATLAB matrix and vector operations. As stated in the MATLAB documentation [57], Vectorisation is important for several reasons:

1. Appearance - more concise, more like what is seen in textbooks
2. Less Error Prone - less for loops = less lines of code for errors to appear
3. Performance - vectorised code usually runs a lot faster

The initial implementations of both `membership.m` and `nonProbabilistic.m` contained for loops, so experimentation was run before, during and after vectorisation to evaluate the supposed performance increase.

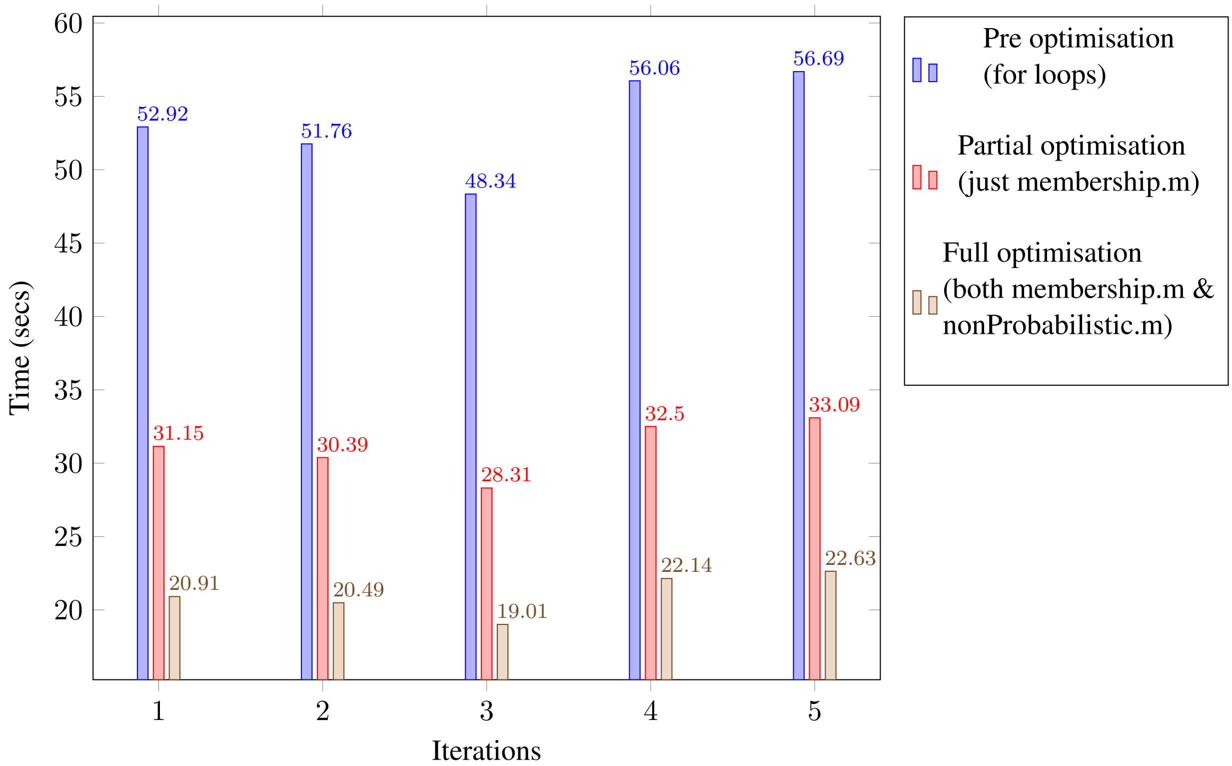


Figure 3.22: Time per iteration before, during and after vectorisation

Figure 3.22 demonstrates the time taken per iteration, in the same environment, to run the `binaryCongeal.m` [1] function. A marked improvement can be seen just by vectorising the `membership.m` function, and further improvements once the `nonProbabilistic.m` function for Non-Probabilistic entropy was vectorised.

[1]The function which calls the specified entropy algorithm.

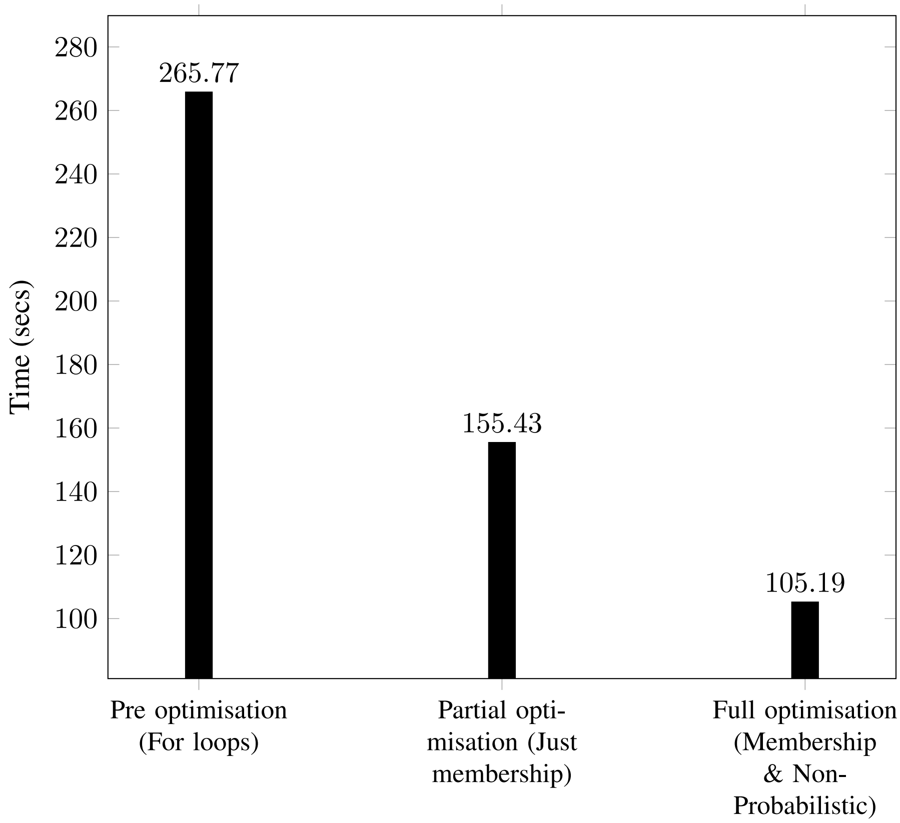


Figure 3.23: A comparison of the total time to run 5 iterations prior to vectorisation, during (part vectorisation) and post-vectorisation.

Figure 3.23 outlines the total time taken to run 5 iterations of Non-Probabilistic Entropy before vectorisation, once vectorisation was complete on the `membership.m` function, and finally after full vectorisation.

Chapter 4

Results and Conclusions

4.1 Results

4.1.1 Alignment results

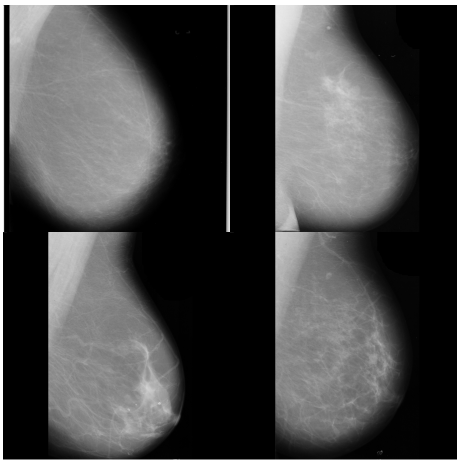


Figure 4.1: 4 input scans.

Figure 4.1 shows the input image utilised in Figure 4.2. This large file contains 4 scans from the BI-RADS I classifcation, containing no masses in the tissue. Whilst relatively similar in size and shape, the tissue make-up inside the breast is varied.

Set No.	BI-RADS Class	No. of images	Pixel dimensions
1	BI-RADS I	4	2048 x 2048
2	BI-RADS II	6	3072 x 3972
3	BI-RADS III	8	3072 x 3072
4	BI-RADS IV	5	3072 x 3072

Table 4.1: Input image information.

Results for each set outlined in Table 4.1 can be found in Appendix E.

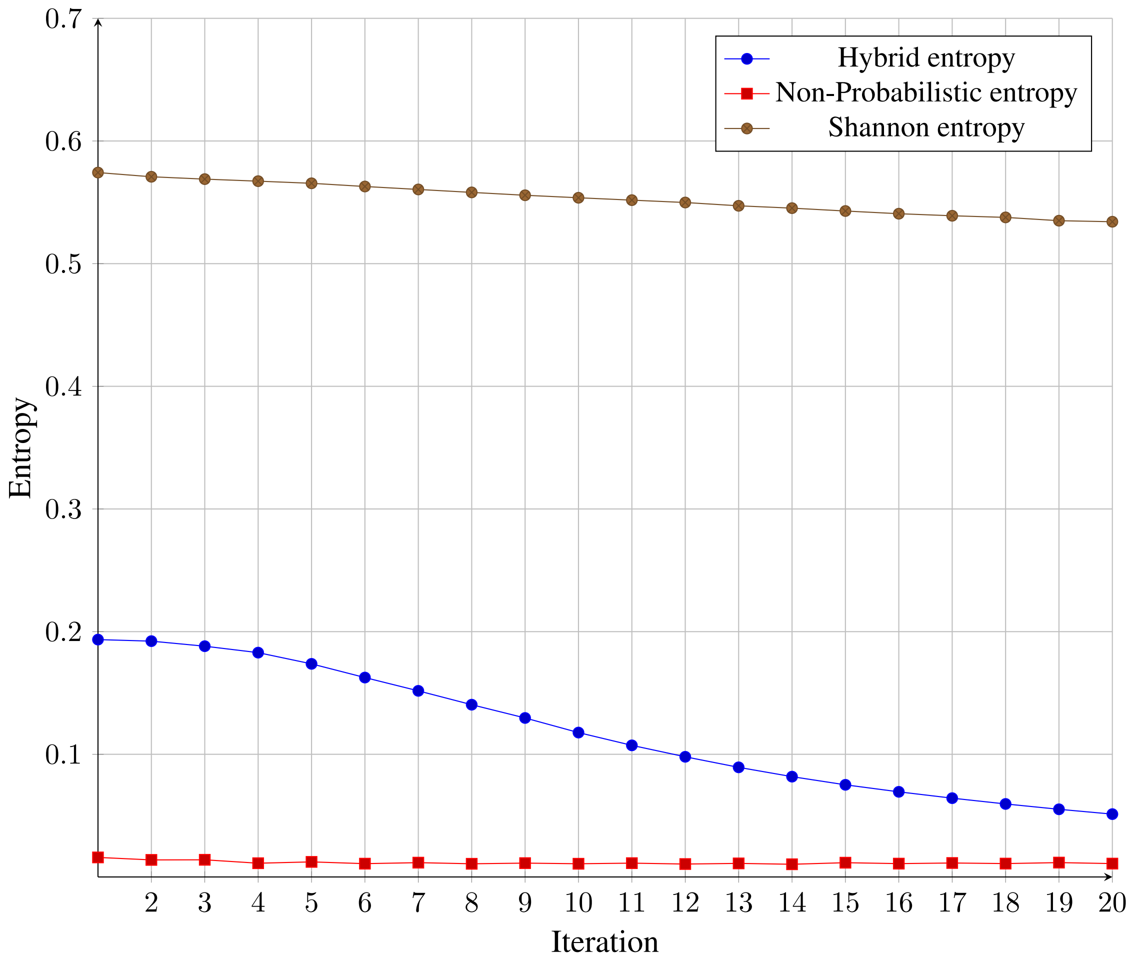
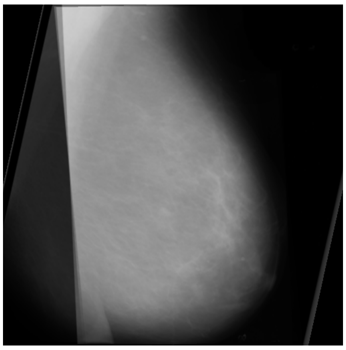


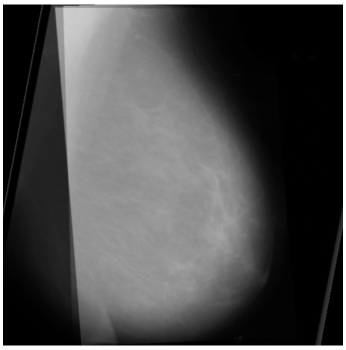
Figure 4.2: Comparison of the reduction in entropy on each iteration on Sample Set 1.

Whilst the entropy decline of Shannon entropy and the fuzzy entropy algorithms can be plotted on one graph to show the general trend of decreasing entropy, it is worth noting that they are indeed not comparitive. As Shannon entropy does not contain any possibilistic uncertainty, the entropy outcome is vastly different in it‘s calculation to that of both Non-Probabilistic and Hybrid.

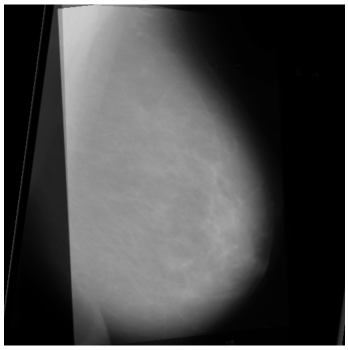
4.1.1.1 Shannon Entropy



(a) 5 Shannon Entropy iterations on BI-RADS I sample set.



(b) 10 Shannon Entropy iterations on BI-RADS I sample set.



(c) 20 Shannon Entropy iterations on BI-RADS I sample set.

