

BREAST IMAGE CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

Aaron Mikaelian

Bachelor of Engineering
Computer Engineering

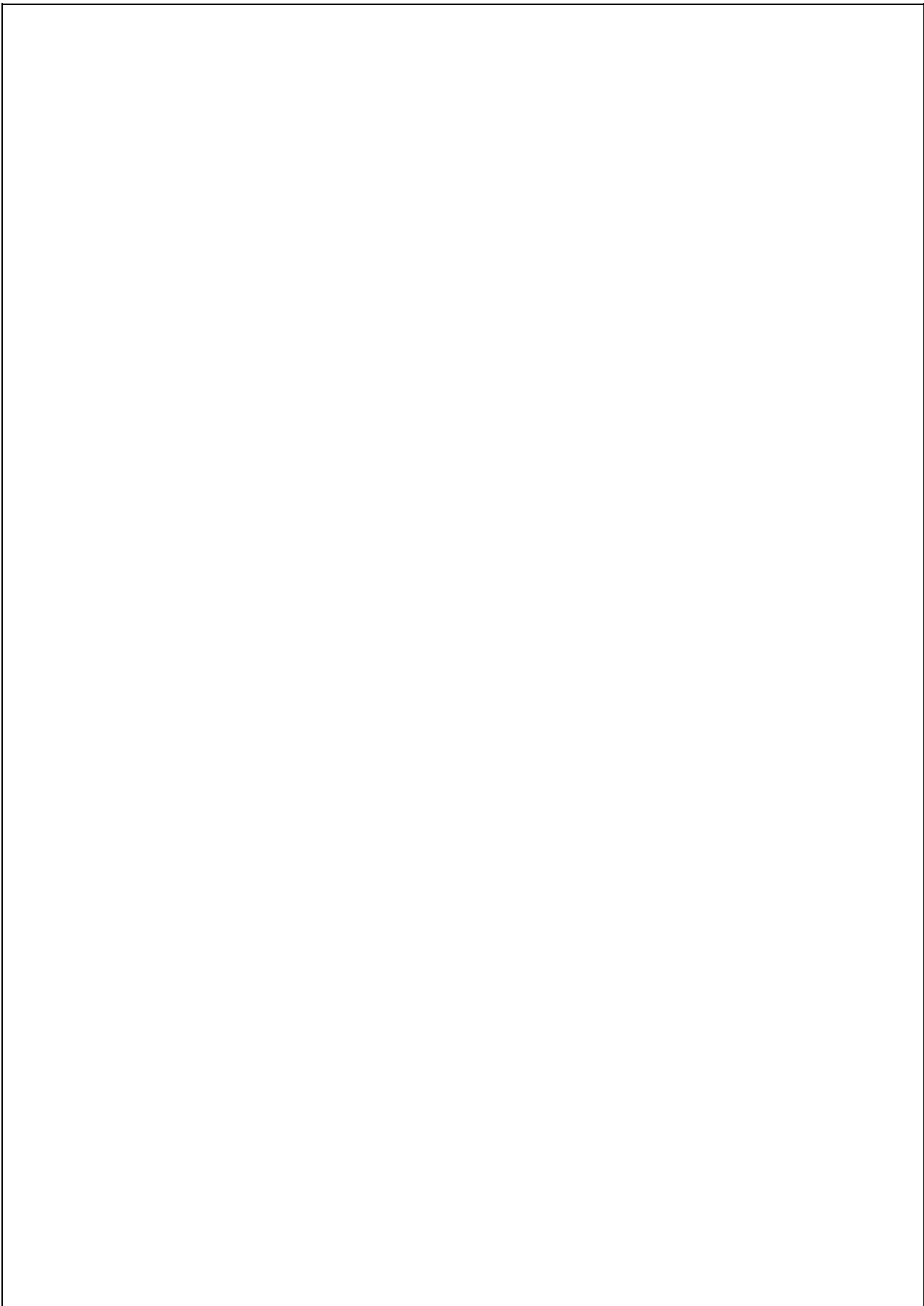


MACQUARIE
University

Department of Electronic Engineering
Macquarie University

November 7, 2016

Supervisor: Dr Yinan Kong



ACKNOWLEDGMENTS


I would like to thank Dr Yinan Kong & Mr Abdullah-Al Nahid, for their help and guidance.



STATEMENT OF CANDIDATE

I, Aaron Mikaelian, declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the Department of Computer Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment at any academic institution.

Student's Name: Aaron Mikaelian

Student's Signature: 

Date: November 7, 2016



ABSTRACT

Proper diagnosis of the breast images may be performed through a combination of Image Processing and Machine Learning techniques. Provided an accurate diagnosis, the time in which a doctor would personally analyse the images is reduced.

Practically speaking, the diagnosis process consists of Feature Detection and Classification. This research project investigates the performance of Tamura's Texture Analysis in combination with Support Vector Machines (SVM), as a means to carry out this process. Seeing as the provided data consists of three classes; Normal, Benign and Cancer, multiclass SVM techniques will be investigated.



Contents

Acknowledgments	iii
Abstract	vii
Table of Contents	ix
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Project Outcomes and Impact	1
1.2 Thesis Overview	2
1.3 Project plan	3
2 Literature Review	5
2.1 Tamura's Features	5
2.1.1 Coarseness	6
2.1.2 Contrast	7
2.1.3 Directionality	7
2.1.4 Line-likeness	9
2.1.5 Regularity	9
2.1.6 Roughness	10
2.2 Support Vector Machine	10
2.2.1 Two Class Support Vector Machines	10
2.2.2 Multiclass Support Vector Machines	12
2.2.3 Unclassifiable Regions	15
3 Mammographic Images	19
3.0.1 Pre-processing	19
4 Tamura's Features	21
4.1 Coarseness	21
4.2 Contrast	23

4.3	Directionality	25
4.4	Line-likeness	26
4.5	Regularity	28
4.6	Roughness	29
4.7	Chapter Summary	30
5	Feature Data	31
5.1	Feature Data Representation	31
5.2	Feature Analysis	31
5.3	Feature Data Distribution	33
5.3.1	Train, Test Distribution	33
5.3.2	Training Pairs	33
5.3.3	Test Set	34
6	Support Vector Machines	37
6.1	Two Class Support Vector Machine	37
6.2	Multiclass Support Vector Machine	38
6.2.1	Standard Pairwise Support Vector Machine	38
6.2.2	Fuzzy Support Vector Machine	39
6.2.3	Decision Tree Based Support Vector Machine	41
7	Results & Discussion	43
7.1	Feature Optimization	43
7.1.1	Coarseness	43
7.1.2	Contrast	44
7.1.3	Directionality	44
7.1.4	Line-likeness	45
7.1.5	Regularity	45
7.1.6	Roughness	45
7.1.7	Optimal Set	46
7.2	Two Class Performance	48
7.3	Multiclass Performance	50
8	Conclusion	51
9	Future Work	53
10	Abbreviations	55
A	Scheduling & Planning	57
A.1	Project Timeline	57
A.2	Consultation Meetings Attendance Form	58
	Bibliography	58

List of Figures

2.1	Original OAA Design [1]	12
2.2	Pairwise Design [1]	14
2.3	Fuzzy SVM design [1]	16
2.4	Decision Tree [1]	17
3.1	Image Crop, Before and After	20
4.1	Comparing Neighbours with Circular Shift	22
4.2	Greyscale Intensity Levels for Breast Image	24
4.3	Histogram pHist for Sample Image	26
5.1	Sample PDF Curves with FDR Measure	32
7.1	PDF Curves for Optimal Feature Models	47
A.1	Consultation Meetings Attendance Form	58



List of Tables

3.1	Mammographic Images: Dimension and Quantity	19
3.2	Cropped Mammographic Images: Dimension and Quantity	20
5.1	Feature Data Arrangement	31
6.1	Confusion Matrix	38
7.1	Fischer's Discriminant Results for Class Pairs: Coarseness	44
7.2	Fischer's Discriminant Results for Class Pairs: Directionality	44
7.3	Fischer's Discriminant Results for Class Pairs: Regularity	45
7.4	Fischer's Discriminant Results for Class Pairs: Roughness	46
7.5	Fischer's Discriminant Results for Class Pairs: Optimal Feature Set	46
7.6	Two Class SVM Performance: Kernel Types	48
7.7	Confusion Matrix: NB	49
7.8	Confusion Matrix: CN	49
7.9	Confusion Matrix: BC	49
7.10	Multiclass SVM Performance	50



Chapter 1

Introduction

In 2014 one in eight Australian women were diagnosed with breast cancer before they had turned 85 [2]. Through early detection of cancer, the survival rates of cancer patients is positively influenced. Therefore, as a result of scheduled mammograms, the effects of Cancer can be reduced.

This project will specifically look at reducing the time frame between the breast scan being taken, and the classification of the generated images. Traditionally a doctor or professional would personally analyse the breast image in order to diagnose the patient. Through a process of feature detection and classification, we can effectively analyse the breast images. Although the process may not provide an absolute classification, it will serve to aid the decision making process undertaken by the healthcare professional.

Tamura's set of six features will be investigated as a means to detect features within images. In order to classify the feature data in an effective manner, a Support Vector Machines (SVM) will be used. Traditionally, a SVM makes the distinction between two classes. Multi-class SVM approaches will be investigated to cater for the three classes involved in this project, Normal, Benign and Cancer.

The aforementioned processes will be applied to images provided Macquarie University, sourced from Aachen University, Germany.

1.1 Project Outcomes and Impact

The main outcome of this project is to assess the effectiveness of Tamura's features in the classification of breast images. The performance of which, will determine the usefulness of the project as a potential aid for healthcare professionals. The following outcomes are to be achieved in order to reach such a conclusion:

- (i) a literature review to establish the relevant theoretical models and processes
- (ii) preparation of mammographic images for processing
- (iii) practical implementation of Tamura's features

- (iv) establish tools required to analyze extracted feature data
- (v) optimization of feature extraction models
- (vi) practical implementation of two class and multiclass support vector machine solutions
- (vii) assessment of support vector machine solutions

Through the fulfillment of the outcomes, the feasibility of Tamura's features in the context of breast image classification may be assessed. It should be noted that such an assessment is specific to the experimental processes outlined in this report. The effectiveness of Tamura's features as a feature extraction model is not conclusively rated for all applications.

1.2 Thesis Overview

The entire project report may be divided into four main categories: Research, Experimental Process, Results & Discussion and Conclusion

Research

A literature review is carried out as to provided context to the project, and outline the theory that is to be practically implemented.

The six features outlined in Hideyuki Tamura's 'Textural Features Corresponding to Visual Perception' are explored in detail [3]. What each of the six features aim to represent is presented both qualitatively, and quantitatively — in the form of computational models.

The theory behind the Support Vector Machine is investigated. Predominantly, the way in which data is used to train a model which may later used in a predictive manner. Two class and multiclass support vector machine solutions are explored.

Experimental Process

The experimental process is broken down into the following stages:

- (i) mammographic image preprocessing,
- (ii) practical implementation of Tamura's features,
- (iii) feature data representation, analysis and distribution,
- (iv) and two class and multiclass support vector machine implementations.

Results & Discussion

The results of the experimental process are appropriately presented as to provide relevant information. The benefits and drawbacks of alternative feature models is quantitatively

presented. Classification rates of two class and multiclass support vector machines solutions are presented as to assess the effectiveness of Tamura's features in the context of this project.

The Results are discussed to better understand what each result represents, and why such a result may have occurred.

Conclusion

An overview of what has been accomplished throughout the project is stated. Results which conclusively assess the project's performance are stated. The feasibility of the project in the context of a breast classification aid is assessed.

1.3 Project plan

According to the Project Timeline in Appendix A.1, the milestones laid out in section 1.2 were allocated periods of time. The thirteen weeks available to undertake this project were appropriately delegated among tasks according to the expected level of work required for each particular task.

Regular meetings have taken place, providing a means of guidance and feedback. See Appendix A.2.

Chapter 2

Literature Review

This chapter will explore literature which provide information relevant to this project. The information gathered will serve as the basis for practical applications of the presented theory, as demonstrated in following chapters. The large majority of the literature covered in this section will investigate two areas, Tamura's Features and Support Vector Machines. Hideyuki Tamura's 'Textural Features Corresponding to Visual Perception' and Shigeo Abe's 'Support Vector Machines for Pattern Classification' will cover these areas, respectfully.

2.1 Tamura's Features

There exist various techniques for detecting features within a segment of an image. These techniques are based upon the attributes they are trying to detect.

For the Breast Image Classification project, Tamura's texture analysis technique will be used as a basis for detection of features. This technique uses computational approximation to form features that correspond to human visual perception - coarseness, contrast, directionality, line-likeness, regularity and roughness [3]. Hideyuki Tamura et al introduced these characteristics in 1978, as an alternative to the commonly used edge detection and regional analysis techniques - which were considered too basic to effectively process complicated regions.

Psychological tests were performed, in which human subjects were asked to rank sixteen typical texture patterns according to the six hypothesised qualitative features. Computational models were then created to best replicate the test results.

The mathematical representations of each feature were applied to the sixteen aforementioned texture patterns, and the correspondence between psychological and computational results were calculated. It was concluded that features - coarseness, contrast and directionality, were considered 'successful' in providing the desired correspondence between psychological and computational results. The remaining three features were considered below satisfactory. In addition to the sub-par correspondence rates of the remaining features, there existed strong positive correlation to the 'successful' features, therefore further reducing their usefulness [3].

The following subsections will explore what each feature is attempting to represent in qualitative terms, and their mathematical representations. The mathematical models aim to generate values in a range such that, the maxima and minima represent the qualitative extremes of the feature in question. For example, [0 to 1] \iff [Smooth to Rough], for a measure of Roughness.

2.1.1 Coarseness

Coarseness is a measure of how coarse or fine a texture is. The measure is directly affected by scale, and the repetition rate of elements within that range. Coarseness aims to identify the largest size at which a texture exists, whilst in the presence of smaller micro textures [4].

First averages are taken at all points of the image. Neighbourhoods (also known as windows or segments) of varying sizes are centred over each pixel of the image. The average value of a neighbourhood of size $2^k \times 2^k$, centered over point (x,y), is demonstrated in the following algorithm:

$$A_k(x, y) = \frac{1}{2^{2k}} \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i, j)$$

Variation between $2^k \times 2^k$, non-overlapping, neighbourhood averages is then calculated in both horizontal and vertical directions. This process is undertaken by comparing A_k values, situated on either side of each point:

$$E_{k,h}(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|$$

$$E_{k,v}(x, y) = |A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1})|$$

The k value that induces the highest output value from the above formulae, irrespective of the direction, is considered the optimal k value at that particular point. The optimal k value is used to generate S_{best} :

$$S_{best}(x, y) = 2^k$$

Finally, the sum of all S_{best} values is divided by the area of the image in which they exist. The average value of S_{best} is effectively the value representing F_{crs} [3]:

$$F_{crs} = 1/(mn) \sum_x^m \sum_y^n S_{best}(x, y)$$

Tamura et al make mention of the fact that a boundary strip of the image cannot be processed properly. Seeing as each neighbourhood extracts data from an area surrounding a particular pixel, focal points toward the edges of an image will attempt to access regions exceeding the image's dimensions.

2.1.2 Contrast

This feature aims to determine the level of contrast within a given texture. The example is given that when two patterns differ only in grey-level distribution, a measure of differing contrast levels will be observed. However, given images of different structures, more factors should be taken into consideration in order to effectively compute the differences in contrast.

The following four factors are considered to influence a measure of contrast [3]:

- (i) dynamic range of grey-levels,
- (ii) polarisation of the distribution of black and white,
- (iii) sharpness of edges,
- (iv) period of repeating patterns.

Factor i is simply the range of grey-levels within an image. To address factor ii, the standard deviation, σ , about the mean of grey-level distribution was calculated. The measure was considered to represent factor ii to 'some extent' [3].

It was noted that σ was extremely biased to grey level distributions which had a single peak of high intensity at one of either ends of the greyscale range.

To better represent the polarisation of the greyscale distribution, a measure of kurtosis, α_4 , is implemented. α_4 is defined as follows:

$$\alpha_4 = \mu_4 / \sigma^4$$

μ_4 : Fourth moment about the mean

σ^4 : Variance²

In order to combine both factors i and ii, the following measure of contrast was formed:

$$F_{\text{con}} = \sigma / (\alpha_4)^n$$

Through application of the now established F_{con} formula, it was found that a n value of 0.25 yielded the best correlation between psychological and computational results.

Tamura et al leave the formula as it stands, satisfying an approximation for both factors i and ii. It is stated that a measure of contrast has been established in a 'narrow sense' based on the grey-level distribution. Despite the lack of completeness, F_{con} proved successful during testing on the sixteen texture patterns [3].

2.1.3 Directionality

Directionality aims to determine how directional or non-directional a region is. Directionality is a property that does not aim make distinctions between orientations or patterns,

rather a global property which measures the total degree of directionality over the entire region. The feature responds positively to images that seem to resemble parallel lines, rather than an image consisting of randomly scattered points [5].

Tamura et al state that an image may be converted to be viewed in the Fourier domain. Through the generation of a histogram of Fourier power, a measure of directionality may be calculated. However, such features generated in the Fourier domain were found to be unreliable and time consuming, in comparison to those generated in the Spatial domain [3].

A histogram of local edge probabilities is used, in combination with the directional angle of those edges. This method utilizes the fact that a gradient provides both magnitude and direction.

The following 3x3 Prewitt filters are applied to the image in order to calculate both horizontal and vertical differences at each point, the result of which is considered the gradient :

$$\Delta_H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \Delta_V = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

The results from the application of each filter is used to calculate the magnitude, ΔG , and local edge direction, θ , relevant to each point.

$$|\Delta G| = (|\Delta_H| + |\Delta_V|)/2$$

$$\theta = \tan^{-1}(\Delta_V/\Delta_H) + \pi/2$$

The mentioned, ‘histogram of local edge probabilities’ may now be generated by quantizing θ and counting the number of points with $|\Delta G|$ greater than threshold t [3] [4]:

$$H_D(k) = N_\theta(k) / \sum_{i=0}^{n-1} N_\theta(i), \quad k = 0, 1, \dots, n-1$$

where $N_\theta(k)$ is the number of points with conditions $(2k-1)\pi/2n \leq \theta < (2k+1)/2n$ and $|\Delta G| \geq t$. Threshold t serves to withhold any ‘unreliable’ directions, which should not be regarded as edge points. In the conducted experiments, values of $n = 16$ and $t = 12$ are used.