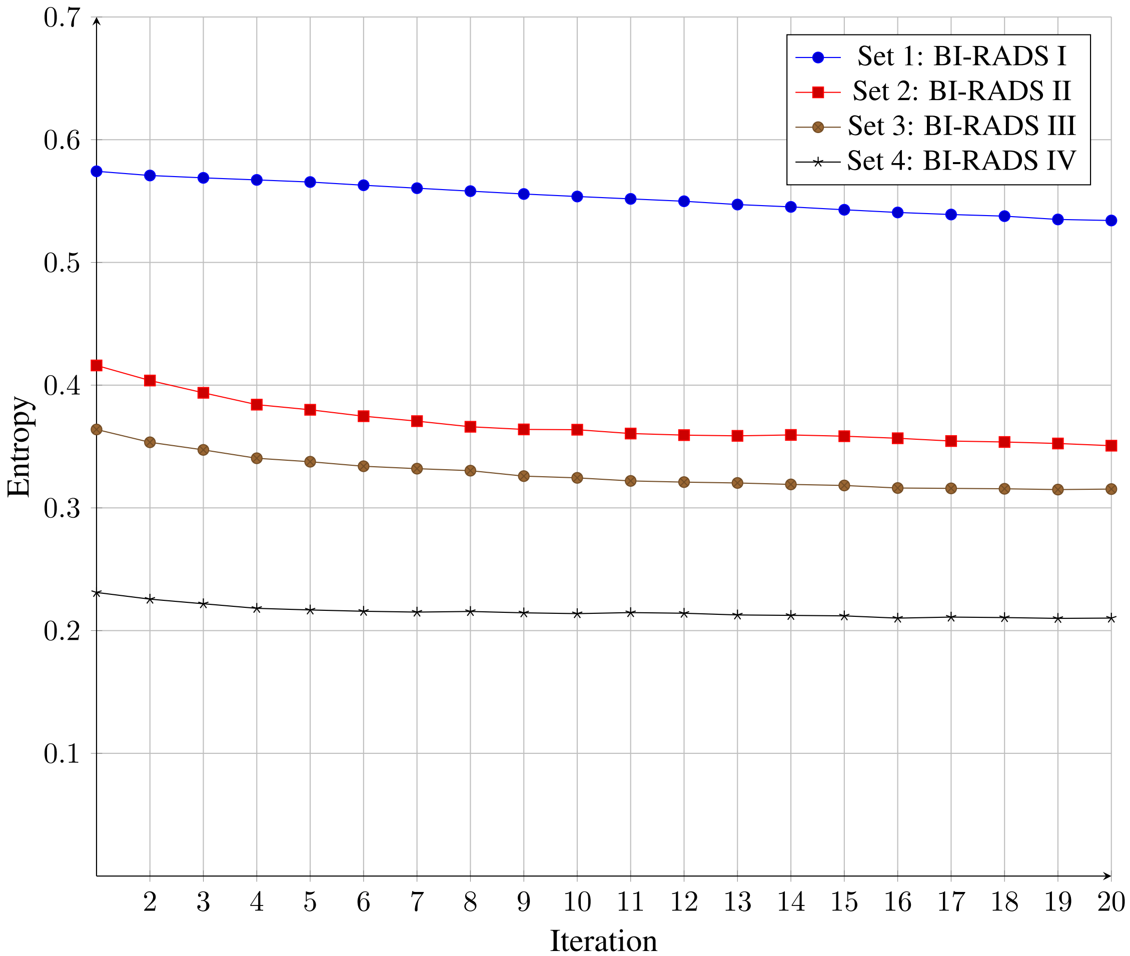


Figure 4.4: Shannon: Comparison of the reduction in entropy from each Sample set, as summarised in Table 4.2.

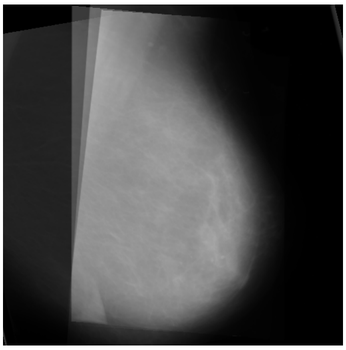
Sample	Starting entropy	Final entropy	Entropy change
BI-RADS I	0.416080	0.350700	0.06538
BI-RADS II	0.574292	0.534195	0.040097
BI-RADS III	0.363914	0.315340	0.048574
BI-RADS IV	0.231113	0.210216	0.020897

Table 4.2: Shannon entropy difference table for each sample set.

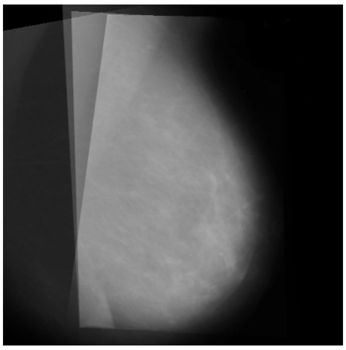
The largest decrease in Shannon entropy over all 4 of the sample sets can be seen in BI-RADS I. The output for each of the sample sets is relatively sensible (as can be seen in Appendix E - Section 5.2) and the run time of each iteration is quick due to the lookup table implementation (covered in later Subsection 4.1.2), demonstrating that Learned-Miller‘s original code is still useful even upon mammograms.

The gradual straightening of the curve of all 4 sets, as represented in Figure 4.4, indicates a slowing in the reduction of entropy, which would most like result in the over-congealment of the input images. Over-congealing is when the entropy is reduced to such a point that any further iterations could either: increase the entropy; or reduce the entropy to an unreasonable level.

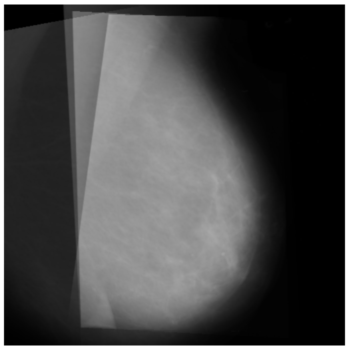
4.1.1.2 Non-Probabilistic Entropy



(a) 5 Non-Probabilistic entropy iterations on BI-RADS I sample set.



(b) 10 Non-Probabilistic entropy iterations on BI-RADS I sample set.



(c) 20 Non-Probabilistic entropy iterations on BI-RADS I sample set.

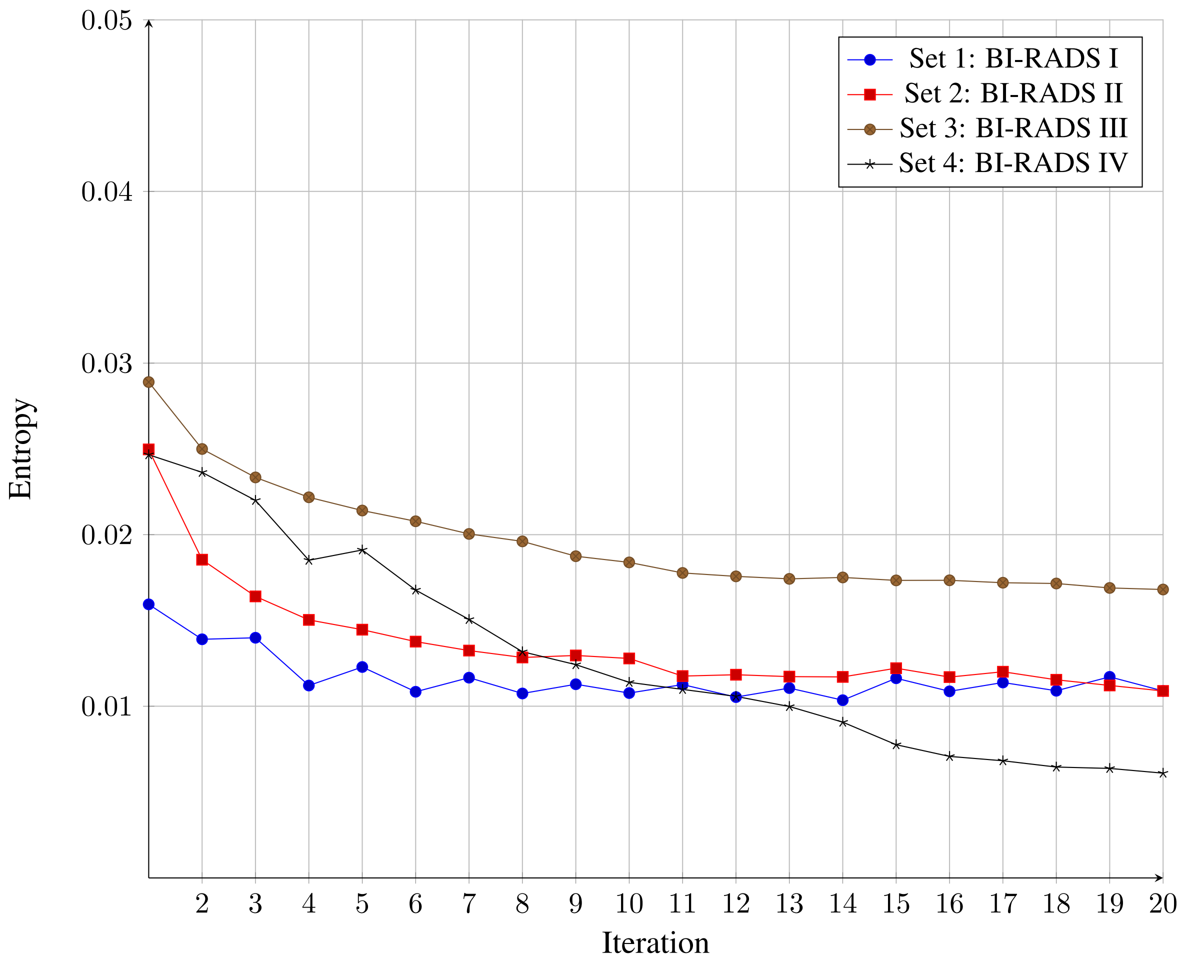


Figure 4.6: Non-Probabilistic: Comparison of the reduction in entropy over iterations, as in Table 4.3.

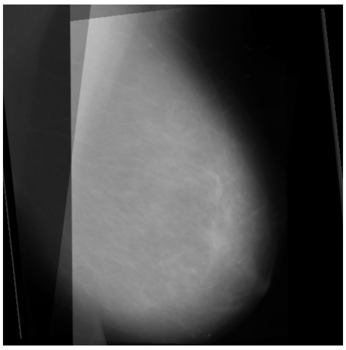
Sample	Starting entropy	Final entropy	Entropy change
BI-RADS I	0.015932	0.010889	0.005043
BI-RADS II	0.024955	0.010888	0.014067
BI-RADS III	0.028877	0.016796	0.012081
BI-RADS IV	0.024644	0.006101	0.018543

Table 4.3: Entropy table for Non-Probabilistic

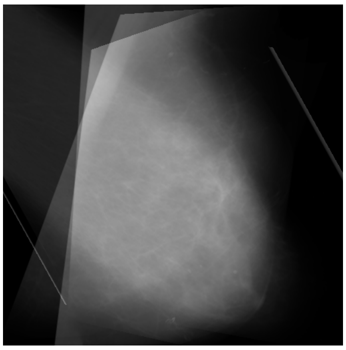
As outlined in both Figure 4.6, and Table 4.3, the entropy for image alignment using the Non-Probabilistic algorithm tends to be quite low. Because of this, often the entropy decline seems to be quite low, however generally the intial entropy can be lower than the final entropy of Hybrid entropy.

Sample sets 1 & 2 can be seen in Figure 4.6 to be fluctuating, especially in the latter iterations. This is due to over-congealing the image, as mentioned in the previous section. In Set 1 for example, the entropy of the images barely decreases between iterations 4 and 20, therefore the algorithm could be stopped a lot quicker, saving time for the user.

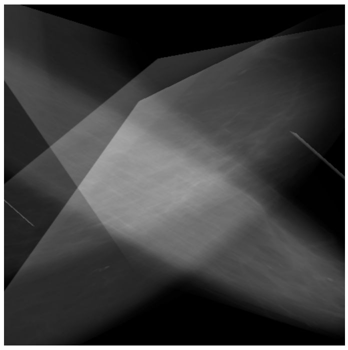
4.1.1.3 Hybrid Entropy



(a) 5 Hybrid entropy iterations on BI-RADS I sample set.



(b) 10 Hybrid entropy iterations on BI-RADS I sample set.



(c) 20 Hybrid entropy iterations on BI-RADS I sample set.

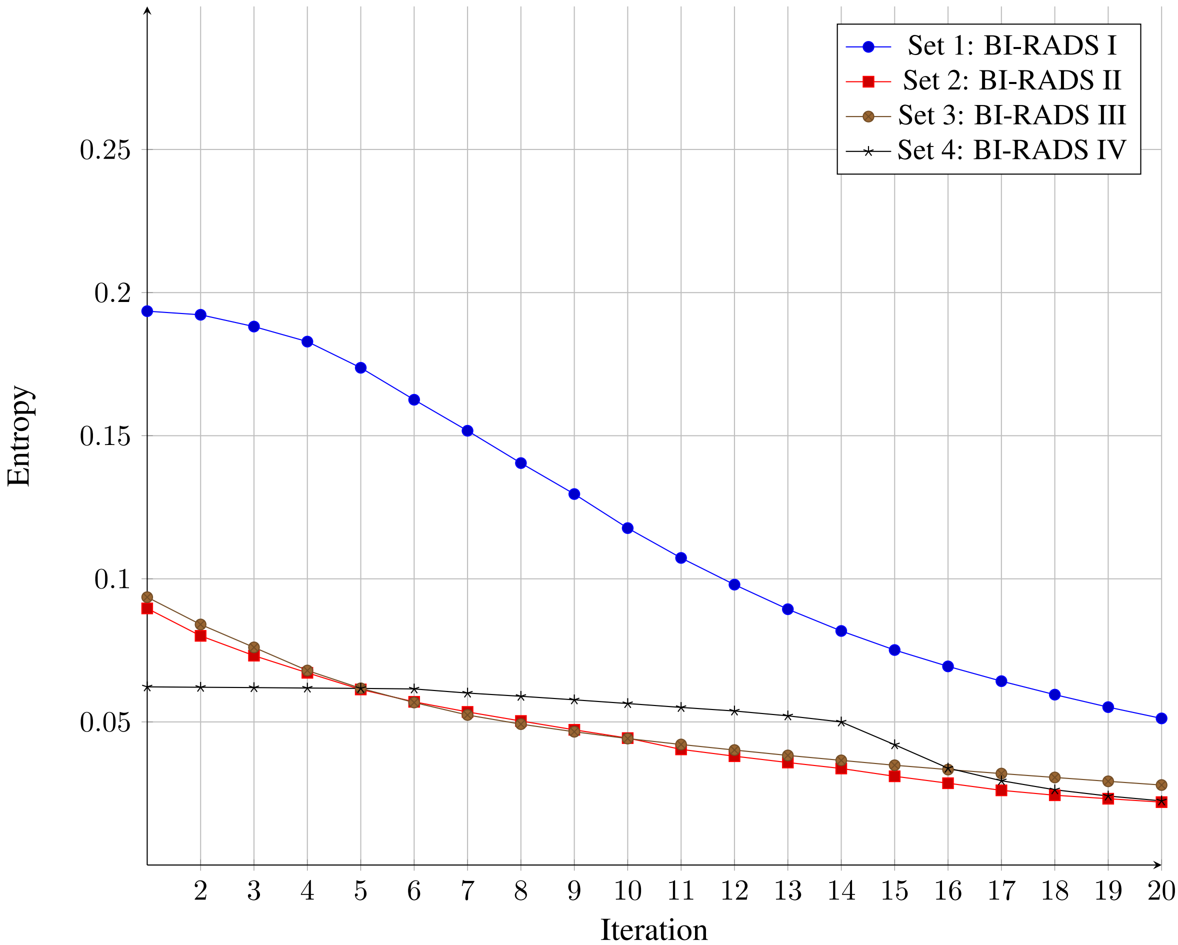


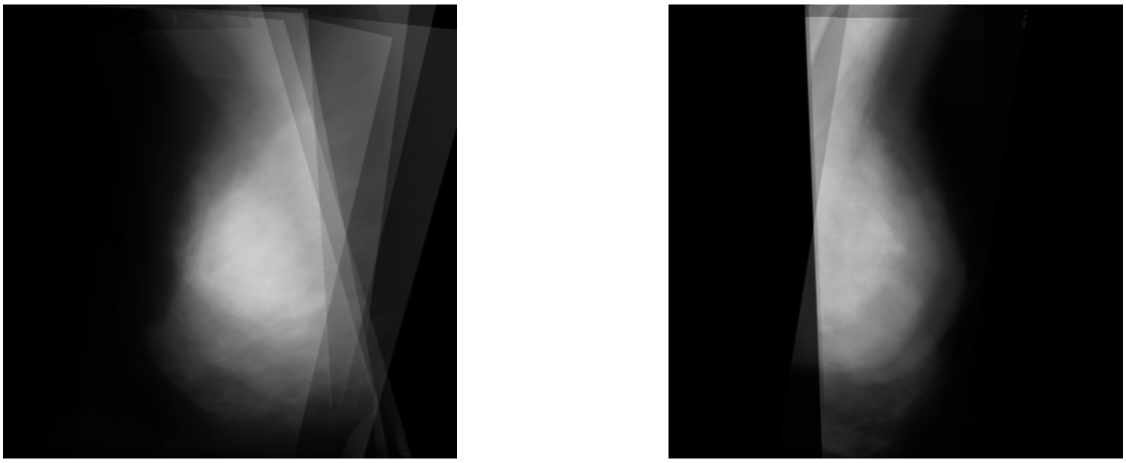
Figure 4.8: Hybrid: Comparison of the reduction in entropy over iterations, as in Table 4.4.