At this point of the computational definition of Directionality, a histogram of local edge probabilities has been established. The histogram represents the occurrences of n angles, distributed over the $[0 \text{ to } \pi]$ range. In order to calculate a quantitative measure of Directionality, Tamura et al propose the degree of sharpness of the peaks within H_D . Given multiple peaks, the sum of second moments around each peak from valley to valley is calculated [3] [6] [7]:

$$F_{\text{dir}} = 1 - r \cdot n_p \cdot \sum_p \sum_{\phi \in w_p}^{n_p} (\phi - \phi_p)^2 \cdot H_D(\phi)$$

- n_p : number of peaks
- ϕ_p : p^{th} peak position of H_D
- W_p : range of the p^{th} peak between valleys
- r : normalising factor related to the quantizing levels of ϕ
- ϕ : quantized direction code

In the experiments carried Tamura et al did not consider any more than two peaks, as to best replicate the psychological test results. It is noted that for general use, the definition should be modified.

2.1.4 Line-likeness

Line-likeness describes the element of a texture that is composed of lines [3]. In other words, the feature responds positively to images that seem to resemble an ensemble of lines [5]. In comparison to Directionality, this feature does not place as emphasis on the consistency of direction over the entire image.

When the direction of two neighbouring edges are nearly equal, the group of edge points are considered a 'line' [3] [7]. Appropriately, a direction co-occurrence matrix, of elements $P_{Dd}(i,j)$, is generated. Each element of $P_{Dd}(i,j)$ is defined as "the relative frequency with which two neighboring cells separated by a distance d along the edge direction occur" [3]. The comparison of cell direction i and cell direction j is taken is performed.

Co-occurrences in the same direction are weighted by +1, and co-occurrences with directions perpendicular to each another are weighted -1. The following equation computes a measure of Line-likeness:

$$F_{\text{lin}} = \sum_i^n \sum_j^n P_{Dd}(i,j) \cos[(i-j)(2\pi/n)] / \sum_i^n \sum_j^n P_{Dd}(i,j)$$

Tamura et al the distance of four, d_4 , as the distance between two points.

2.1.5 Regularity

Regularity is a measure of how regular or irregular a region is. Regularity describes the variations in the placement of texture elements. The regularity of a pattern based image, such as a chess board, may easily be quantified as irregular. However, it is particularly difficult to accurately detect regularity in naturally occurring images without knowing information regarding the size or shape of the elements within an image [3].

Tamura et al makes the assumption that for a given texture, if any feature varies over the whole image, the image is considered irregular. To determine this, partitioned sub-images are taken, and the variation of each feature is calculated between each sub-image [3]. The sum of the variation, for each of the four previously mentioned features, is the measure of regularity.

$$F_{\text{reg}} = 1 - r(\sigma_{\text{crs}} + \sigma_{\text{con}} + \sigma_{\text{dir}} + \sigma_{\text{lin}})$$

r : normalising factor

Note, σ_{xxx} represents the standard deviation of each feature in question, $\sigma(F_{\text{xxx}})$.

2.1.6 Roughness

This feature describes the tactile sense of roughness. According to the results of Tamura's experiments, a combination of coarseness and contrast best aligns with the psychological results [3].

$$F_{\text{rgh}} = F_{\text{crs}} + F_{\text{con}}$$

2.2 Support Vector Machine

A Support Vector Machine (SVM) is a supervised learning model, which aims to develop a function that maps two or more sets of observations with two or more, operator defined categories. The iterative procedure gradually reduces the difference between the predicted and expected result [8].

The SVM is considered 'supervised' as the feature data is labeled to inform the SVM trainer as to where the data has come from, rather than the classes being generated from unidentified data. Traditionally, there exist only two labels to distinguish between two groups of data. Under the circumstance that another group is added, a sole two class SVM cannot distinguish between the three groups.

A multiclass SVM is created by using multiple binary SVMs to classify data consisting of three or more groups. Seeing as the generated Breast Image Feature Data consists of three groups, Normal, Benign and Cancer, multiclass classification methods will be investigated.

2.2.1 Two Class Support Vector Machines

SVMs utilize statistical learning theory, in order to generate a decision function capable of quickly classifying large amounts of data, after a training period. [9] [10]. The aim of the SVM is to determine the location of decision boundaries, in order to produce decision function that creates the largest margin between two data sets, commonly referred to as 'classes'.

The following subsections will briefly go into detail as to how the margin is formed given separable and non-separable data sets.

Hard Margin

Given the circumstance that data set is linearly separable, a hard margin may be utilized to form the distinction between classes within the data set. Let training inputs x_i belong to either class 1 or class 2, where $i = 1$ to n number of inputs. If the input data x is linearly separable, the following decision function may be defined [1]:

$$D(x) = w^T x + b$$

To better understand the outputs of such a function, we define the output of the decision function according to the class that input x_i belongs to. For class 1, $y_i = 1$ and for class 2, $y_i = -1$. Such a relationship is demonstrated as follows:

$$w^T x + b \begin{cases} > 0 & \text{for } y_i = 1 \\ < 0 & \text{for } y_i = -1 \end{cases}$$

However seeing as the training data is linearly separable, there is no data to satisfy the condition $w^T x + b = 0$. Therefore to better control the separability of the data, we consider the following definition of the hyperplane [1]:

$$w^T x + b \begin{cases} \geq 1 & \text{for } y_i = 1 \\ \leq -1 & \text{for } y_i = -1 \end{cases}$$

Kernel Tricks

In a support vector machine, the generalization ability of the SVM is dependent on the optimization of the hyperplane. An optimal hyperplane is determined as to maximize the performance of the support vector machine. However, given non-separable training data, the process to establish a hyperplane capable of effectively distinguishing between classes becomes much more complex [1].

To replace the more conventional method of mapping data about a hyperplane, the concept of a kernel was introduced. A kernel allows the original input space to be mapped to a high-dimensional space, therefore providing a means to separate data more effectively [1].

The following kernel methods are commonly used:

- | | |
|-----------------------------|---|
| (i) Linear | $K(x, x') = x^T x'$ |
| (ii) Polynomial | $K(x, x') = (x^T x' + 1)^d$ |
| (iii) Radial Basis Function | $K(x, x') = \exp(-\gamma \ x - x'\ ^2)$ |

2.2.2 Multiclass Support Vector Machines

Various methods have been developed to solve multiclass problems, through manipulation of support vector machines. Seeing as binary support vector machines employ direct decision functions, the combinational process to a multiclass solution is not simple [1].

Shigeo Abe states that there are roughly four approaches to the multiclass problem:

1. one-against-all,
2. pairwise (one-against-one),
3. error-correcting output code (ECOC),
4. all-at-once.

The following subsections will go into some detail as to how one-against-all and pairwise approaches are designed, and how common issues such as unclassifiable regions are overcome.

One-Against-All

The one-against-all (OAA) approach, also known as ‘one against the rest’ or ‘winner takes all’, was first introduced by Vladimir Vapnik in 1995 [11] [10]. Suppose the data is separated into K classes. A K number of two class SVM classifiers are created to distinguish between one class and the remaining $K-1$ classes [9].

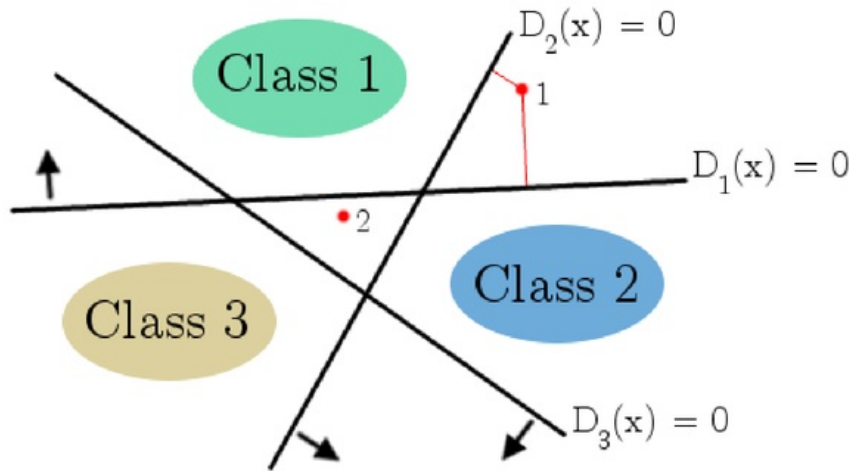


Figure 2.1: Original OAA Design [1]

Given an optimal hyperplane of $D_i(x) = 0$, for an input vector x , if $D_i(x) > 0$, the input vector would be classified as belonging to class i [1]. Initially, Vapnik's formulation of the OAA method classified a data point if a particular class's SVM accepted it and all other classes' SVMs rejected it [10]. For a three class scenario, the input vector would be classified as class 1, according to the following outputs:

$$D_1(x) > 0, D_2(x) < 0, D_3(x) < 0$$

However, if data were to be accepted by two or more classifiers, or none at all, the data would remain unclassified:

$$D_1(x) > 0, D_2(x) > 0, D_3(x) < 0$$

$$D_1(x) < 0, D_2(x) < 0, D_3(x) < 0$$

Points '1' and '2' demonstrate the above two scenarios in figure 2.1, respectfully. Note that the arrows extending off each hyperplane, point toward the positive case for the associated decision hyperplane.

To improve the performance of the OAA SVM, Vapnik proposed in 1998 [12] the use of the continuous decision functions, such that x is classified into the following class [1]:

$$\arg \max_{i=1,\dots,n} D_i(x)$$

The data is classified by determining which margin from each i hyperplanes is the largest [9]. Therefore, previously unclassified point 1 in figure 2.1, is now classified as class 1 ($D_1(x) > D_2(x)$). Point 2 in figure figure 2.1 remains unclassified.

Pairwise

Also known as the 'One against One' SVM, the Pairwise approach involves creating SVM classifiers for all possible pairs of classes. For K classes, $K(K-1)/2$ classifiers are trained. The number of classifiers utilized by this method is greater than the OAA method, given any more than 3 classes. However, the number of training data vectors required per classifier is smaller, 2 vs K , for OAA versus Pairwise respectively [9].

The pairwise approach selects the class according to the maximum number of 'votes' contributed to a particular class by each classifier [10].

Let the decision function that separates class i against class j , be defined as $D_{ij}(x)$, where:

$$D_{ij}(x) = -D_{ji}(x)$$

To demonstrate the regions generated by such decision functions, Figure 2.2 presents the regions for a three class example. Note, the arrows indicate the positive case for each associated decision function $D_{ij}(x)$.

Vector x is classified by voting in the following manner. For each class i , we assess the values generated by decision functions D_{ij} , where $j = 1$ to n , and $j \neq i$. The process is formulated as follows [1]:

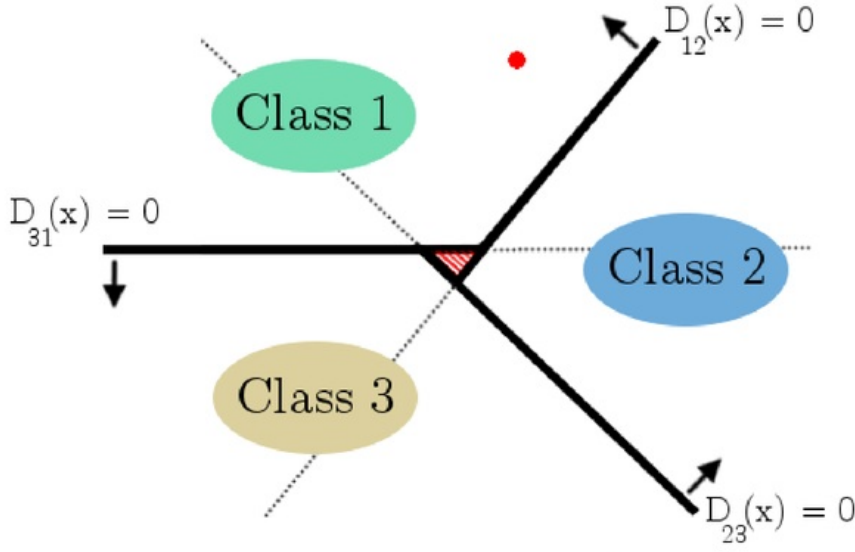


Figure 2.2: Pairwise Design [1]

$$D_i(x) = \sum_{j \neq i, j=1}^n \text{sign}(D_{ij}(x))$$

where

$$\text{sign}(a) = \begin{cases} 1, & \text{for } a \geq 0. \\ -1, & \text{for } a < 0. \end{cases}$$

and we classify x into the class

$$\arg \max_{i=1, \dots, n} D_i(x)$$

To demonstrate, the above process is carried out for the single red point in Figure 2.2:

$$\begin{aligned} \text{sign}(D_{12}(x)) = 1 \quad \& \quad \text{sign}(D_{13}(x)) = 1 \quad \implies \quad D_1(x) = 2 \\ \text{sign}(D_{21}(x)) = -1 \quad \& \quad \text{sign}(D_{23}(x)) = 1 \quad \implies \quad D_2(x) = 0 \\ \text{sign}(D_{31}(x)) = -1 \quad \& \quad \text{sign}(D_{32}(x)) = -1 \quad \implies \quad D_3(x) = -2 \end{aligned}$$

Seeing as $D_1(x) = n - 1$ and $D_k < n - 1$ for $k \neq 1$, x is classified into class 1. However, if any of the $D_i(x)$ is not equal to $n - 1$, x may be classified into more than one class. Such an occurrence is represented by the red-lined region at the centre of figure 2.2, where $D_1(x) = D_2(x) = D_3(x) = 0$ [1].

2.2.3 Unclassifiable Regions

When a particular observation falls into a unclassifiable region, a commonly observed response is to classify the observation at random. Two, more suitable countermeasures are explored in this section: fuzzy and decision-tree based solutions.

Fuzzy Support Vector Machines

Shigeo Abe introduces the idea of a membership function as to resolve the issue of unclassifiable regions. It is emphasized that the results from conventional pairwise classification is retained for the classifiable regions.

For optimal hyperplane used in conventional pairwise classification, $D_{ij}(x)$ for $i \neq j$, membership functions $m_{ij}(x)$ formed in the direction orthogonal to $D_{ij}(x) = 0$ [1]:

$$m_{ij}(x) = \begin{cases} 1 & \text{for } D_{ij}(x) \geq 1, \\ D_{ij}(x) & \text{otherwise} \end{cases}$$

Class i membership of x is defined by either a minimum or average operation, respectively:

$$m_i(x) = \min_{j \neq i, j=1, \dots, n} m_{ij}(x)$$

$$m_i(x) = \frac{1}{n-1} \sum_{j \neq i, j=1}^n m_{ij}(x)$$

Vector x is classified into the following class based on either of the above methods [1]:

$$\arg \max_{i=1, \dots, n} m_i(x)$$

The distribution of the once ‘unclassifiable’ region is demonstrated in figure 2.3.

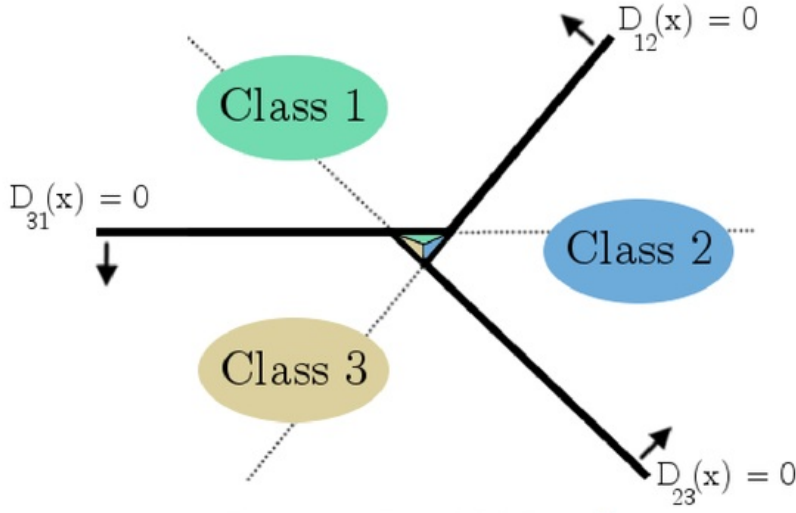


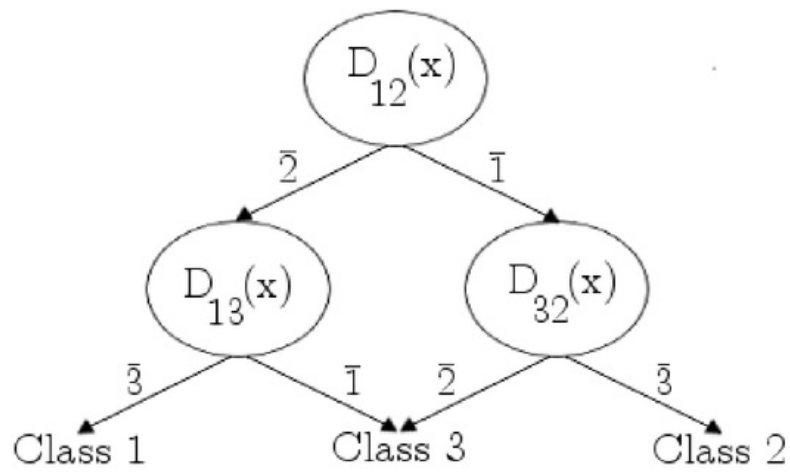
Figure 2.3: Fuzzy SVM design [1]

Decision Tree based Support Vector Machines

As demonstrated in figure 2.4, a decision tree based support vector machine utilizes a hierarchical approach. Also known as the decision directed acyclic graph (DDAG) support vector machine, the DDAG uses a process of ruling out classes as the decision tree is transversed.

At the top-level of the DDAG, the decision function D_{ij} for a chosen class pair ij , serves to rule out the possibility that an observation belongs to either class i or class j . For $D_{ij} \geq 0$, class j is no longer considered and for $D_{ij} < 0$, class i is no longer considered [1].

For the given configuration in figure 2.4, the top-level classifier D_{12} rules out the possibility of an observation belonging to either class 1 or class 2. Accordingly, a decision function capable of distinguishing between the remaining two classes is utilized as the final stage of classification.

**Figure 2.4:** Decision Tree [1]

Chapter 3

Mammographic Images

The images used in this project were provided by Macquarie University, sourced from Aachen University. The entire image set contains 9548 images, distributed as follows:

Image Set	Resolution Range	Number of Images
Normal	[1112 x 2988] - [5311 x 6781]	2716
Benign	[1561 x 3256] - [5311 x 6871]	3291
Cancer	[1561 x 3376] - [5641 x 6886]	3541

Table 3.1: Mammographic Images: Dimension and Quantity

The images are of varying dimensions. The range of minimum and maximum resolutions in table 3.1 are determined by the total surface area of the images.