Sample	Starting entropy	Final entropy	Entropy change
BI-RADS I	0.193522	0.051228	0.142294
BI-RADS II	0.089658	0.021976	0.067682
BI-RADS III	0.093583	0.027946	0.065637
BI-RADS IV	0.062255	0.022408	0.039847

Table 4.4: Entropy table for Hybrid

In the previous two sections, over-congealing has been spoken about. Figure 4.7c is a perfect example of an image becoming overly-congealed. However, the graph does not seem to mimic the erratic behavior displayed by Non-Probabilistic entropy. This brings to light how different the fuzzy entropy implementations can be, and how the non-deterministic nature of the Congealing algorithm can lead to dramatically different results, depending upon which order the image transformations are run.

It could be mistakenly understood that Hybrid entropy should be stopped between 5 and 10 iterations, however it is purely dependent upon the dataset fed into the Congealing algorithm utilising Hybrid alignment, as is demonstrated in Figure 4.9. Whilst Figure 4.9a, utilising the third data set (BI-RADS III,) could be seen to past the optimal alignment at only 5 iterations, Figure 4.9b, utilising the fourth data set (BI-RADS IV) is nearing the best alignment possible. This is one reason it is difficult to ascertain when to stop the Congealing algorithm automatically.



(a) Sample set 3 (BI-RADS III) after 5 iterations.

(b) Sample set 4 (BI-RADS IV) after 10 iterations.

Figure 4.9: Example of how different datasets behave when aligning with Hybrid entropy.

4.1.2 Run-time results

Alignment metric	No. of iterations	Time to run (secs)			Average time per 1 iteration
		5 iterations	10 iterations	20 iterations	
Shannon	BI-RADS I	6.4	11.9	23.5	**1.22**
	BI-RADS II	8.7	16.11	30.0	**1.62**
	BI-RADS III	11.8	21.2	40.7	**2.17**
	BI-RADS IV	8.9	15.5	29.9	**1.61**
Non - Probabilistic	BI-RADS I	135.2	277.2	573.6	**27.81**
	BI-RADS II	214.6	425.8	893.8	**43.40**
	BI-RADS III	275.5	595.3	1243.0	**58.93**
	BI-RADS IV	143.3	324.4	706.4	**32.14**
Hybrid	BI-RADS I	19.3	38.3	76.1	**3.83**
	BI-RADS II	30.2	56.9	108.7	**5.72**
	BI-RADS III	40.6	76.5	150.6	**7.77**
	BI-RADS IV	25.6	50.1	98.6	**5.02**

Table 4.5: Run-time statistics for each set over 5, 10 & 20 iterations.

Table 4.5 outlines the run-time statistics for each input set, along with the average time taken to run 1 iteration for each set.

When comparing run-times to the number of images congealed, there is a clear correlation between more images congealed and greater run time (e.g. Set 3: BI-RADS III), with some algorithms on average taking twice as long to run 1 iteration. This is to be expected due to the larger amount of information to be processed upon each iteration as each image is aligned with the other.

Shannon entropy has the quickest run-time of the three algorithms. This is likely to be due to the implementation of a lookup table - something which could have been considered for the two fuzzy entropy algorithms.

Non-Probabilistic can be seen to be much slower than it‘s counterparts. This could be due to implementation, however when analysed alongside the entropy result, it could also be down to a higher accuracy rate. This is a trade-off that the user would have to consider when aligning images.

Hybrid entropy has admirable run-time statistics given the complexity of the mathematics needed to compute the entropy value. However the algorithm can be seen to struggle on the larger dataset (Set 3: BI-RADS III), with a runtime of 3.6 times that of Shannon entropy per iteration.

4.2 Conclusions

The project provides results which are a promising sign towards implementing image alignment techniques using light-weight fuzzy entropy algorithms. The results show a varied output of aligned mammographic images, each usually maintaining a natural shape.

4.2.1 Does the use of fuzzy entropy alignment metrics improve the alignment of mammograms?

The judgement as to whether the fuzzy entropy metrics align images ‘better’ than standard Shannon entropy is a subjective one. However this project does show that the output gained from fuzzy entropy Congealing is somewhat different to that of Shannon entropy, and therefore could be perceived to be more useful to a mammographer.

4.2.2 Do mammographers find the output at all useful?

Unfortunately due to the time constraints of this project, the author was unable to ascertain a firm conclusion to this hypothesis. It is hoped that given the sensible nature of the outputs produced, that this work would indeed be useful to mammographers in their classification of breast tissue density, as it provides a generalised average image of each BI-RADS density classification.

4.2.3 What advantages / disadvantages does each entropy alignment metric entail?

4.2.3.1 Shannon entropy

Advantages: The Shannon entropy implementation provided by Learned-Miller has the quickest performance rate due to the lookup table implementation, and straightforward mathematics involved. Combining this with the sensible output the user can expect after 20+ iterations, makes this an extremely strong bench-mark for the two fuzzy entropy implementations.

Disadvantages: However, this project was about utilising uncertainty and using it to model breast tissue accurately. Shannon entropy provides no such uncertainty, and therefore the alignment of the images is based purely on if the pixels match one another exactly, with no variation allow. This rigidity can translate into a slower alignment process.

4.2.3.2 Non-Probabilistic entropy

Advantages At 20 iterations, Non-Probabilistic entropy can be seen in some experiments to be “over-congealing” the input images, yet the output images were never corrupted nor distorted. This natural slowing on entropy decline, with fluctuating results in the latter stages is extremely reflective of nature. Were a human try to align two images by hand, their initial results would see large changes in alignment, however the more precise they try to be, the more the alignment is likely to just “miss” the mark.

This alignment metric exhibits a quick decrease in entropy in the first few iterations of the Congealing algorithm. This results in fewer iterations necessary for an accurate alignment when compared to it‘s counterparts.

Disadvantages Whilst the metric takes very few iterations to align the input images accurately, the performance of this implementation is not ideal. With run-times up to 4830% longer (*1.22 vs 58.93 seconds per iteration*) than that of Shannon entropy, it is by no means a quick solution. This could be due to implementation, however the final run-times produced for Non-Probabilistic entropy are inclusive of the vectorisation optimisations carried out during implementation. Prior to vectorisation run-times were significantly worse, as detailed in Subsection 3.6.4.

As the algorithm name suggests, this fuzzy entropy metric does not model probabilistic uncertainty, just that of possibilistic, therefore does not model true uncertainty to the same level as Hybrid entropy.

4.2.3.3 Hybrid entropy

Advantages The run-times produced by Hybrid entropy are close to that of Shannon entropy‘s. This is admirable given the complexity of the calculations involved in the generation of the input image entropy value. This quick run-time brings to the user a truly usable image alignment method which leverages fuzzy entropy.

Whilst the entropy was often not as small as that output by Non-Probabilistic entropy, it produced an admirable attempt at initially aligning the input images, and went on to produce the highest degradations in entropy seen in this project.

Disadvantages

One issue faced by Hybrid entropy was over-congealing. Whilst the entropy value did not indicate towards over-congealment, the image output afterwards did cause some concern. As mentioned previously (Subsection 4.1.1.3), estimating when to cut off Hybrid entropy is difficult, as over-congealment often happened at varying iterations given different data-sets.

4.2.4 Future Advancements

Several features were identified for improvement, plus additional functionality could be added to the project if the project time-frame had been longer, or if someone continues this project at a later stage.

4.2.4.1 Automated medical marker removal

Feature extraction is an ongoing research area. This could be leveraged in the future for the automatic detection and removal of medical markers and/or artifacts. Some interesting areas of research which may be viable for implementation include:

Corner detection: As the artifacts are typically rectangular in shape, a corner detector could be used to identify their presence - such as Harris Corners, or KLT. Harris Corners was developed by Chris Harris and Mike Stephens in 1988, in a technique which combines an edge detector with a corner detector [35]. Harris Corners algorithm is looking for a sharp change in intensity within the image, such as passing from the black mammogram background onto the white area of the artifact. If a right-angle in intensity difference can be detected using several sliding windows over the image, then a corner is detected.

The work of Kanade-Lucas-Tomasi on the self-titled algorithm KLT builds upon that of Harris and Stephens [71] [78]. This implements a scoring function where the highest value (e.g. x) above a threshold are classed as a corner, and all other values which neighbour x are removed from contention.

When compared, both Harris and KLT corners perform very similarly and are both still regularly used today.

SURF: Speeded Up Robust Features (SURF) [16] is a speeded up version of an older feature extraction method called Scale Invariant Feature Transform (SIFT) [45]. SIFT boasts a number of useful features such as being scale invariant (useful as not all artifacts are the same size), invariant to rotation (again useful as artifacts are often at different orientations) and features are selected only if there is a good enough contrast (as artifacts are typically white on a black background, this is helpful). SURF improves upon SIFT by utilising different-sized box features rather than Difference of Gaussian (DOG) filters for refining the space in which scale is analysed, therefore making it quicker.

4.2.4.2 Classify unseen mammographic images

If further work was undertaken to build upon this project, the application could be used to classify new, unseen mammographic images into the correct BI-RADS classification using some kind of Machine Learning method. The application could be trained upon the existing Mini-MIAS dataset [76], along with other datasets which contain healthy tissue, such as DDSM [36] [37] to create an accurate model of what each classification should tend towards.

Research is ongoing into which Machine Learning technique is the best for classification of images, however the likely choice would be either an Artificial Neural Network (ANN) or a Support Vector Machine (SVN) given their high accuracy rate, as outlined in [39]. Currently both of these are slow to train, however that wouldn't necessarily be a problem for a Doctor to deal with, as it could be trained prior to their usage, or trained annually on new available data to become even more accurate. A high accuracy rate is more important than the training rate due to the human-diagnosis element of the application.

4.2.4.3 Optimisation of Non-Probabilistic

The functionality to run all 3 alignment metrics together was not implemented due to the performance issue of Non-Probabilistic entropy, however if further work was done to optimise the performance, then the author can see no issue as to why all three alignment metrics could be run, and the results displayed side-by-side for comparison.

The current implementation means the user would have to run the three alignment techniques sequentially, the outputs saved to their PC, and then later opened in an image preview software to compare the results alongside each other. This is not a major issue, however it would be advantageous if the user could view them side-by-side easily.

4.2.4.4 Further optimisation

Further optimisation could be carried out for the two fuzzy entropy algorithms by implementing a lookup table structure, as in Learned-Miller's example of Shannon entropy. Whilst this is unlikely to make massive gains for Hybrid entropy, for Non-Probabilistic entropy this could see a significant reduction in run-time, making the algorithm more usable in real-world environments.

4.2.4.5 GUI alterations

Small edits would be made to the `removeMarker` GUI, such as a clear button, and the functionality to draw more than one area to remove from an image.

The load functionality in the main GUI could be compiled into one button - this was initially attempted during the project, however a lack of knowledge into how to detect if a file already exists, along with time constraints, inhibited this advancement.

Chapter 5

Critical Evaluation

I believe it is important for myself to reflect upon the work carried out during this project and to pinpoint areas of success and weakness. The main feeling is one of accomplishment, from learning a new programming language, to delivering a proof-of-concept application which could one day be built upon and help Doctors in their fight against breast cancer (or any other cancer if they so wished).

This section will analyse areas of the project where there have been both achievements and deficiencies, and delve deeper into the topics where there is greater discussion to be had.

These sections will be:

- Initial requirements
- Process
- Choice of implementation tool
- Choice of technologies
- Use of blog
- If the project were to be restarted
- Degree relevance
- Final words

5.1 Did the initial requirements still stand at the end of the project?

The answer to this question is both yes, and no. The main gist of the project was very much the same, and the outcome was no different than initially planned, but the work involved did evolve over time - as is the nature of an Agile project.

Initially Dr. Neil Mac Parthaláin, my supervisor, and I,identified a need for an application which would take in a set of mammographic images, and would return the output from 3 different fuzzy entropy algorithms. Unfortunately, due to my lack of knowledge in MATLAB, initial progress was slower than I anticipated, causing some worry about the amount of work which could be completed in the project time-frame. Along with this, the deficit in fuzzy entropy knowledge led me to believe that Fuzzy Shannon entropy would be a good algorithm to implement, however when meeting with Dr. Mac Parthaláin it was found to not be suitable for the project. Because of this, the initial requirement to have 3 fuzzy entropy implementations was reduced to 2, as there was a lack of time left in the project.

A requirement identified about mid-way through the process was the need to remove medical markers / other artefacts from the scans. This addition falls into an active area of research, so in the time available to myself, it was only possible to implement a very simple and rudimentary way to remove such unwanted markings.

5.2 How did the choice of Process support the project?

It was identified very early on in the project that the project would undertake an Agile process. This was due to a lot of uncertainty surrounding both the requirements, and the actual work which would need to be undertaken. Overall, I'm extremely happy with their choice of process, as it provided just enough structure to ensure the project stayed on track and was well thought-out. The project may have benefited from a true TDD approach, however upon reflection, I was too new to the concept of TDD and the language worked in, to apply it to the fullest effect.

Taiga, the tool utilised to support my use of adapted-Scrum, was invaluable in detailing which task were remaining in my product backlog, and which tasks did I still have to undertake that week. The burndown chart for each week was useful for tracking my progress over time, and for pinpointing any days where development might have been slowed by an issue or outside concern. This in itself helped me to write my weekly blog posts - see Section 5.5.

5.3 Was the choice of implementation tool correct for this project?

MATLAB was chosen as the main implementation tool for this project due to the fact the original Congealing algorithm was implemented in it. Unfortunately, more research could have been carried out prior to starting any development, and if that had happened then it is likely that an alternative, such as the open-source alternative Octave [75], would've been chosen.