There exist computer aided detection techniques that involve the utilize meta-data associated with the image set. Such meta-data may include, the class and severity of the abnormality, as well as the co-ordinates of the abnormality within the image [13]. The supplied image sets were categorized by the folders in which they were contained. The three categories were Normal, Benign and Cancer — no further meta-data was supplied. For this reason, the mammographic images are to analysed in their entirety, rather than investigating specific regions or sub-images. The following subsection will go into some detail as to what preprocessing steps were undertaken on the mammographic image set.

3.0.1 Pre-processing

Seeing as Tamura’s features use the entire matrix of greyscale values than comprise an image, it is vital that areas of the image that are not relevant to the breast are removed.

The image set was manually cropped as to remove aspects of the image that may interfere with the analysis of said image. Such features include film-labels, white areas caused by prior modification, and white strips on image edges. Figure 3.1 demonstrates the process undertaken on a sample image.

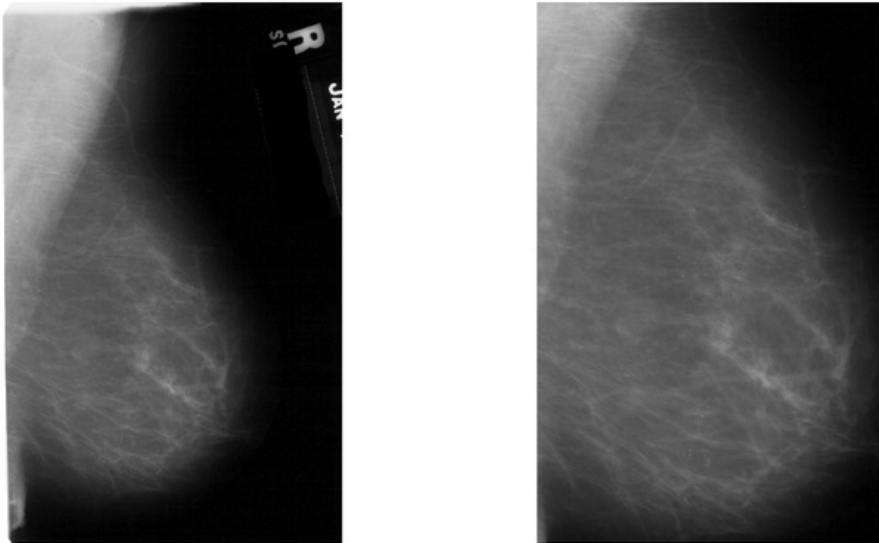


Figure 3.1: Image Crop, Before and After

As mentioned, this process was performed manually as the size, orientation and arrangement of the images varied throughout the entire image set. Additionally any image deemed corrupt, and therefore unusable, was removed from the image set. It should be noted that the outlined preprocessing method not only serves to provide more accurate image analysis, the computational time required to analyse said image is reduced.

As demonstrated in table 3.2, the preprocessing method generates an image set of reduced resolution and quantity.

Image Set	Resolution Range	Number of Images
Normal	[876 x 1446] - [4206 x 6341]	2716
Benign	[778 x 2674] - [4554 x 6312]	3291
Cancer	[1297 x 1014] - [4871 x 5918]	3520

Table 3.2: Cropped Mammographic Images: Dimension and Quantity

Chapter 4

Tamura's Features

Tamura's "Computational Definitions of Basic Textural Features" are to be implemented in MATLAB to produce feature data for each image, be it Normal, Benign or Cancer [3].

Although the previous chapter addresses the fact that cropping the images effectively reduces computation time, the dimension and quantity of images within the preprocessed mammographic image set remains considerably large.

The following implementations of Tamura's features aim to best represent models outlined in section 2.1, whilst keeping computation time to a minimum. As to reduce computation times, operations involving the traversal of all pixels in an image are avoided. Common countermeasures such as masks, filters and matrix manipulations are employed.

The following sections will explore the practical approaches to each of the six features. Although the each feature may be implemented in MATLAB as a compact representation, the following pseudocode representations serve to demonstrate Tamura's computational models from a more generalized perspective.

4.1 Coarseness

The average of greyscale values within neighbourhoods of size $2^k \times 2^k$ at all points, (x,y) , are calculated. The MATLAB implementation involves the use of a 'average' filter.

The filter calculates the mean over a square window of a specified dimension, and stores the generated value in the 'centre' pixel of the window, for all pixels of the image.

Note, in Tamura's original equation, the dimension of the neighbourhood is specified by a pixel range in horizontal and vertical directions. For $k = 0$, of which is intended to generate a 1×1 neighbourhood, the given equation is invalid. The average value for a 1×1 neighbourhood is simply the pixel value itself. The 'average' filter takes this into account and assigns the entire image to the neighbourhood average values for $k = 0$.

Algorithm 1 Neighbourhood Averages

```

kMax = user defined
for k = 0 to kMax do
  av(:, :, k) = filter(image, 'average', [2k, 2k])
end for

```

Next, the difference in the neighbourhood averages of non-overlapping $2^k \times 2^k$ neighbourhoods is calculated horizontally and vertically, about each pixel. The k value associated with highest difference, be it horizontally or vertically, is recorded for each pixel.

It should be noted that in order compare neighbourhoods about a pixel, the following process may be undertaken.

- Duplicate the image
- Rotate both images in opposite directions by $2^{(k-1)}$ pixels
- Subtract the two matrices from one another

Although Figure 4.1 demonstrates this method for only one point, all points are calculated concurrently through this method, drastically reducing computation time. The same method can be applied vertically.

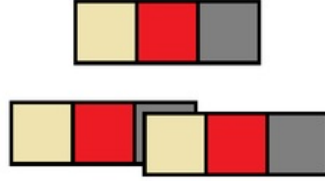


Figure 4.1: Comparing Neighbours with Circular Shift

The following algorithm represents the comparative process. Note, for $k=0$, the distance $2^{(k-1)}$ is rounded up.

Algorithm 2 Neighbourhood Comparison

```

for k = 0 to kMax do
  x1 = circular shift av(:, :, k) by 2(k-1) in the +X direction
  x2 = circular Shift av(:, :, k) by 2(k-1) in the -X direction
  horizontal(:, :, k) = |x1 - x2|

  y1 = circular shift av(:, :, k) by 2(k-1) in the +Y direction
  y2 = circular shift av(:, :, k) by 2(k-1) in the -Y direction
  vertical(:, :, k) = |y1 - y2|
end for

```

The maximum difference in neighbourhood averages is determined for each k value, regardless of direction. The k index corresponding to the maximum difference in neighbourhood averages for all k values is then calculated, represented by $kOpt$. Strips in which neighbourhood comparisons may not accurately be calculated are removed and the optimal k -value at each pixel is averaged over the newly defined region.

At this point a measure of coarseness is achieved however to normalize the result into a range of $[0 \ 1]$, the value is divided by the maximum F_{con} value of 2^{kMax} .

Algorithm 3

```

for  $k = 0$  to  $kMax$  do
     $e(:, :, k) = \max(\text{horizontal}(:, :, k), \text{vertical}(:, :, k))$ 
end for

 $kOpt(:, :) = k \text{ index corresponding to } \max(e(:, :, k))$ 

 $kOpt = kOpt((2^{kMax-1}+1):(end-2^{kMax-1}), (2^{kMax-1}+1):(end-2^{kMax-1}))$ 

 $F_{con} = (\text{sum}(kOpt)/(\text{height} \times \text{width}))/(2^{kMax})$ 

```

In Tamura's experiments, $kMax = 5$ enabled the computational model to best correlate with the experimental results. The effects of $kMax$ are to be investigated.

4.2 Contrast

In order to calculate the Contrast, the number of occurrences of each unique greyscale level must be recorded. This can be achieved through the generation of a histogram, or manually, by recording of unique greyscale value occurrences. Figure 4.2 demonstrates how a histogram may be used to represent the intensity of each unique greyscale level. The formulation for Contrast consists of a combination of statistical techniques, of which are applied to the mentioned greyscale intensity levels.

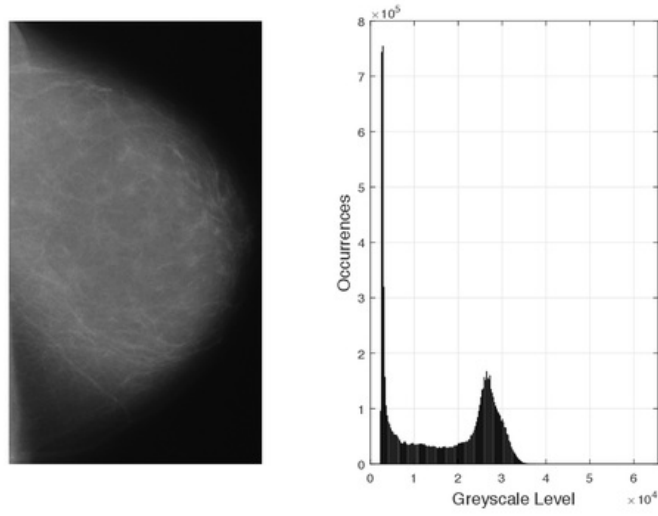


Figure 4.2: Greyscale Intensity Levels for Breast Image

Algorithm 4 Contrast

L = unique greyscale levels of image

I = number of occurrences of each greyscale level in L

$$\mu = \text{sum}(L \times I) / \text{sum}(I)$$

$$\sigma_2 = \text{sum}((L - \mu)^2 \times I) / \text{sum}(I)$$

$$\sigma_4 = \sigma_2^2$$

$$\mu_4 = \text{sum}((L - \mu)^4 \times I) / \text{sum}(I)$$

$$\alpha_4 = \mu_4 / \sigma_4$$

$$\sigma = \sqrt{\sigma_2}$$

$$n = 0.25$$

$$F_{con} = \sigma / (\alpha_4^n)$$

In Tamura's experiments, $n = 0.25$ enabled the computational model to best correlate with the results from the psychological experiments. This model of F_{con} will remain as is, that is, no further investigation into n .

4.3 Directionality

The following pseudocode representation of Directionality is a modification of Francesco Bianconi and Fernando Domnguez's work. The model used in their paper journal article 'Discrimination between tumour epithelium and stroma via perception-based features' [5] [14].

First step, Sobel masks are applied to generate the gradients about each pixel. The angle of the gradient, θVal , is then calculated.

Algorithm 5 Gradient via Sobel mask

```

h = [-1 0 1; -2 0 2; -1 0 1]
v = [ 1 2 1; 0 0 0; -1 -2 -1]
gradientH = filter(image,h)
gradientV = filter(image,v)
 $\thetaVal = \tan^{-1}(\text{gradientV}/\text{gradientH})$ 

```

Next, the values of θVal , are distributed according to the number of bins defined by nBins. The range $[-\pi/2 \pi/2]$, is split into smaller sub-ranges as to group angles of similar values. The effect of the number of nBins, as inputted by the user, is to be investigated.

A histogram is build based off θVal distributed over nBins. In MATLAB the histogram generation function 'histc' treats the uppermost bin as unique count [15]. In this case, all occurrences of $\pi/2$ are counted separately. A 'trick' is employed in which the uppermost bin is increased, to allow occurrences of $\pi/2$ to fall into the previous bin. The uppermost bin is removed as there exists no θVal greater than $\pi/2$.

As demonstrated in histogram within Figure 4.3, the final bin of ~ 1.44 is inclusive of values up to, and including, $\pi/2$.

Algorithm 6 Histogram of local edge probabilities

```

nBins = user defined
n = width of  $\thetaVal$ 
m = height  $\thetaVal$ 
weight =  $\pi/nBins$ 
edges =  $-\pi/2 + (0 \text{ to } (nBins - 1)) * \text{weight}$ 
binRanges = [edges,  $1.1(\pi/2)$ ]
 $\thetaVector = \thetaVal$  reshaped as a vector of length (n x m)
nanCount = 0
for values in  $\thetaVector$  do
  if  $\thetaVector == \text{NaN}$  then
    nanCount++
  end if
end for
pHist = histogram( $\thetaVector$ , binRanges)

```

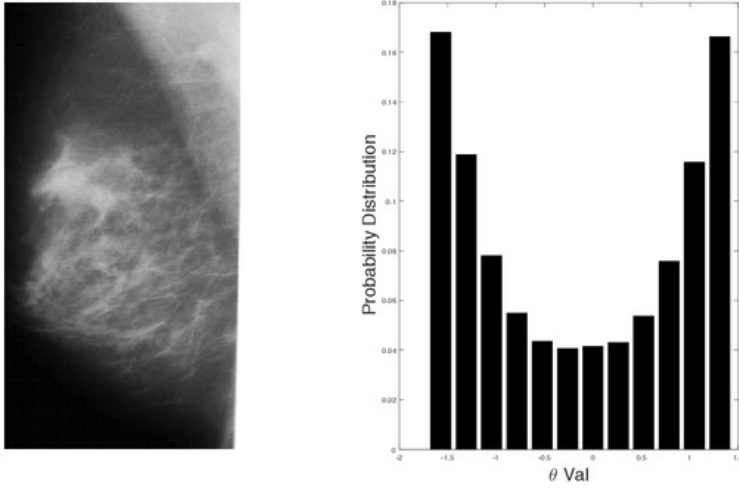


Figure 4.3: Histogram pHist for Sample Image

A uniform distribution is generated based on the number of bins defined. The euclidean distance is measured from the uniform distribution to histogram pHist. The measured distance is a representative of the level of Directionality, F_{dir} .

As demonstrated in Figure 4.3, the histogram pHist is representative of the gradient angles within the image. The peaks toward $\pm 1.5\pi$ inform us that occurrences of that particular angle are more likely to occur within the image. Such peaks are indicative of a highly directional image.

Algorithm 7 peaks

```
uniform = (1/nBins) x array of 1's length nBins
distance = (sum((pHist - uniform)2)0.5
 $F_{dir}$  = distance
```

4.4 Line-likeness

The following pseudocode representation of Line-likeness is a modification of Francesco Bianconi and Fernando Domnguez's work. The model used in their paper journal article 'Discrimination between tumour epithelium and stroma via perception-based features' [5] [14].

Similar to directionality, a sobel mask is applied to the image as to generate gradients about each pixel. The four quadrant inverse tangent (atan2) is used to calculate the angle,

θVal , at each pixel, of range $[-\pi \pi]$.

Algorithm 8 Gradient via Sobel mask

```

h = [-1 0 1; -2 0 2; -1 0 1]
v = [ 1 2 1; 0 0 0; -1 -2 -1]
gradientH = filter(image,h)
gradientV = filter(image,v)
 $\theta\text{Val} = \text{atan2}(\text{gradientV}/\text{gradientH})$ 

```

The angle values, θVal , are to be quantized over the range of $n\text{Bins}$. To do so, the range of θVal is adjusted from $[-\pi \pi]$ to $[0 1]$, and then multiplied by the number of $n\text{Bins}-1$. This process aims to group gradient angles of similar values over an $n\text{Bin}$ number of groups.

As recommended by Tamura et al, the number of $n\text{Bins}$ is to be set according to the number of $n\text{Bins}$ used in the computation of Directionality [3].

Displacements D is defined according to distance d . A value of $d = 4$ will be used as recommended by Tamura et al [3].

Algorithm 9 Adjust θVal and define Displacements

```

nBins = user defined
d = 4
 $\theta\text{Val} = \theta\text{Val}$  range adjusted from  $[-\pi \pi]$  to  $[0 1]$ 
 $\theta\text{Val} = \theta\text{Val} \times (n\text{Bins}-1)$ 
 $D = [0 \ d; -d \ d; -d \ 0; -d \ -d]$ 

```

The newly established displacements and gradient angles are to be inputted to construct a co-occurrence matrix. The co-occurrence matrix represents the distribution of co-occurring θVal 's at a particular offsets, as defined by D .

The first column of D refers to the vertical direction, and the second column refers to the horizontal direction. In combination, a displacement is defined. For example, $[-d \ d]$ refers to d pixels to the left, and d pixels upward.

A co-occurrence matrix is generated for each row in D , that is, a co-occurrence matrix for each displacement. Multiple displacements are defined as to reduce the effects of potentially biased results caused by image rotation.

Algorithm 10 Weighted Co-occurrence matrices

```

cM = cooccurrenceMatrices( $\theta$ Val, nBins, D)

weight = zeros(nBins);
for i = 0:nBins do
    for j = 0:nBins do
        weight(i,j) = abs(cos((i - j)x2 $\pi$ /nBins));
    end for
end for

for d = 1:4 do
    cM(:, :, d) = cM(:, :, d) x weight
    Fd(d) = sum(cM(:, :, d))
end for

Flin = mean(Fd)

```

The weights are defined such that when neighbouring gradient angles, θ Val, are in the same direction, the weight is equal to 1. When co-occurrences are perpendicular, weight is equal to 0.

The weights were applied to each co-occurrence matrix. The average result of the four weighted co-occurrence matrices is averaged, the result of which is the measure of Line-likeness, F_{lin} .

4.5 Regularity

In Tamura et al paper, the texture samples used to conduct all experiments are 256 by 256 pixel images. Appropriately, they divide the image into quarters in order to detect changes in Coarseness, Contrast, Directionality and Line-likeness over each newly created sub-image.

As established in section 3, the provided mammographic image set contains predominantly portrait orientated images of varying resolution. The below implementation has been designed such that the number of horizontal and vertical slices applied to the image may be modified.

Algorithm 11 F_{reg} : Measure of Regularity over (n x m) sub-images

```

n = user defined
m = user defined
w = (width)/n
h = (height)/m
count = 0

for i = 1 to n do
  xMin = w x (i-1)
  xMax = h x i
  for j = 1 to m do
    yMin = w x (j-1)
    yMax = h x i

    subImage = image(xMin:xMax , yMin:yMax)
     $F_{crs}(count) = \text{Coarseness}(subImage)$ 
     $F_{con}(count) = \text{Contrast}(subImage)$ 
     $F_{dir}(count) = \text{Directionality}(subImage)$ 
     $F_{lin}(count) = \text{Linelikeness}(subImage)$ 
    count++
  end for
end for
 $F_{reg} = 1 - (\sigma(F_{crs}(:)) + \sigma(F_{con}(:)) + \sigma(F_{dir}(:)) + \sigma(F_{lin}(:)))/(n \times m)$ 

```

To satisfy the predictive nature of this project, the average aspect ratio of the entire image set was calculated by taking the average width/height of all images. The average aspect ratio was measured to be 1 : 1.743. Based upon this result two more appropriate sub-image distributions were proposed:

$$2 : 3.486 \implies 2 : 3 \qquad 3 : 5.229 \implies 3 : 5$$

The intention is to calculate a measure of regularity based on Tamura's original 2 by 2 distribution, and then assess the performance in comparison to the more appropriate subdivisions, 2 by 3 and 3 by 5.

4.6 Roughness

Roughness is simply the sum of the Coarseness and Contrast value for each image.

Algorithm 12 F_{rgh} : Tamura's Definition

$$F_{rgh} = F_{crs} + F_{con}$$

As per Tamura's definition, very little computation is required. The model merely adds the result of F_{crs} and F_{con} , both of which have already been computed.