MATLAB was for one, costly. £29.00 for the initial base MATLAB software, plus £16.00 each for the Image Processing Toolbox and the Fuzzy Logic toolbox, totaling £51.00 spent of implementation software alone. Whilst it has not been verified, it is highly likely that Octave would've been able to run with the Congealing demo code after a few alterations.

Secondly, MATLAB was difficult to set up, and often gave non-descript error messages which were difficult to debug quickly. Whilst the documentation and MATLAB forums were extremely useful, during times of unique implementation - such as that of implementing three fuzzy set trapezia - I often found myself unstuck with no useful guidance online.

5.4 How did the author find working with their choice of technologies?

Before the project I had no worked with MATLAB, nor worked with any kind of Fuzzy logic system - so this project was a steep learning curve. That being said, I now feel extremely accomplished given I have learned a new language in such a short space of time, and implemented a usable application.

I found the MATLAB documentation to be extremely useful and extensive, and the forum in which MATLAB users can post queries and answers held many answers to small quirky issues discovered along the way.

5.5 How did the use of a blog aid the author in this project?

A blog outlining the weekly reviews & retrospectives can be found at `lauramcollins.co.uk/blog`. It provided wonderful insight each week into the work completed, and was invaluable when writing this report to reflect upon the work week by week. It also allowed an informal environment for me to work through issues via a couple of "Hurdles" blog posts - where I would talk about the bigger hurdles through the project, and I tried to keep them updated with the solutions I had discovered.

The blog was also a good source of screenshots, as often many would get lost throughout the 3 month-timeline.

5.6 If the project could be restarted, what would the author keep the same or do differently?

Primarily I would keep the project objectives and hypotheses to be tested the same. This kept me motivated through the 3-month long project and I enjoy the feeling of knowing my work could one day help people in need. I would also keep my Agile approach to process as it supported the project very well and did not make me feel over-burdened with formalities such as documentation.

Something I would change, or at least investigate changing, would be the choice of implementation tool. As mentioned earlier, MATLAB became expensive and was often cumbersome and slow, so there does lie the possibility that Octave might've been a more reasonable choice.

5.7 How relevant was this project to the author's degree scheme?

My degree scheme is *Artifical Intelligence and Robotics*, so whilst there is no element of Robotics, my project can be seen to be venturing into the world of Artificial Intelligence somewhat as entropy, fuzzy sets and machine learning techniques fall within this category.

Whilst this project wasn't the closest alignment to my degree scheme offered by the University, my choice of project was fueled by a personal interest into the topic, and I believe that helped me carry on development when things became hard. If I had not had a vested interested in the general topic I likely wouldn't have enjoyed this project as much.

5.8 Final words

Finally I would like to close with my overall thoughts on my Major Project. It has offered me new insight into how medical practitioners classify breast tissue, it's brought me challenges in the form of a new programming language and a new development style and it has granted me the opportunity to close out my Bachelor's degree with a project on a topic close to my heart. I am extremely pleased in the outcome of the project, whilst the final application mightn't have been as complex in it's calculations and output as I had first hoped, it does signify a step in the right direction for the classification of patient's breast tissue using image processing techniques.

Appendices

Appendix A

Third-Party Code and Libraries

1.1 Congealing Code

The project focused on extending the existing Congealing Code implemented by Learned Miller et al in 2005. A Congealing demo is available on the Congealing website [41] which is open for experimentation. The original demo code was modified and extended to be able to read in mammograms and to work with 2 Fuzzy Entropy algorithms.

1.2 MATLAB

MATLAB was the main implementation tool for this project [58]. It allowed the developer to leverage several built-in functions, such as `max` and `bsxfun` - the former to find the maximum value between two number, and the latter to apply a function (such as max) to all elements in an array. This streamlined the development process by ensuring that menial tasks could often be reduced down to a one-line implementation, especially when leveraging the extra Toolboxes provided by MATLAB. This is also covered in the main body of the report in Subsection 3.2.1 about implementation tools.

1.2.1 Image Processing Toolbox

The Image Processing Toolbox [51] allowed the developer to leverage ready-made algorithms and functions vital for pre-processing and processing of images. Functions utilised in this project include ensuring the image was converted to grey-scale, reading and writing image data from/to files and obtaining image information, such as size and image type.

1.2.2 Fuzzy Logic Toolbox

The Fuzzy Logic Toolbox [50] was primarily used for modeling the fuzzy set membership functions utilised by Non-Probabilistic and Hybrid entropy - using functions such as `trapmf` and `evalmf`. More information is covered in the main body of the report in Subsection 3.4.1.

Appendix B

Ethics Submission

AU Status
Undergraduate or PG Taught

Your aber.ac.uk email address
lac32@aber.ac.uk

Full Name
Laura Collins

Please enter the name of the person responsible for re
Reyer Zwiggelaar

Please enter the aber.ac.uk email address of the perso
rrz@aber.ac.uk

Supervisor or Institute Director of Research Departm
cs

Module code (Only enter if you have been asked to do
CS39440

Proposed Study Title
Entropy based metrics for joint image alignment

Proposed Start Date
25th January 2016

Proposed Completion Date
4th May 2015

Are you conducting a quantitative or qualitative resea
Mixed Methods

Does your research require external ethical approval u
No

Does your research involve animals?
No

Please provide a brief summary of your project (150 word max)
I will be investigating the use of Congealing multiple MIAS dataset mammogram
alignment metrics. If time permits I plan on speaking to a specialist (radiologist) t
output mean images of the congealing process are of any significant use to the res
detection.

**I can confirm that the study does not involve vulnerable participants includi
age of 18, those with learning/communication or associated difficulties or tho
unable to provide informed consent?**
Yes

**I can confirm that the participants will not be asked to take part in the study
knowledge at the time and participants will be fully informed of the purpose
what data will be gathered and how it shall be used during and after the stud
given time to consider whether they wish to take part in the study and be giv
any given time.**
Yes

**I can confirm that there is no risk that the nature of the research topic might
the participant concerning their own involvement in illegal activities or other
risk to themselves or others (e.g. sexual activity, drug use or professional mis
disclosure be made, you should be aware of your responsibilities and bounda
aware of whom to contact should the need arise (i.e. your supervisor).**
Yes

**I can confirm that the study will not induce stress, anxiety, lead to humiliatio
other negative consequences beyond the risks encountered in the participant**
Yes

Please include any further relevant information for this section here:

**Where appropriate, do you have consent for the publication, reproduction or
material?**
Yes

Will appropriate measures be put in place for the secure and confidential sto
Yes

Does the research pose more than minimal and predictable risk to the resear
No

**Will you be travelling, as a foreign national, in to any areas that the UK Fore
Office advise against travel to?**
No

Please include any further relevant information for this section here:

**If you are to be working alone with vulnerable people or children, you may n
Tick to confirm that you will ensure you comply with this requirement shoul
require one.**

Please include any further relevant information for this section here:

Appendix C

Function Outlines

3.1 New functions

membership.m Function to handle the creation and calculation of 3 trapezia fuzzy sets as utilised by Non-Probabilistic entropy.

hybridMembership.m Function to handle the creation and calculation of 2 trapezia fuzzy sets as utilised by Hybrid entropy.

TESTmembership.m Test function to hold and run unit tests associated to both `membership.m` and `hybridMembership.m`.

hybrid.m Function which calculates the Hybrid entropy for the input images.

TESThybrid.m Test function to hold and run unit tests associated to `hybrid.m`.

nonProbabilistic.m Function which calculates the Non-Probabilistic entropy for the input images.

TESTnonProbabilistic.m Test function to hold and run unit tests associated to `nonProbabilistic.m`.

pgm2bigPgm.m Script which takes a directory of images, and saves 1 image containing them all.

TESTpgmCreation.m Test function to hold and run unit tests associated to `pgm2bigPgm.m`.

removeMarker.m Function which handles the removing medical markers GUI's functionality.

Enantiomorph.m Function which handles the main GUI's functionality.

3.2 Modified functions

incrTrans.m This function finds an incremental change in the transformation. It also helps the Congealing algorithm in running the correct alignment metric selected by the user.

binaryCongeal.m This function takes in the series of images wanting to be congealed and returns the adjusted input images, the mean images, the matrix of transformations, and the array of entropy values after congealing.

testCongeal.m This function runs the main congealing algorithm. It also loads in the series of images, resizes them appropriately and runs the `binaryCongeal.m` function, passing in the correct parameters.

3.3 Existing functions

fastEntLookup.m Function which creates the Shannon entropy lookup table and returns the entropy after comparing it against the input images.

saveSeries.m Function which saves the pgm files out in the correct order and with the header needed for `loadSeries.m`.

loadSeries.m Function which loads in the large concatenated pgm file needed for Congealing.

showSer.m Function which transforms the 3D array of image values into a figure with each iteration‘s mean images / the adjusted input images.

getXfrm.m

getXfrms.m

computeXfrmImg.m

computeXfrmImgs.m

Appendix D

CRC Cards

The following CRC Cards are only an outline of the most important functions and scripts in the project. Other, less-important, existing functions/scripts may be referred to in the collaboration section.

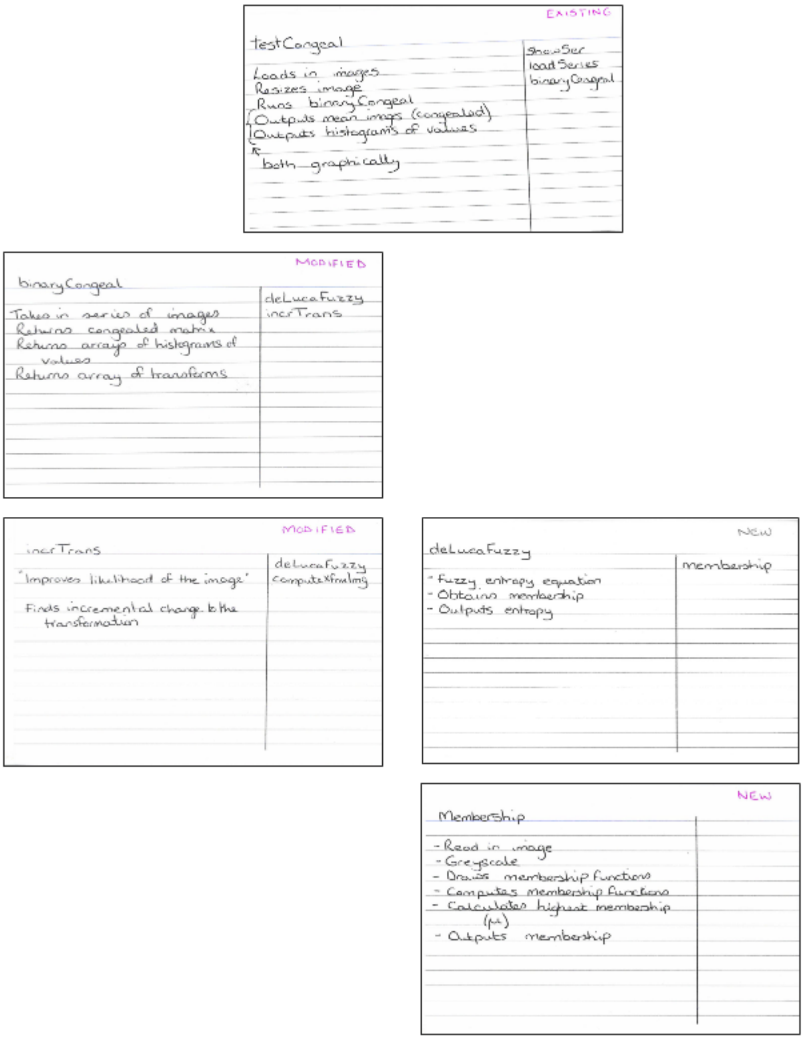


Figure D.1: Initial design

Figure D.1 details the initial planned design for the implementation of De Luca & Termini's Non-Probabilistic entropy. The top-down structure represents the order in which the scripts or functions were called, with testCongeal being the function run by the user.

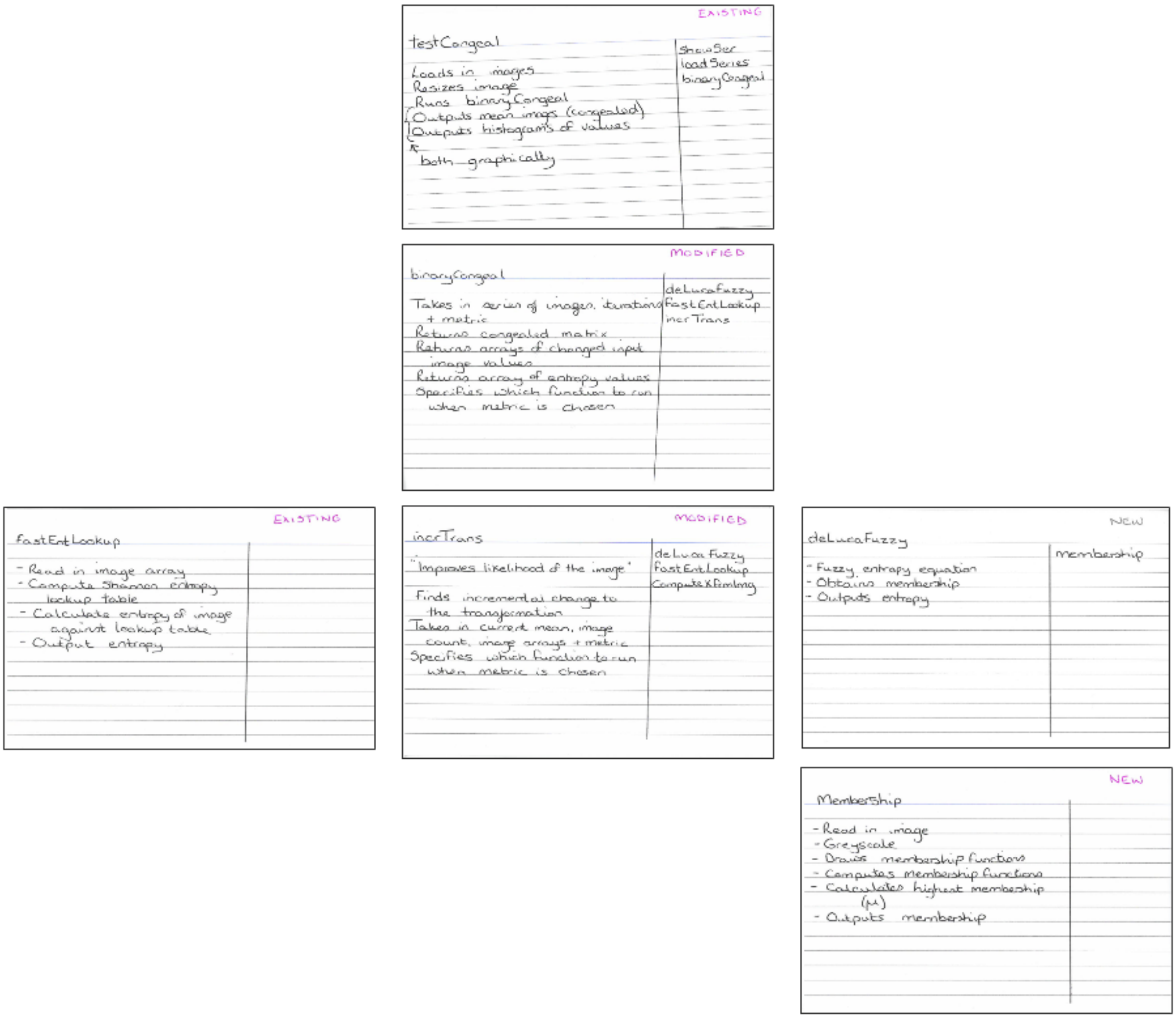


Figure D.2: Secondary design

Figure D.2 shows the second design iteration. Note the change in `incrTrans` and `binaryCongeal` CRC cards, and the addition of `fastEntLookup` when Learned-Miller's Shannon Entropy metric is reintroduced.

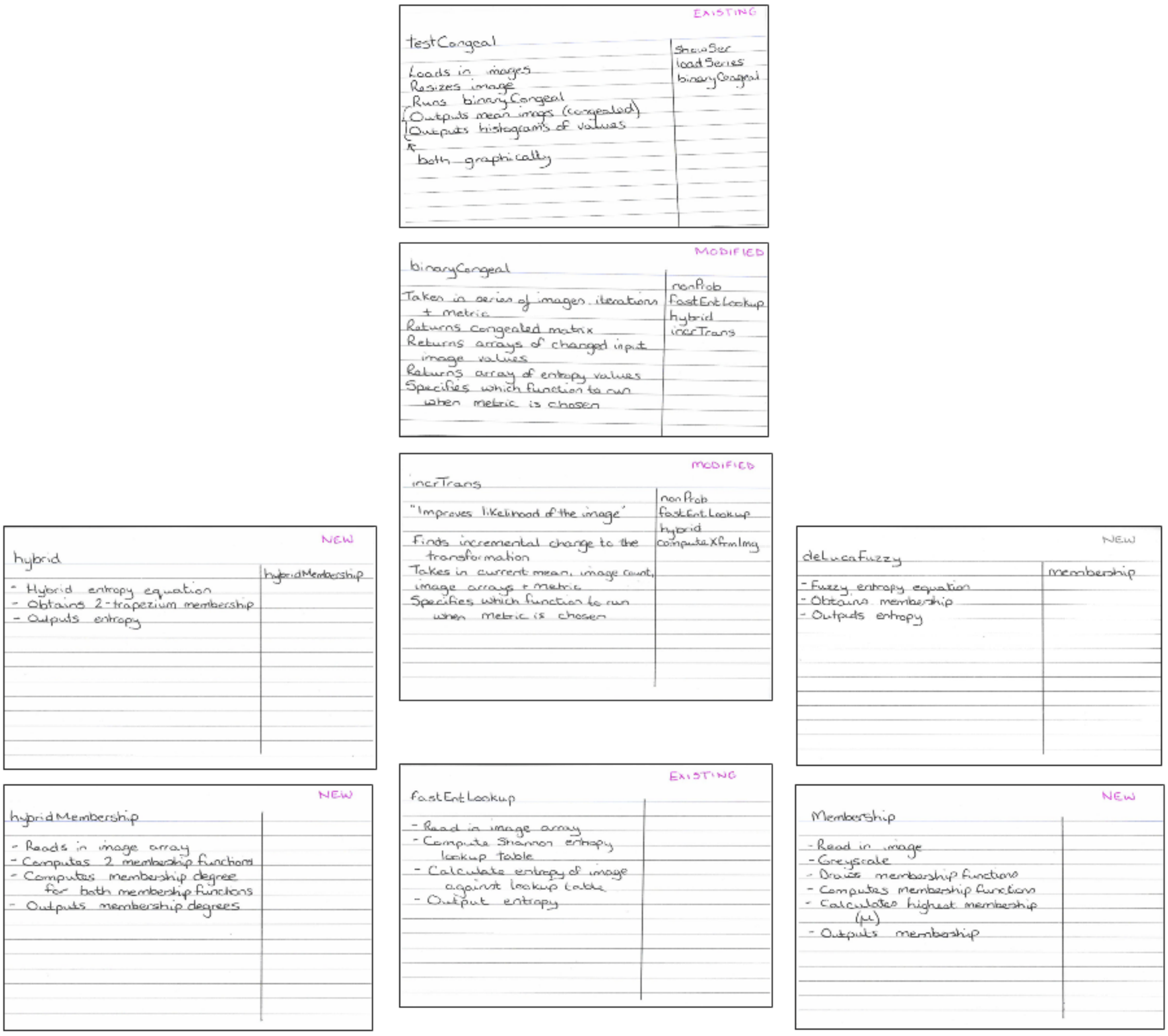


Figure D.3: Third design

Figure D.3 represents the third iteration. The new `hybrid` and `hybridMembership` functions have been added, and `binaryCongeal` and `incrTrans` have once again been updated to reflect this.

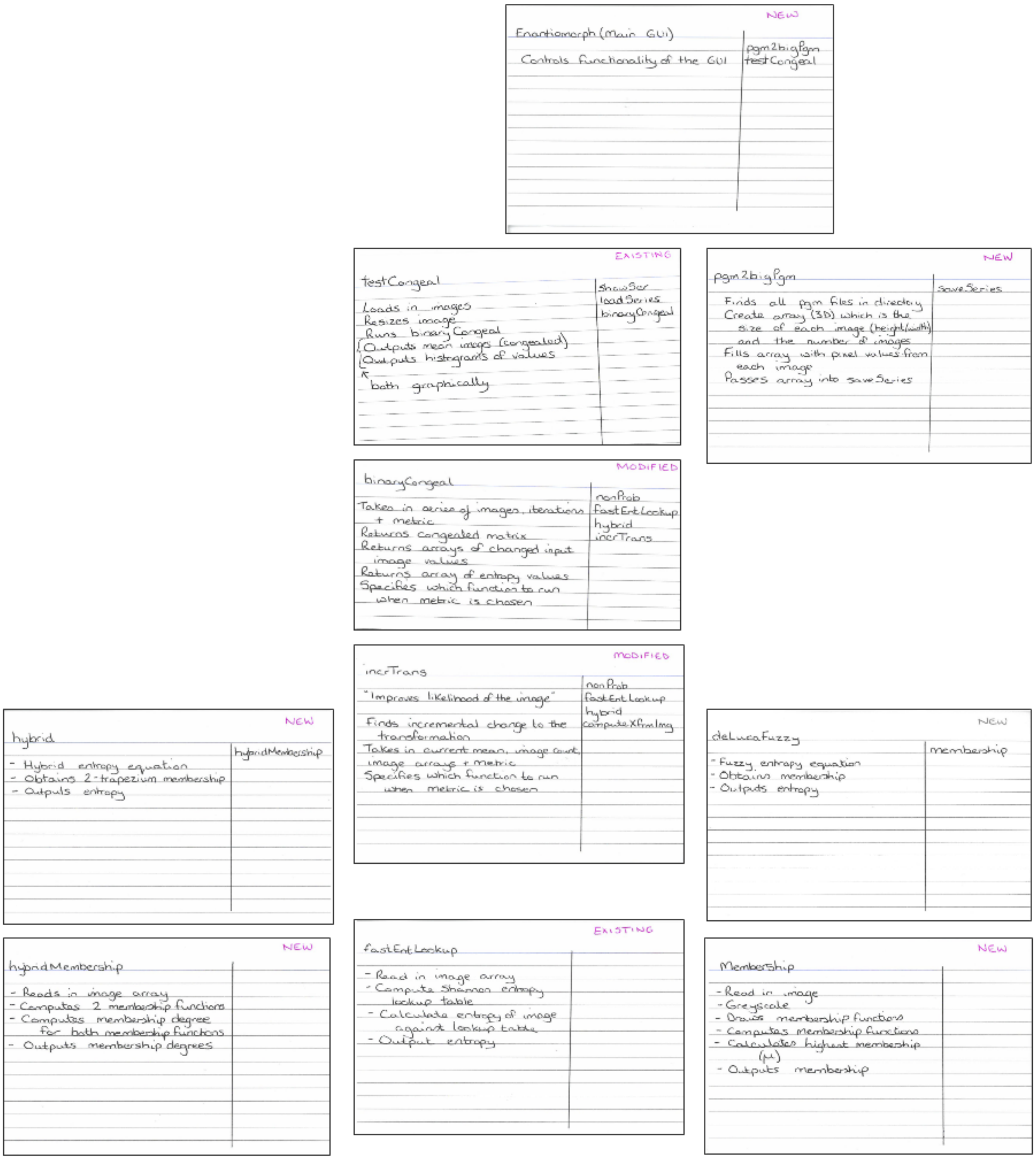


Figure D.4: Fourth design

Figure D.4 shows the addition of the GUI element, called `Enantiomorph` and also the function designed to handle the concatenation of mammogram pgm files `pgm2bigPgm`.

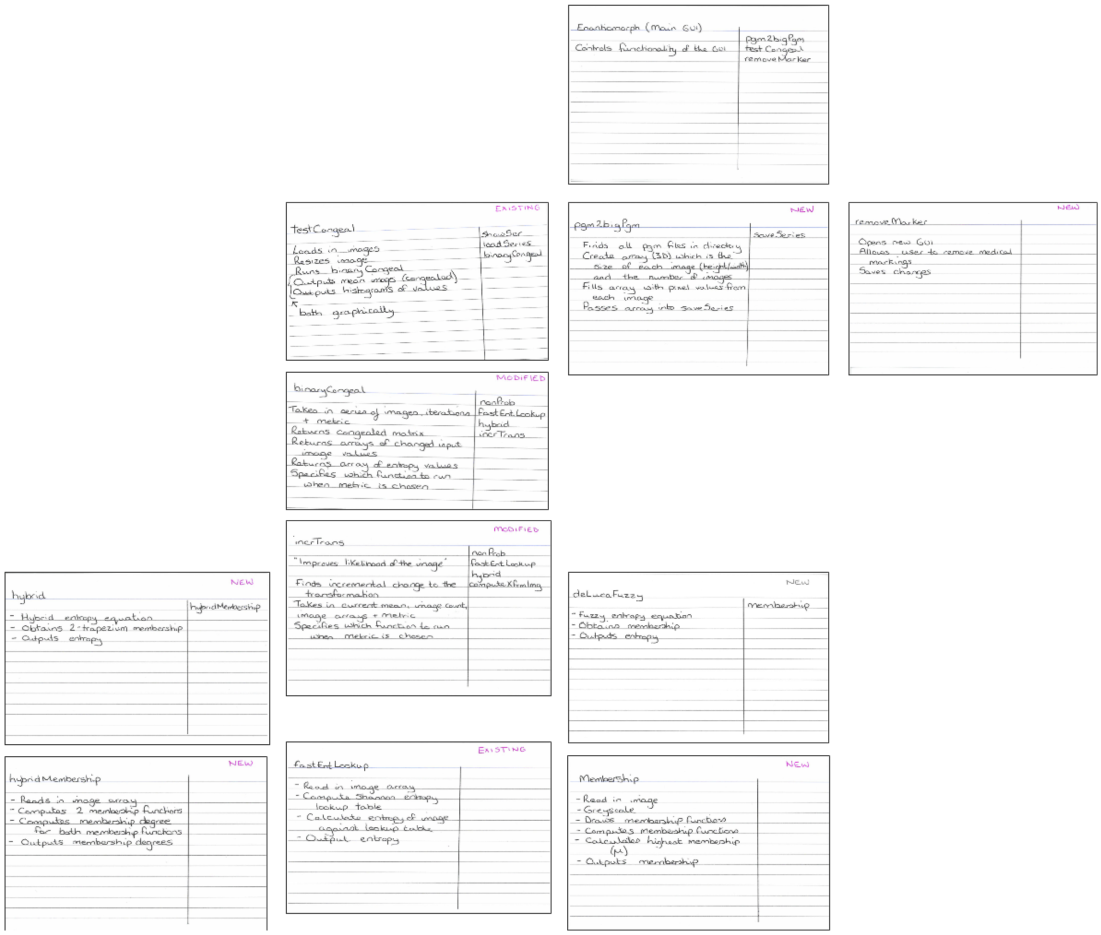


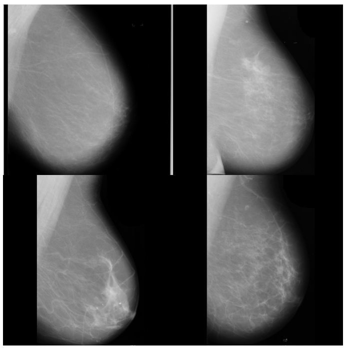
Figure D.5: Final design

Figure D.5 is the final high-level design of the system. It includes the addition of the secondary GUI, `removeMarker` which handles the removal of any medical markings from the mammograms.

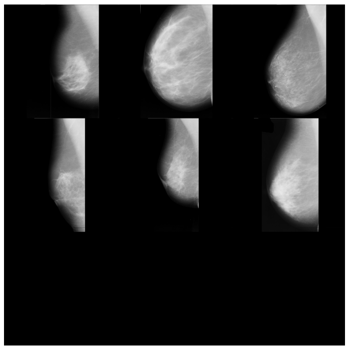
Appendix E

Further Results

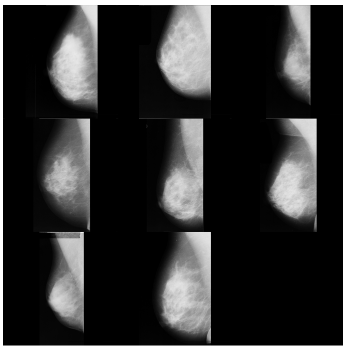
5.1 Input images



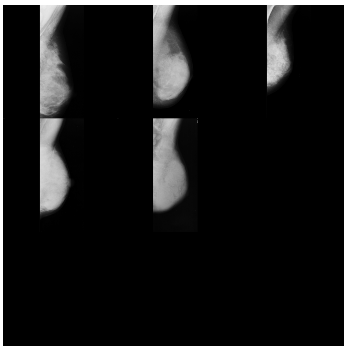
(a) BI-RADS I classification.



(b) BI-RADS II classification.



(c) BI-RADS III classification.

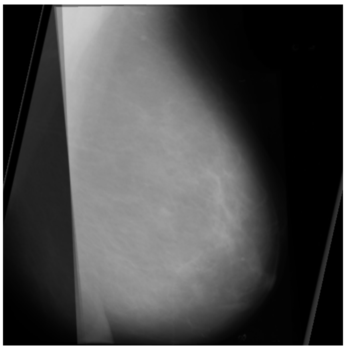


(d) BI-RADS IV classification.