However according to Tamura's experimental results, the performance of Roughness was below all other features. Roughness had poor correspondence with the psychological results and seeing as it is formed by adding F_{crs} and F_{con} , high correlation to those features may be expected.

Bianconia et al propose a new measure of roughness. Their approach implements a definition of surface roughness based on the amplitude parameter ' S_q ' from ISO standard 25178-2 [5] [16]. ' S_q ' is defined as the 'Root-mean square height of the scale-limited surface' (RMS). RMS is more commonly known as standard deviation when the RMS calculation is performed from a baseline value [17], in this case we will use the mean of the greyscale value of the image.

Algorithm 13 F_{rgh} : ISO 25178 Definition

$$F_{\text{rgh}} = \sigma(\text{image}(:))$$

The performance of both models of F_{rgh} are to be tested.

4.7 Chapter Summary

Till this point, each of Tamura's features have been established as practical models. For particular features, the established model is absolute, that is, no further modification is to be applied. However, for a majority of the features, the performance benefits or drawbacks associated with user definable inputs is to be investigated.

To provide a good reference point for later stages in the report, the user definable inputs which are to be investigated are collated here:

- (i) Coarseness: kMax
- (ii) Directionality: number of nBins
- (iii) Regularity: sub-image distribution
- (iv) Roughness: Tamura versus ISO model

The means of measuring the performance of each feature variation is addressed in the following chapter.

Chapter 5

Feature Data

For each directory of cropped images, Normal, Benign and Cancer, the six feature extraction algorithms described in Chapter 4 were applied. The generated feature data was initially stored in separate arrays, according to its source.

5.1 Feature Data Representation

In order to distinguish between sources, given circumstances of either combining or splitting data, the identifier column is used to keep track of the data's source. The following table 5.1 demonstrates the structure of the feature data.

	F_{crs}	F_{con}	F_{dir}	F_{lin}	F_{reg}	F_{rgh}	Identifier
Image 1							
Image 2							
Image 3							
...							

Table 5.1: Feature Data Arrangement

5.2 Feature Analysis

In section 4.7, the intention to explore the effects of particular user-definable inputs, in regards to Tamura's features, is established. The purpose of doing so is to choose the user inputs that correspond to the most effective variant of Tamura's Features.

The aim of this process is to maximize the separability of the feature data produced by each feature extraction model, as to improve the generalization ability of the two class

support vector machines. To reiterate, feature data that establishes a clear distinction between classes will best serve the classification process.

To assess each variant of the feature extraction models, Fishers Discrimination Ratio (FDR) was used as to provide a measure of separability. FDR is applied to data originating from each class pair — normal & benign, cancer & normal, benign & cancer — for each of the feature extraction models.

$$\text{FDR}_{ij} = \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}$$

To visualize what a particular measure of FDR represents, the probability density functions (PDF) of two sets of sample data are plotted in figure 5.1. The FDR measure associated with each pair of PDF curves is included as to demonstrate the responsiveness of FDR to the separability of the data.

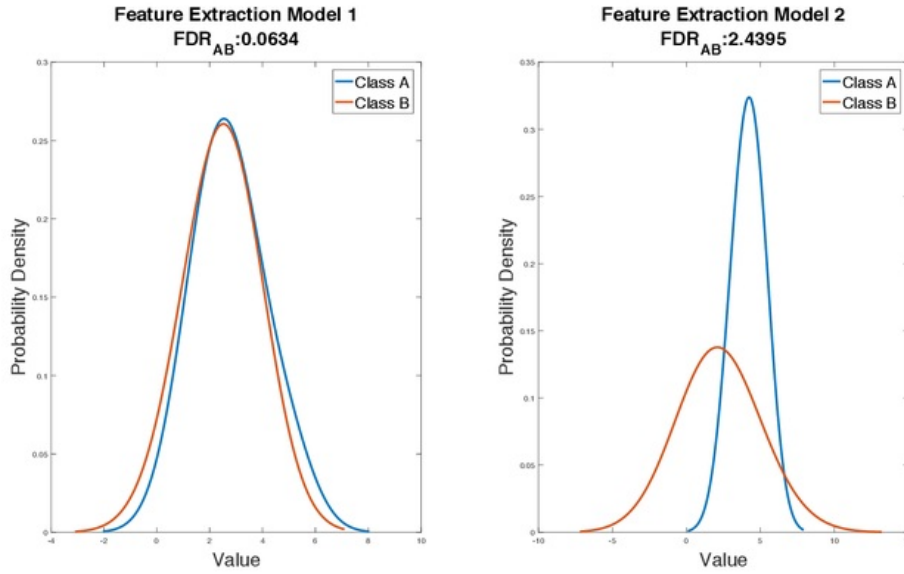


Figure 5.1: Sample PDF Curves with FDR Measure

As per the definition, FDR may only perform a measure between two classes. Each combination of class pairs is assessed, of which is appropriate as each data pair is inputted into the two class support vector machines. However to form a generalized measure of the amount of separability that a particular feature provides, the sum of the FDR measures over all class pairs is utilized [18]:

$$\text{FDR}_{\Sigma} = \left(\sum_{i=1}^K \sum_{j \neq i}^K \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \right) / 2$$

5.3 Feature Data Distribution

At the conclusion of the feature optimization stage, an optimal set of features has been defined, and the data generated accordingly. In order to prepare the feature data for classification, the data must be distributed in a particular manner.

The data must be prepared in order to both train the support vector machines, and test the performance of the newly trained support vector machines. The following subsections will go into some detail as to how the data is distributed as to the predictive nature of this project.

5.3.1 Train, Test Distribution

The generated feature data is delegated with 70:30 distribution into training and testing groups, respectively.

The following is the process undertaken:

- (i) feature data from each source labeled
- (ii) all feature data is concatenated, and randomized
- (iii) 70% is allocated to training data
- (iv) 30% is allocated to test data

Such a process is demonstrated in the pseudocode below, as to demonstrate the manipulation of the feature data arrays and their identifying labels.

Algorithm 14 Feature Distribution: Train and Test Data

```

normalData(:, 7) = 0
benignData(:, 7) = 1
cancerData(:, 7) = 2
trainPercentage = 0.7

allData = [normalData; benignData; cancerData]
allLength = number of rows in allData
cutoff = round(trainPercentage x allLength)
allData = row-wise randomisation(allData)

trainData = allData(1:cutoff,:)
testData = allData((cutoff + 1):allLength,:)

```

5.3.2 Training Pairs

The now established training set consists of data originating from Normal, Benign and Cancer image sources, with identifying labels 0, 1 and 2 respectively. However a two class

support vector machine may only be trained using data that has been categorized into two classes.

The following is the process undertaken:

- (i) normal, benign and cancer data is extracted from the training set
- (ii) redistributed into training pairs \implies NB, CN & BC
- (iii) all training pairs randomized

Note, this particular data distribution will only be applicable in later stages for a pairwise multiclass support vector machine.

Algorithm 15 Training Pairs Distribution

```

for i = 1 to length of trainData do
  if trainData(i,7) == 0 then
    normalTrain(i,:) = trainData(i,:)
  end if

  if trainData(i,7) == 1 then
    benignTrain(i,:) = trainData(i,:)
  end if

  if trainData(i,7) == 2 then
    cancerTrain(i,:) = trainData(i,:)
  end if
end for
normalTrain(:,7) = 0
benignTrain(:,7) = 1
NB = row-wise randomisation([normalTrain; benignTrain])
cancerTrain(:,7) = 0
normalTrain(:,7) = 1
CN = row-wise randomisation([cancerTrain; normalTrain])
benignTrain(:,7) = 0
cancerTrain(:,7) = 1
BC = row-wise randomisation([benignTrain; cancerTrain])
  
```

5.3.3 Test Set

Up until now the identifying label has been left alongside the data. For testing, the data must remain anonymous as to satisfy the predictive nature of the testing stage. The identifier labels are used post-classification to measure the performance of the classification process.

Algorithm 16 Test Data

```
testInput = testData(:,1:6)
testTarget = testData(:,7)
```

As demonstrated, the column of ‘testData’ containing the identifying labels is removed and stored for post classification. Note, no further randomization is applied to either of the newly generated arrays as each target is specific to a particular input. The row in which the input and target exist, is the only point of reference for post-classification performance measurements.

Chapter 6

Support Vector Machines

6.1 Two Class Support Vector Machine

As it stands, the standard binary SVM is quite straightforward through utilizing the ‘Statistics and Machine Learning Toolbox’ [19]. The two class support vector machines will utilize the ‘training pairs’ as allocated in section 5.3.2. It is important that the two class classifiers are configured such that optimal performance is achieved for the given data.

As discussed in section 2.2.1, there exist various methods to map data in a high-dimensional space. The following kernel functions will be tested, as provided in the MATLAB toolbox mentioned above:

- | | |
|------------------|--|
| (i) rbf | $K(x_1, x_2) = \exp(-\ x_1 - x_2\ ^2)$ |
| (ii) linear | $K(x_1, x_2) = x_1'x_2$ |
| (iii) polynomial | $K(x_1, x_2) = (1 + x_1'x_2)^p$ |

In order to assess the performance of each kernel method, k-fold cross validation is applied. Such a method serves to predict the performance of the two class support vector machine, based purely on the training data.

This method of cross validation involves generating a ‘k’ number of folds within the training data. Each fold distinguishes between a newly formed ‘train’ and ‘test’ set within the original training data. An average over the k-folds is taken reduce the variability of results generated by each fold.