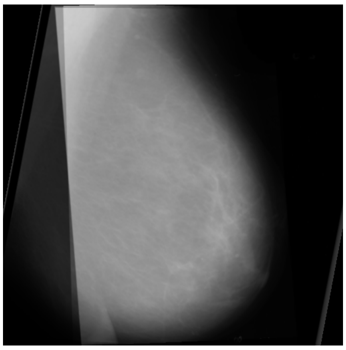
5.2 Shannon Entropy

Sample 1 - BI-RADS I

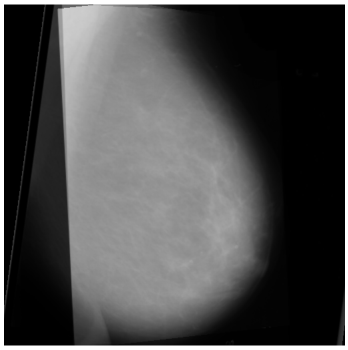
Input image: Figure E.1a



(a) 5 Shannon Entropy iterations.



(b) 10 Shannon Entropy iterations.



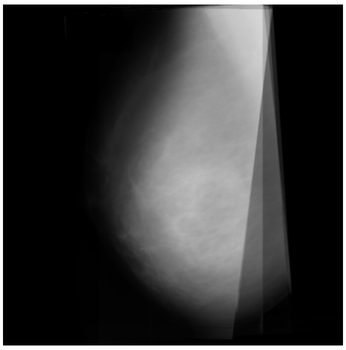
(c) 20 Shannon Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.574292	11	0.551807
2	0.570867	12	0.549876
3	0.568946	13	0.547188
4	0.567278	14	0.545286
5	0.565542	15	0.542942
6	0.562933	16	0.540746
7	0.560522	17	0.539067
8	0.558121	18	0.537744
9	0.555785	19	0.535056
10	0.553746	20	0.534195

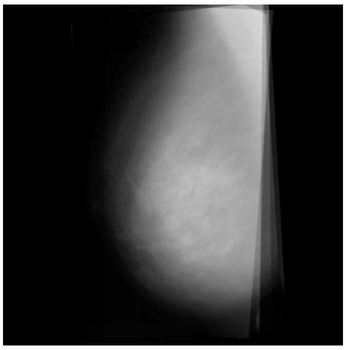
Table E.1: Shannon entropy on Sample 1

Sample 2 - BI-RADS II

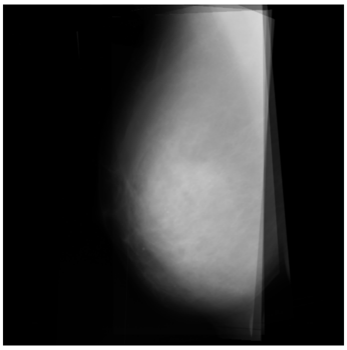
Input image: Figure E.1b



(a) 5 Shannon Entropy iterations.



(b) 10 Shannon Entropy iterations.



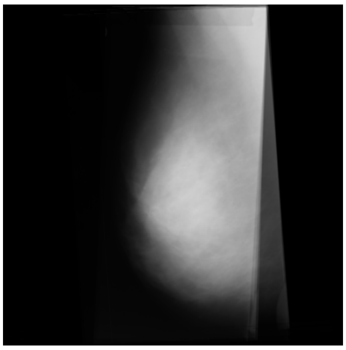
(c) 20 Shannon Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.416080	11	0.360599
2	0.403801	12	0.359308
3	0.393788	13	0.358712
4	0.384148	14	0.359453
5	0.379955	15	0.358421
6	0.374722	16	0.356750
7	0.370704	17	0.354491
8	0.366110	18	0.353710
9	0.363956	19	0.352490
10	0.363712	20	0.350700

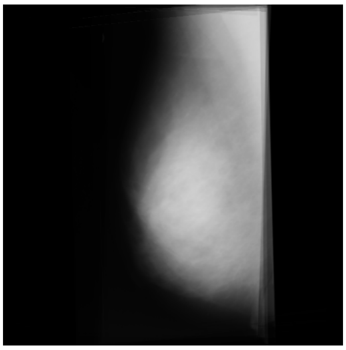
Table E.2: Shannon Entropy on Sample 2

Sample 3 - BI-RADS III

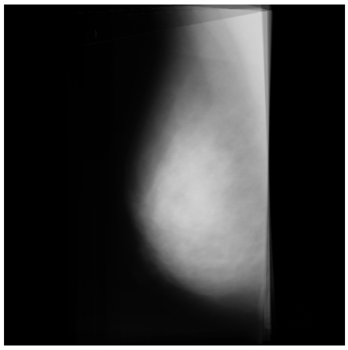
Input image: Figure E.1c



(a) 5 Shannon Entropy iterations.



(b) 10 Shannon Entropy iterations.



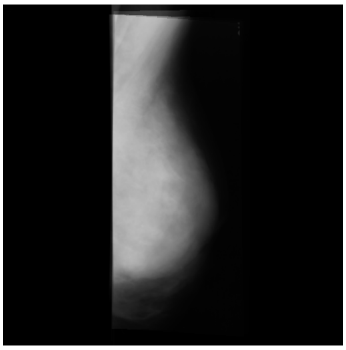
(c) 20 Shannon Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.363914	11	0.321993
2	0.353462	12	0.321014
3	0.347279	13	0.320316
4	0.340452	14	0.319137
5	0.337625	15	0.318285
6	0.333940	16	0.316206
7	0.331973	17	0.315913
8	0.330349	18	0.315580
9	0.325900	19	0.314940
10	0.324482	20	0.315340

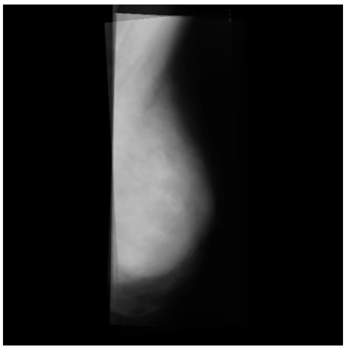
Table E.3: Shannon Entropy on Sample 3

Sample 4 - BI-RADS IV

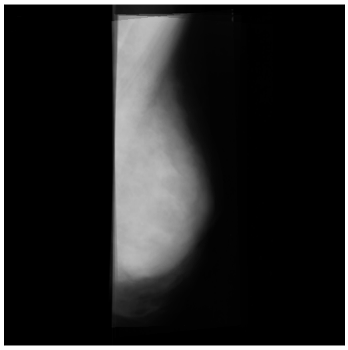
Input image: Figure E.1d



(a) 5 Shannon Entropy iterations.



(b) 10 Shannon Entropy iterations.



(c) 20 Shannon Entropy iterations.

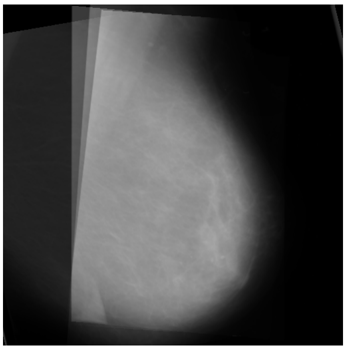
Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.231113	11	0.214690
2	0.225648	12	0.214167
3	0.221902	13	0.212779
4	0.218164	14	0.212419
5	0.216845	15	0.212023
6	0.215778	16	0.210181
7	0.215096	17	0.211036
8	0.215606	18	0.210636
9	0.214506	19	0.209981
10	0.213850	20	0.210216

Table E.4: Shannon Entropy on Sample 4

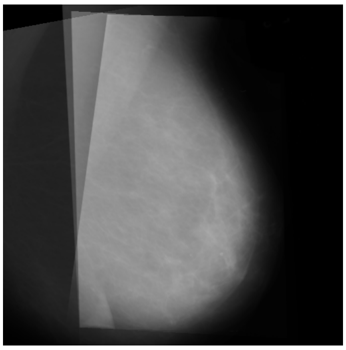
5.3 Non-Probabilistic Entropy

Sample 1 - BI-RADS I

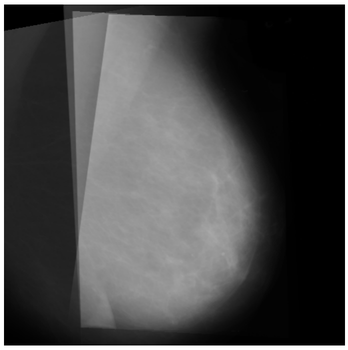
Input image: Figure E.1a



(a) 5 Non-Probabilistic Entropy iterations.



(b) 10 Non-Probabilistic Entropy iterations.



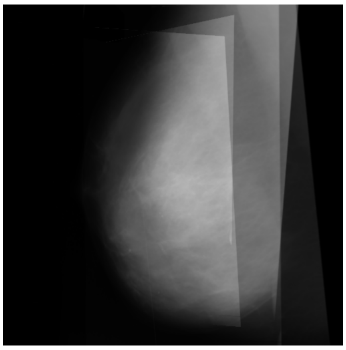
(c) 20 Non-Probabilistic Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.015932	11	0.011260
2	0.013897	12	0.010531
3	0.013986	13	0.011056
4	0.011205	14	0.010350
5	0.012283	15	0.011628
6	0.010841	16	0.010867
7	0.011662	17	0.011381
8	0.010741	18	0.010900
9	0.011279	19	0.011710
10	0.010766	20	0.010889

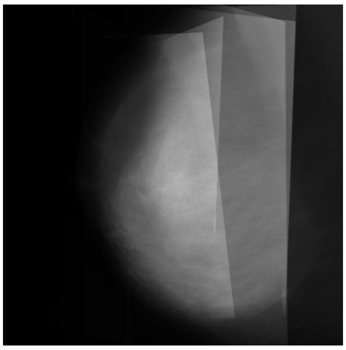
Table E.5: Non-Probabilistic Entropy on Sample 1

Sample 2 - BI-RADS II

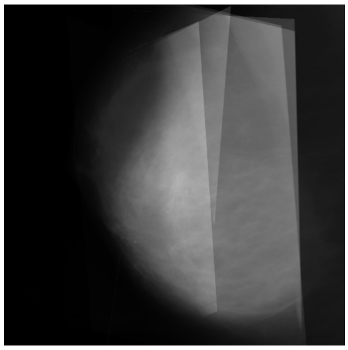
Input image: Figure E.1b



(a) 5 Non-Probabilistic Entropy iterations.



(b) 10 Non-Probabilistic Entropy iterations.



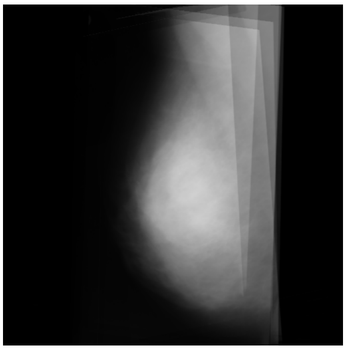
(c) 20 Non-Probabilistic Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.024955	11	0.011755
2	0.018532	12	0.011833
3	0.016395	13	0.011723
4	0.015026	14	0.011706
5	0.014457	15	0.012213
6	0.013762	16	0.011696
7	0.013240	17	0.012007
8	0.012838	18	0.011538
9	0.012955	19	0.011209
10	0.012781	20	0.010888

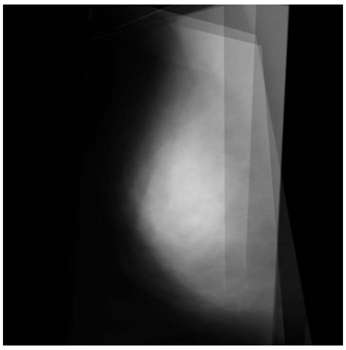
Table E.6: Non-Probabilistic Entropy on Sample 2

Sample 3 - BI-RADS III

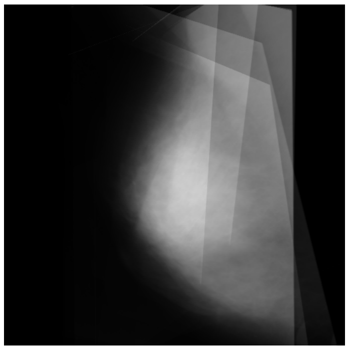
Input image: Figure E.1c



(a) 5 Non-Probabilistic Entropy iterations.



(b) 10 Non-Probabilistic Entropy iterations.

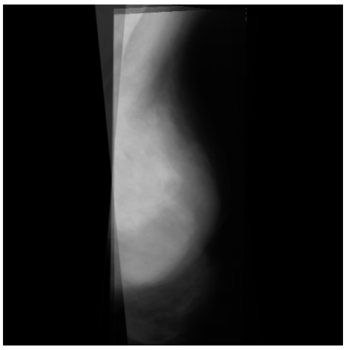


(c) 20 Non-Probabilistic Entropy iterations.

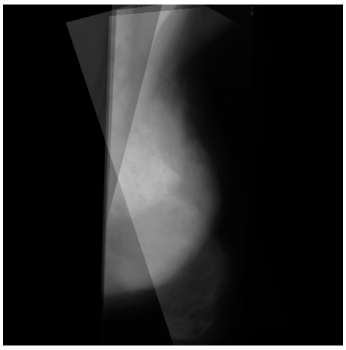
Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.028877	11	0.017761
2	0.024979	12	0.017562
3	0.023324	13	0.017416
4	0.022167	14	0.017499
5	0.021396	15	0.017327
6	0.020770	16	0.017333
7	0.020033	17	0.017191
8	0.019599	18	0.017146
9	0.018735	19	0.016887
10	0.018375	20	0.016796

Table E.7: Non-Probabilistic Entropy on Sample 3

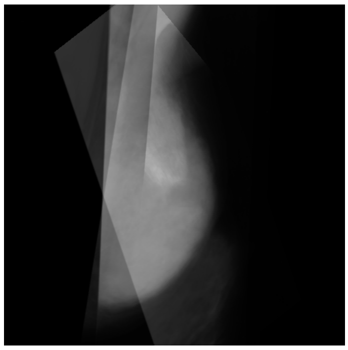
Sample 4 - BI-RADS IV Input image: Figure E.1d



(a) 5 Non-Probabilistic Entropy iterations.



(b) 10 Non-Probabilistic Entropy iterations.



(c) 20 Non-Probabilistic Entropy iterations.

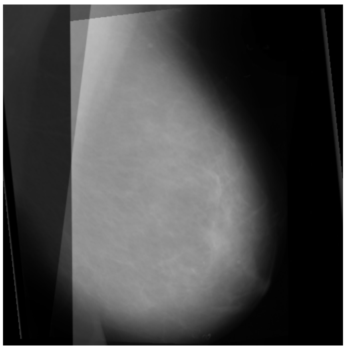
Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.024644	11	0.010979
2	0.023619	12	0.010571
3	0.021987	13	0.009983
4	0.018499	14	0.009071
5	0.019099	15	0.007758
6	0.016766	16	0.007077
7	0.015052	17	0.006823
8	0.013173	18	0.006457
9	0.012420	19	0.006377
10	0.011391	20	0.006101

Table E.8: Non-Probabilistic Entropy on Sample 4

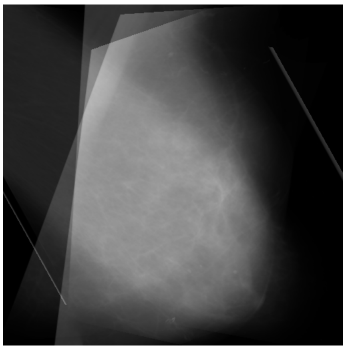
5.4 Hybrid Entropy

Sample 1 - BI-RADS I

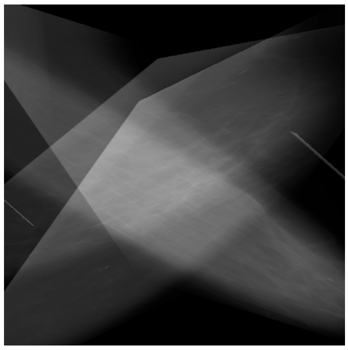
Input image: Figure E.1a



(a) 5 Hybrid Entropy iterations.



(b) 10 Hybrid Entropy iterations.



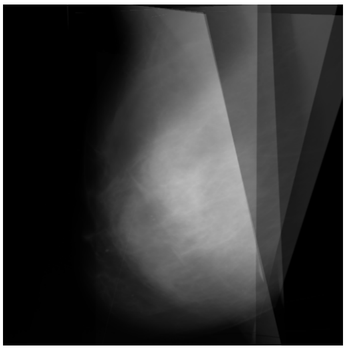
(c) 20 Hybrid Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.193522	11	0.107298
2	0.192257	12	0.097968
3	0.188139	13	0.089378
4	0.182887	14	0.081780
5	0.173737	15	0.075122
6	0.162572	16	0.069405
7	0.151736	17	0.064241
8	0.140438	18	0.059527
9	0.129620	19	0.055169
10	0.117715	20	0.051228

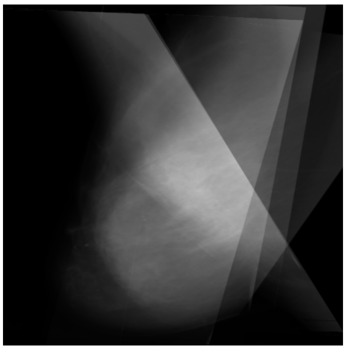
Table E.9: Hybrid Entropy on Sample 1

Sample 2 - BI-RADS II

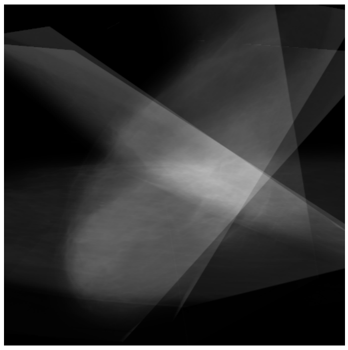
Input image: Figure E.1b



(a) 5 Hybrid Entropy iterations.



(b) 10 Hybrid Entropy iterations.



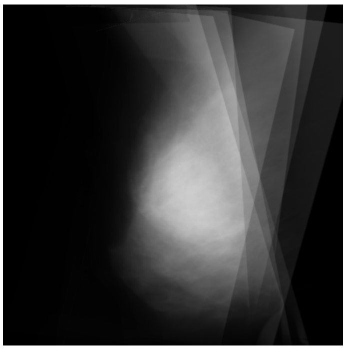
(c) 20 Hybrid Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.089658	11	0.040404
2	0.080077	12	0.038025
3	0.073158	13	0.035813
4	0.067161	14	0.033728
5	0.061344	15	0.031005
6	0.057023	16	0.028583
7	0.053494	17	0.026092
8	0.050325	18	0.024386
9	0.047243	19	0.023165
10	0.044346	20	0.021976

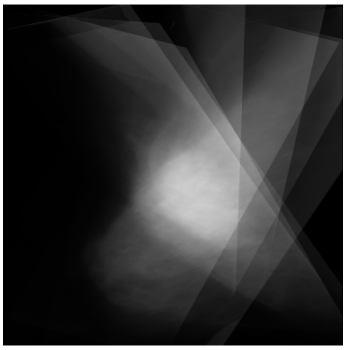
Table E.10: Hybrid Entropy on Sample 2

Sample 3 - BI-RADS III

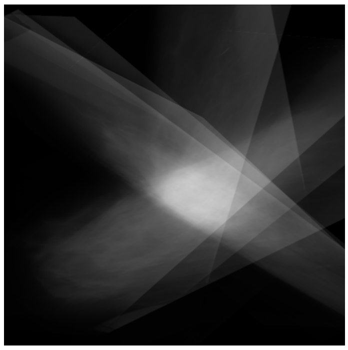
Input image: Figure E.1c



(a) 5 Hybrid Entropy iterations.



(b) 10 Hybrid Entropy iterations.



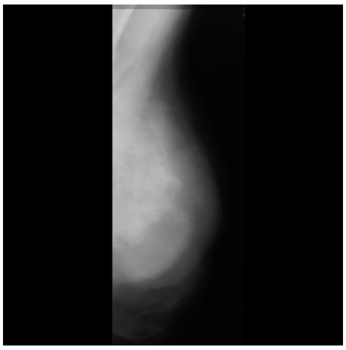
(c) 20 Hybrid Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.093583	11	0.042118
2	0.084049	12	0.040171
3	0.076050	13	0.038293
4	0.067965	14	0.036569
5	0.061740	15	0.034860
6	0.056782	16	0.033347
7	0.052414	17	0.031953
8	0.049218	18	0.030595
9	0.046558	19	0.029245
10	0.044184	20	0.027946

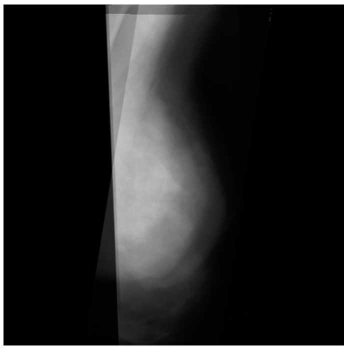
Table E.11: Hybrid Entropy on Sample 3

Sample 4 - BI-RADS IV

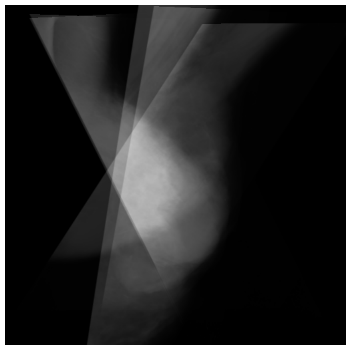
Input image: Figure E.1d



(a) 5 Hybrid Entropy iterations.



(b) 10 Hybrid Entropy iterations.



(c) 20 Hybrid Entropy iterations.

Iteration (1/2)	Entropy (1/2)	Iteration (2/2)	Entropy (2/2)
1	0.062255	11	0.055067
2	0.062107	12	0.053820
3	0.061995	13	0.052123
4	0.061836	14	0.049991
5	0.061708	15	0.042035
6	0.061526	16	0.033866
7	0.060085	17	0.029454
8	0.058960	18	0.026289
9	0.057723	19	0.024111
10	0.056426	20	0.022408

Table E.12: Hybrid Entropy on Sample 4

Glossary

Congealing is an algorithm concerned with joint image alignment, developed by Learned-Miller [42].

CRC Class, Responsibilities, and Collaboration.

DDSM Digital Database for Screening Mammography.

DOG Difference of Gaussian.

GUI Graphical User Interface.

KLT Kanade-Lucas-Tomasi.

MIAS Mammographic Image Analysis Society.

non-deterministic is when a system or algorithm, even if given the same input, can behave in a different manner each time it is executed - i.e. in the Congealing algorithm, there is no set order in which the image transformations can be run, therefore they can be run in a different order each time the images are congealed.

PGM Portable Gray Map.

SCM Software Configuration Management.

SIFT Scale Invariant Feature Transform.

SURF Speeded Up Robust Features.

SVN Apache Subversion.

TDD Test Driven Development.

transposition to change the order of two or more objects.

XP eXtreme Programming.

Annotated Bibliography

[1] [Online]. Available: http://www.merriam-webster.com/dictionary/entropy

Definition of Entropy with regards Information Theory (and thermodynamics) - as defined in the Merriam-Webster dictionary.

[2] "The agile movement." [Online]. Available: http://agilemethodology.org/

Site dedicated to the Scrum and it's principles.

[3] "The anatomy and physiology of the breast." [Online]. Available: http://www.cancer.ca/en/cancer-information/cancer-type/breast/anatomy-and-physiology/?region=on

Website outlining the make up of breast structure - this aided in the author's understanding.

[4] "Definition of enantiomorph." [Online]. Available: http://www.merriam-webster.com/dictionary/enantiomorph

Definition of the name chosen for the application - as defined in the Merriam-Webster dictionary.

[5] "Definition of indiscernibility." [Online]. Available: http://www.merriam-webster.com/dictionary/indiscernibility

Helps define Indiscernibility uncertainty - as defined in the Merriam-Webster dictionary.

[6] "Definition of indiscernible." [Online]. Available: http://www.merriam-webster.com/dictionary/indiscernible

Helps define Indiscernibility uncertainty - as defined in the Merriam-Webster dictionary.

[7] "Definition of possibility." [Online]. Available: http://www.merriam-webster.com/dictionary/possibility

Helps define Possibilistic uncertainty - as defined in the Merriam-Webster dictionary

[8] "Definition of probability." [Online]. Available: http://www.merriam-webster.com/dictionary/probability

Helps define Probabilistic Entropy - as defined in the Merriam-Webster dictionary

[9] "Manifesto for agile software development." [Online]. Available: http://agilemanifesto.org/

The Agile manifesto - it is following these values the author tried to build the application.

[10] "Masking out image area using binary mask - matlab answers - matlab central." [Online]. Available: http://uk.mathworks.com/matlabcentral/answers/38547-masking-out-image-area-using-binary-mask

Chosen method for removing Medical Markers - draw an outline, create a mask and overlay the mask over the drawn area.

[11] "Taiga - love your projects." [Online]. Available: https://taiga.io/

Tool used to promote a scrum methodology as used by the author.

[12] "Want to remove white patches of a image - matlab answers - matlab central." [Online]. Available: http://uk.mathworks.com/matlabcentral/answers/141008-want-to-remove-white-patches-of-a-image

Thread on removing white objects/glare from an image - useful for removing medical markers from mammograms.

[13] "Basic information about the typical mlo and cc views of mammography," Apr 2016. [Online]. Available: http://breast-cancer.ca/mammopics/

An informal website outlining some information about MLO and CC breast scans.

[14] I. C. about Cancer Screening, "Nhs breast screening: Helping you decide," Jun 2013. [Online]. Available: http://www.breasttestwales.wales.nhs.uk/sitesplus/documents/1025/1554%20Yogi%20-%20Breast%20Screening%20English%20v4.pdf

Leaflet about Breast Screening created by Informed Choice about Cancer Screening with Cancer Research UK. The leaflet helped the author understand the options currently open to women of mammography-screening age.

[15] S. Al-Sharhan, F. Karray, W. Gueaieb, and O. Basir, "Fuzzy entropy: a brief survey," in *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 1135–1139. [Online]. Available: http://dx.doi.org/10.1109/fuzz.2001.1008855

Paper outlining the different implementations of Fuzzy Entropy, of which 2 will be selected and focused on during this Project. It provides a general overview of both algorithms, and clearly explains all derivations needed for implementation. Due to the age of the algorithms implemented, the papers associated with them were difficult to locate and once obtained, that this paper above provided the main gist of the original papers anyway.

[16] H. Bay, T. Tuytelaars, and L. V. Gool, *SURF: Speeded Up Robust Features*. Springer Berlin Heidelberg, May 2006, p. 404417. [Online]. Available: http://link.springer.com/chapter/10.1007/11744023_32

Feature extraction method which could be utilised in the future for the automated removal of medical markers.

[17] N. F. Boyd, J. W. Byng, R. A. Jong, E. K. Fishell, L. E. Little, A. B. Miller, G. A. Lockwood, D. L. Tritchler, and M. J. Yaffe, "Quantitative classification of mammographic densities and breast cancer risk: Results from the canadian national breast screening study," *Journal of the National Cancer Institute*, vol. 87, no. 9, p. 670675, May 1995, pMID: 7752271.

This paper clearly outlines significant links between breast density and breast cancer risk. Whilst other papers are available to outline the connection, this paper provided a clear insight into the methods in which they conducted the experiments.

[18] J. L. Champaign and G. J. Cederbom, "Advances in breast cancer detection with screening mammography," *The Ochsner Journal*, vol. 2, no. 1, p. 33, Jan 2000, pMID: 21765659.

Paper outlining the advancements in screening mammography in the years up to 2000.

[19] M. Cohn, "User stories and user story examples by mike cohn." [Online]. Available: https://www.mountaingoatsoftware.com/agile/user-stories

User Story example given in the Software section

[20] L. Collins, "Hurdle #4 - removing labels from mammograms," Mar 2016. [Online]. Available: http://lauramcollins.co.uk/blog/removing-labels-from-mammograms/

My personal blog post containing information about the methods in which I tried to remove Medical Markers from the data set of mammograms.

[21] E. Commission, "Health statistics - atlas on mortality in the european union - product - eurostat," 2009. [Online]. Available: http://ec.europa.eu/eurostat/en/web/products-statistical-books/-/KS-AC-04-000

2009 paper published by the European Commission on statistics into mortality rates and causes in the EU. This paper helped to formalise accurate knowledge into the area of breast cancer

[22] T. F. Cootes, C. J. Twining, V. S. Petrovic, K. O. Babalola, and C. J. Taylor, "Computing accurate correspondences across groups of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, p. 19942005, Nov 2010.

Another example of facial image alignment, plus medical imagery alignment using a groupwise registration algorithm.

[23] M. Cox, S. Sridharan, S. Lucey, and J. Cohn, *Least squares congealing for unsupervised alignment of images*, Jun 2008, p. 18.

A disregarded adaption of the Congealing algorithm - however was useful in highlighting performance issues in the original algorithm. Something which was near continuously faced when implementing heavier fuzzy entropy alignment metrics such as Non-Probabilistic entropy.

[24] A. De Luca and S. Termini, "A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory," *Information and Control*, vol. 20, no. 4, p. 301312, May 1972.

De Luca & Termini's 1972 paper focuses on their definition of Non-probabilistic entropy, along with several example derivations. This paper was useful for background information to their work, however Al-Sharhan's paper provides a more concise explanation of the algorithm itself.

[25] debonet. [Online]. Available: https://github.com/debonet/Congealing

C++ implementation of the Congealing algorithm available on Github

[26] M. H. DeGroot, *Optimal Statistical Decisions*. Reprinted: McGraw-Hill, 2004.

DeGroot's paper gives a simple explanation about uncertainty in terms of probability, first coined by Savage. His paper led the author onto finding Savage's original work on probabilistic uncertainty.

[27] M. Elter, R. Schulz-Wendtland, and T. Wittenberg, "The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process." vol. Medical Physics 34(11), p. 41644172, 2007.

Mammogram dataset focusing purely upon scans containing masses - therefore not suitable for this project.

[28] K. GEORGE J. and Y. BO, "Fuzzy sets and fuzzy logic, theory and applications," -, Mar 2008. [Online]. Available: http://digilib.uin-suka.ac.id/7049/

This paper provides a clear explanation of the differences between crisp and fuzzy sets. Their diagram, as utilised in this document, is a simple example of how temperature works better within fuzzy sets, then that of crisp.

[29] Git. [Online]. Available: https://git-scm.com/about/small-and-fast

Statistics on git runtimes

[30] Github. [Online]. Available: github.com

Github is an online tool for Git repositories as utilised by the author.

[31] ——. [Online]. Available: https://desktop.github.com/

Github desktop tool utilised during this project by the author.

[32] github.com, "Git's GPL license at github," Jan. 2014.

Git was utilised for version control in this project

[33] I. T. Gram, E. Funkhouser, and L. Tabár, "The Tabár classification of mammographic parenchymal patterns. - pubmed - ncbi." [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/9097055

Tabár method for categorising scans using an anatomic-mammographic comparison technique. The author decided not to work with this classification, as the BI-RADS system is often favoured by practitioners.

[34] H. Gray, *Anatomy, Descriptive and Surgical*. Philadelphia: Lea & Febiger, 1907, 1-58734-102-6.

Extremely influential book first written in 1858 by Henry Gray. It has been revised and republished, up until September 2015. Whilst this cited version is not 100% medically accurate by today's standards, the diagrams still hold true. The diagram, as used in the main body of this text, is still accurate due to the slow evolution of the human body.

[35] C. Harris and M. Stephens, *A combined corner and edge detector*, 1988. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.231.1604

An early corner and edge detector - which could be useful for the automatic detection of rectangular medical markers.

[36] M. Heath, K. Bowyer, D. Kopans, R. Moore, P. Kegelmeyer, and C. Processing, *THE DIGITAL DATABASE FOR SCREENING MAMMOGRAPHY*.

An example of an open mammogram database, not utilised during this project due to the large nature of the images provided - First citation needed for the DDSM database.

[37] M. Heath, K. Bowyer, D. Kopans, W. P. Kegelmeyer, R. Moore, K. Chang, and S. MunishKumaran, "Current status of the digital database for screening mammography," *Proceedings of the Fourth International Workshop on Digital Mammography*, no. 457–460, 1998. [Online]. Available: http://marathon.csee.usf.edu/Mammography/software/HeathEtAlIWDM_1998.pdf

An example of an open mammogram database, not utilised during this project due to the large nature of the images provided - Second citation needed for the DDSM database

[38] G. B. Huang, V. Jain, and E. Learned-Miller, *Unsupervised Joint Alignment of Complex Images*. IEEE, Oct 2007, p. 18. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4408858

Further work done upon Congealing by a team including Learned-Miller. This was disregarded for the project due to the complexity added for inputting complex images - something mammograms are not.

[39] S. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, no. 3, 2007.

Paper outlining the advantages and disadvantages of machine learning classification techniques.

[40] C. Kurniawan, "Detect square in image." [Online]. Available: http://uk.mathworks.com/matlabcentral/answers/24943-detect-square-in-image

Mathworks forum thread on how to detect squares in an image using a morphological operation. This was investigated for possible use in medical marker removal.

[41] E. Learned-Miller, "The congealing page." [Online]. Available: https://people.cs.umass.edu/~elm/congealing/

This webpage was the source of the demo Congealing code modified and built upon in this project. It also contains several useful papers where Congealing has been used in research - such as Learned-Miller's own paper on Congealing MNIST handwriting data and MRI scans.

[42] E. G. Learned-Miller, "Data driven image models through continuous joint alignment," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 2, pp. 236–250, Feb. 2006. [Online]. Available: http://dx.doi.org/10.1109/tpami.2006.34

Learned-Miller's original Congealing method is the basis for this Project - however the application implemented will be looking to further extend the alignment capabilities using fuzzy entropy metrics, rather than standard Shannon entropy as currently implemented. This paper was extremely useful for understanding of the basic concepts behind it, and will be a good reference guide throughout the project.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

MNIST handwriting dataset was utilised by Learned-Miller in the original Congealing algorithm. This however was not useful for our fuzzy entropy implementations due to the binary nature of the images (either white or black pixels).

[44] F. Li, Q. Shen, Y. Li, and N. M. Parthaláin, *A fuzzy image congealing-based handwritten Chinese character recognition and classification system.* IEEE, Sep 2014, p. 18. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6930166

Paper building upon work of that of Strange and Mac Parthaláin. Concludes that similarity relation fuzzy entropy may be too computationally costly for a standard application.

[45] D. G. Lowe, *Object Recognition from Local Scale-Invariant Features.* IEEE Computer Society, Sep 1999, p. 1150. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

Feature detector upon which SURF was built. SURF is being considered for the automated removal of medical markers.

[46] B. D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision.* Morgan Kaufmann Publishers Inc., Aug 1981, p. 674679. [Online]. Available: http://dl.acm.org/citation.cfm?id=1623264.1623280

Lucas and Kanade approach to aligning images using gradient-descent - this is what Cox et al based their Least Squares Congealing algorithm off of.

[47] N. Mac Parthalain and H. Strange, *Fuzzy-entropy based image congealing*, Jul 2013, p. 18.

Paper outlining a Fuzzy Entropy congealing implementation - something more computationally costly than this project desires due to the automatic generation of membership functions.

[48] MathWorks, "fwite - write data to binary file." [Online]. Available: http://uk.mathworks.com/help/matlab/ref/fwrite.html

Documentation for the fwrite function which writes binary data to file. As it writes the data column-wise, this was causing a transposition issue initially.

[49] ——, "Using matlab with other programming languages." [Online]. Available: http://uk.mathworks.com/solutions/matlab-and-other-programming-languages/?refresh=true

Initially the author considered using MATLAB in conjunction with another programming language, this page provided information on how to do so.

[50] ——, "Fuzzy logic toolbox users guide," Mar 2016. [Online]. Available: http://cn.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf

Fuzzy Logic toolbox utilised in this project to calculate fuzzy sets.

[51] ——, "Image processing toolbox users guide," Mar 2016. [Online]. Available: http://it.mathworks.com/help/pdf_doc/images/images_tb.pdf

Image processing toolbox used to handle images

[52] MATLAB, "Guide or programmatic workflow." [Online]. Available: http://uk.mathworks.com/help/matlab/guide-or-matlab-functions.html

GUIDE was used to create the GUIs in the project

[53] ——, "imfreehand - create draggable freehand region." [Online]. Available: http://uk.mathworks.com/help/images/ref/imfreehand.html

Documentation about a tool the user utilises to draw freehand the area they wish to remove from the scan.

[54] ——, "permute - rearrange dimensions of n-d array." [Online]. Available: http://uk.mathworks.com/help/matlab/ref/permute.html

Documentation for the transposition of ND arrays (arrays greater than 2-dimensions.)

[55] ——, "regionprops - measure properties of image regions." [Online]. Available: http://uk.mathworks.com/help/images/ref/regionprops.html

Documentation about a possible solution for removing medical markers from mammograms.

[56] ——, "Transpose vector or matrix - matlab - mathworks united kingdom." [Online]. Available: http://uk.mathworks.com/help/matlab/ref/transpose.html?refresh=true

Transpose function was leveraged to flip arrays being fed into fwrite.

[57] ——, "Vectorization." [Online]. Available: http://uk.mathworks.com/help/matlab/matlab_prog/vectorization.html?refresh=true

Vectorisation was extremely useful in this project for reducing the run time of the Non-probabilistic entropy and membership functions. This was a new concept to myself.

[58] ——, *version 9.0.0.341360 (R2016a)*. Natick, Massachusetts: The MathWorks Inc., 2016.

MATLAB citation

[59] Oracle, "1 javafx overview (release 8)." [Online]. Available: http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784

Initially the author planned to implement the GUI using JavaFX, as they were familiar with the programming language. However this idea was later discarded in favour of programming all the elements in one language - MATLAB

[60] A. OHagan, "Probabilistic uncertainty specification: Overview, elaboration techniques and their application to a mechanistic model of carbon flux," 2011. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.319.9570

A paper outlining a way in which to apply probabilistic uncertainty to an engineering problem. The opening sections of the paper give a detailed explanation of probabilistic uncertainty, enough to glean enough understanding to write a detailed analysis and apply it to my own case.

[61] N. R. Pal and S. K. Pal, "Object-background segmentation using new definitions of entropy," *IEE Proceedings E - Computers and Digital Techniques*, vol. 136, no. 4, p. 284295, Jul 1989.

Early fuzzy entropy derivation - similar to that of De Luca & Termini.

[62] ——, "Higher order fuzzy entropy and hybrid entropy of a set," *Information Sciences*, vol. 61, no. 3, p. 211231, Jun 1992.

Pal and Pal's paper outlining both Hybrid Entropy and Higher-order entropy. Hybrid entropy was implemented following their 2 fuzzy-set approach, to ensure that inversion is preserved (sets A and B: A = x and B = 1 - A). Pal and Pal's paper was extremely useful in terms of implementation, as at times their explanation was much clearer than that of Al-Sharhan.

[63] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2233–2246, 2012.

Alternative Joint Image alignment technique. This idea was discarded due to the wasted computational calculations that would be generated thanks to this project's dataset being overly simplistic compared to their benchmark work.

[64] D. Peterson, "What is kanban?" [Online]. Available: http://kanbanblog.com/explained/

Blog outlining Kanban - something not utilised specifically in this project.

[65] J. Poskanzer, "Pgm format specification." [Online]. Available: http://netpbm.sourceforge.net/doc/pgm.html

Webpage outlining the header file of pgm images.

[66] Radswiki, "Mammography views — radiology reference article — radiopaedia.org." [Online]. Available: http://radiopaedia.org/articles/mammography-views

Detailed online encyclopedia outlining the different types of mammogram scan which can be performed and their benefits/weaknesses.

[67] A. Samieh, "Shape recognition - file exchange - matlab central," 2016. [Online]. Available: http://uk.mathworks.com/matlabcentral/fileexchange/15491-shape-recognition

Demonstration on shape recognition which was investigated for the removal of medical markers from the mammograms.

[68] W. Sander, "On measures of fuzziness," *Fuzzy Sets and Systems - FSS*, vol. 29, no. 1, p. 4955, 1989.

Presents a characterisation of a fuzzy entropy - decided not to be included in the project due to the lack of probabilistic modeling.

[69] L. J. Savage, *The Foundations of Statistics*, 1954. [Online]. Available: https://books.google.co.uk/books/about/The_Foundations_of_Statistics.html?id=zSv6dBWneMEC

Savage worked in the area of probabilistic uncertainty. His work was later built upon by DeGroot and his explanation of probabilistic uncertainty is simple for the average man to understand.

[70] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal 27*, 1948.

Paper outlining Shannon's work which resulted in Shannon Entropy (also refered to as standard entropy in this project).

[71] J. Shi and C. Tomasi, *Good features to track*. IEEE, Jun 1994, p. 593600. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=323794

One of two papers required for the citation of KLT features by Kanade-Lucas-Tomasi. Their work could be leveraged in the automatic detection of medical markers in future work on this project.

[72] A. SHIRALIGE, "Agile estimation:8 steps to successful story point estimation." [Online]. Available: http://www.agilebuddha.com/agile/agile-estimation-8-steps-to-successful-story-point-estimation/

Blog discussing the benefits of consistent estimation and how being accurate in estimation is not necessarily as important.

[73] E. Sickles, D. CJ, B. LW, *et al.*, *ACR BI-RADS® Mammography*. In: ACR BI-RADS® Atlas, Breast Imaging Reporting and Data System, 2013.

BI-RADS breast density classification is mostly commonly used today in the medical industry. Therefore it became the focus of this project, and allows classification of the Mini-MIAS dataset.

[74] M. . Simulink, "Testing frameworks." [Online]. Available: http://uk.mathworks.com/help/matlab/matlab-unit-test-framework.html

MATLAB's unit testing documentation.

[75] Sourceforge, "Octave-forge." [Online]. Available: http://octave.sourceforge.net/

Alternative open-source implementation tool to MATLAB.

[76] J. Suckling, "The mammographic image analysis society digital mammogram database," *Exerpta Medica International Congress Series*, vol. 1069, 1994.

> The Mini-MIAS dataset which was used during this project. These images have been reduced in size to allow a more sensible computation time when compared to the orignal MIAS scans, however this will have caused a degredation in the information available in each image. This loss was not a major issue in the project due to it being a proof-of-concept, rather than a finalised tool.

[77] B. Tiddeman and D. Hunter, "Groupwise non-rigid image alignment - results of a multiscale iterative best edge algorithm," Mar 2011. [Online]. Available: http://cadair.aber.ac.uk/dspace/handle/2160/11857

> Paper leveraging image alignment via groupwise registration for aligning facial images.

[78] C. Tomasi and T. Kanade, "Detection and tracking of point features," *Pattern Recognition 37*, pp. 165–168, 2004.

> One of two papers required for the citation of KLT features by Kanade-Lucas-Tomasi. Their work could be leveraged in the automatic detection of medical markers in future work on this project.

[79] C. R. UK, "Breast cancer tests," Apr 2015. [Online]. Available: http://www.cancerresearchuk.org/about-cancer/type/breast-cancer/diagnosis/breast-cancer-tests

> Webpage outlining the different types of breast scan which are available, including Mammogram via MRI, Ultrasound and needle biopsy. This was important to understand the issues faced by practictioners when it comes to a choice of different breast scanning techniques.

[80] E. Untiedt, *A Parametrized Model for Optimization with Mixed Fuzzy and Possibilistic Uncertainty*. Springer Berlin Heidelberg, 2010. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-13935-2_9 978-3-642-13934-5.

> Good description of possibilistic uncertainty versus fuzzy (possibilistic) uncertainty.

[81] D. Wells, "Extreme programming: A gentle introduction." [Online]. Available: http://www.extremeprogramming.org/

> Practices such as CRC cards were leveraged from XP during this projects

[82] ——, "XP - acceptance testing." [Online]. Available: http://www.extremeprogramming.org/rules/functionaltests.html

> Page outlining XP's stance on Acceptance testing - a user story is not considered complete until it has passed it's acceptance test.

[83] J. N. Wolfe, "Risk for breast cancer development determined by mammographic parenchymal pattern," *Cancer*, vol. 37, no. 5, p. 24862492, May 1976.

Paper outlining Wolfe's classification of breast tissue density. The author decided not to work with this classification, as the BI-RADS system is often favoured by practitioners.

[84] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, p. 338353, Jun 1965.

Zadeh's work into Fuzzy Set Theory paved the way for Fuzzy Entropy as implemented in this project.

[85] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros, *FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences*. IEEE, Jun 2015, p. 11911200. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=7298723

Alternative Joint Image alignment technique. This paper was cited due to the reference to Learned-Miller's prior work - the Congealing algorithm.