To better understand the k-fold performance results, confusions matrices are generated. The summation of the positive and negative classification rates over the k-folds is utilized to generate a confusion matrix per two class SVM. Through analysis of the positive and negative rates, the specific source of the classified or misclassified data becomes apparent.

		Predicted Class		Positive Rate (%)	Negative Rate (%)
		T	F		
Actual Class	T	TP	FN	TPR	FNR
	F	FP	TN	FPR	TNR

Table 6.1: Confusion Matrix

6.2 Multiclass Support Vector Machine

This section will look at how the results from the two class support vector machines are compiled as to generate a multiclass solution. To demonstrate the co-operative process, the voting schemes of each of the following **pairwise** solutions are to be demonstrated:

- (i) standard,
- (ii) fuzzy: minimum and average,
- (iii) decision directed acyclic graph.

The two class classification results will be generated using the following MATLAB function:

$$[\text{label}, \text{score}, \text{cost}] = \text{predict}(\text{model}, \text{x}),$$

where ‘model’ represents a trained two class support vector machine, and ‘x’ is the label-less test data that has been reserved till this point [20]. For the purposes of generating a multiclass solution, the first two outputs are to be used.

‘Label’ returns a binary output to indicate the absolute classification of a particular observation, that is, either a zero or a one. However, in order to address unclassifiable regions, a more specific measure of confidence is required. ‘Score’ returns an output indicating the likelihood that a observation belongs to a specific class. The measure of likelihood is the posterior probability that a particular observation in ‘x’ belongs to a class.

6.2.1 Standard Pairwise Support Vector Machine

As mentioned, this solution utilizes the binary output, ‘label’. For each class pair the output of ‘label’ is assigned to $\text{Outputs}_{ij}(\text{x})$ where ‘ij’ refers to the class pair used and ‘x’ is a particular observation.

The following algorithm demonstrates how votes are generated as to classify each observation based on the idea of a majority vote. Each row in ‘votes’ is representative of a particular observation, and each column represents the votes per class for mentioned

observation. In the case of a tied vote, the observation is deemed unclassifiable. Under such condition, an additional vote is randomly cast as to satisfy a majority vote.

Algorithm 17 Standard Pairwise Vote

```

for i = 1 to length of testData do
  if outputs12(i) == 0 then
    votes(i,1)++
  else
    votes(i,2)++
  end if
  if outputs31(i) == 0 then
    votes(i,3)++
  else
    votes(i,1)++
  end if
  if outputs23(i) == 0 then
    votes(i,2)++
  else
    votes(i,3)++
  end if
end for

```

6.2.2 Fuzzy Support Vector Machine

The following voting scheme is representative of the membership defined in section 2.2.3. This solution utilizes the posterior probability output, 'score'. For each class pair the output of 'score' is assigned to $pp_{ij}(x)$, where 'ij' refers to the class pair used and 'x' is a particular observation.

In accordance to the membership defined in section 2.2.3, placeholder votes are generated for each class pair combination in the following manner:

$$v_{ij}(x) = \begin{cases} 1 & \text{for } pp_{ij}(x) \geq 1, \\ pp_{ij}(x) & \text{otherwise} \end{cases}$$

Such a membership is defined as to retain the classification performance of the standard pairwise solution, whilst providing information relevant to the unclassifiable region. The utilization of such membership is demonstrated for all classes in the following pseudocode.

Algorithm 18 Fuzzy: Membership Based Voting

```

for i = 1 to length of testData do
  if pp12(i) ≥ 1 then
    v12 = 1;
  else
    v12 = pp12(i)
  end if
  if pp13(i) ≥ 1 then
    v13 = 1
  else
    v13 = pp13(i)
  end if
  if pp21(i) ≥ 1 then
    v21 = 1
  else
    v21 = pp21(i)
  end if
  if pp23(i) ≥ 1 then
    v23 = 1
  else
    v23 = pp23(i)
  end if
  if pp31(i) ≥ 1 then
    v31 = 1
  else
    v31 = pp31(i)
  end if
  if pp32(i) ≥ 1 then
    v32 = 1;
  else
    v32 = pp32(i)
  end if
end for

```

The placeholder votes for each class pair ‘ij’ are then combined such that a distinct vote for each class ‘i’ is formed. One of the following two methods may be applied.

Minimum

$$v_i(x) = \min_{j \neq i, j=1, \dots, n} v_{ij}(x)$$

Algorithm 19 Fuzzy: Minimum

```

for i = 1 to length of testData do
  votes(i,1) = min(v12, v13)
  votes(i,2) = min(v21, v23)
  votes(i,3) = min(v31, v32)
end for

```

Average

$$v_i(x) = \frac{1}{n-1} \sum_{j \neq i, j=1}^n v_{ij}(x)$$

Algorithm 20 Fuzzy: Average

```

for i = 1 to length of testData do
  votes(i,1) = (v12 + v13)/2
  votes(i,2) = (v21 + v23)/2
  votes(i,3) = (v31 + v32)/2
end for

```

6.2.3 Decision Tree Based Support Vector Machine

The following decision directed acyclic graph (DDAG) support vector machine is representative of figure 2.4 in section 2.2.3. This particular configuration is one of three possible decision tree configurations, where a configuration is distinguished by the top-level decision function. All variants are to be tested.

Algorithm 21 DDAG: Top-level D₁₂

```

for i = 1 to length of testData do
  if outputs12(i) == 0 then
    if outputs31(i) == 0 then
      votes(i,1)++
    else
      votes(i,3)++
    end if
  else
    if outputs23(i) == 0 then
      votes(i,3)++
    else
      votes(i,2)++
    end if
  end if
end for

```

Chapter 7

Results & Discussion

The experimental procedures have been defined up until this point. The practical implementations of Tamura's Features and Support Vector Machines have been explored in detail. This chapter will cover three main sections,

- (i) feature optimization,
- (ii) two class support vector performance,
- (iii) and multiclass support vector machine performance.

At each stage results are discussed to better understand what each result represents, and why such a result may have occurred. Note, 'optimization' is not referred to in its practical sense, rather, the experimental setup associated with optimal results is chosen.

7.1 Feature Optimization

As outlined in section 4.7 and section 5.2, both the intention and means to assess variants of particular feature extraction models is established. The following results demonstrate the selection process undertaken through consideration of the separability a feature provides and the time taken to extract said feature.

7.1.1 Coarseness

As presented in the table 7.1, the variation in the range of k values is investigated. Such a modification alters the range of neighbourhood sizes implemented in the measure of Coarseness. An initial implementation of Coarseness did not test for $k = 0$, the result of which is included as to emphasis the usefulness of smaller neighbourhood sizes, as represented by the k range of 1 to 5.

As demonstrated, the feature data generated by k values of 0 to 4 provided the most separability between classes whilst taking the least time to compute. Neighbourhood sizes larger than $2^4 \times 2^4$ seem to alter the measure of Coarseness as to generate results that were

k	FDR				Extraction Time (s/image)
	NB	CN	BC	Σ	
1 to 5	0.0488	0.0041	0.0236	0.0765	12.92
0 to 4	0.2108	0.0616	0.0493	0.3217	12.88
0 to 5	0.1275	0.0364	0.0368	0.2007	15.74
0 to 6	0.0228	0.0000	0.0208	0.0436	18.54

Table 7.1: Fischer's Discriminant Results for Class Pairs: Coarseness

more uniform across all classes. This effect may be attributed to a bias of optimal k values, generated by neighbourhoods larger than $2^4 \times 2^4$, for all classes.

Due to the large amount of time required in comparison to other features (see table 7.5), no further investigation into Coarseness was undertaken.

7.1.2 Contrast

In Tamura's experiments, $n = 0.25$ enabled the computational model to best correlate with the results from the psychological experiments. This model of Contrast was left unmodified, that is, no further investigation into n . For this reason, the measure of Contrast in table 7.5 was generated with an n input of 0.25.

7.1.3 Directionality

As presented in table 7.2, the variation in the value of $nBin$ is investigated. The modifier, $nBin$, affects the number of bins used to construct the histogram of gradient angles.

nBins	FDR				Extraction Time (s/image)
	NB	CN	BC	Σ	
10	0.2972	0.1477	0.0152	0.4601	1.46
12	0.3098	0.1496	0.0165	0.4759	1.49
14	0.2972	0.1464	0.0148	0.4584	1.51
16	0.3106	0.1489	0.0161	0.4756	1.51

Table 7.2: Fischer's Discriminant Results for Class Pairs: Directionality

Although a $nBin$ value of 16 provided the most separability between Normal and

Benign images, an nBin value of 12 was chosen as it provided the most separability over all classes, as indicated by FDR_{Σ} .

For each nBin value, there is minimal variation in both FDR and extraction time. The minimal effects of nBin may be attributed to the fact that the euclidean distance to the generated histogram is taken as the measure for Directionality, further reducing the effects of each variation in nBin.

7.1.4 Line-likeness

As discussed in section 4.4, the number of nBins chosen for Directionality was applied to the computational model of Line-likeness. For this reason, the measure of Linelikeness in table 7.5 was generated with an nBin input of 12.

7.1.5 Regularity

As presented in table 7.3, the effects of particular sub-image distributions is investigated. The original distribution of 2x2, as defined by Tamura's features, did not perform as well

Sub-image Distribution	FDR				Extraction Time (s/image)
	NB	CN	BC	Σ	
2x2	0.0213	0.0082	0.0025	0.0320	18.48
2x3	0.0525	0.0181	0.0069	0.0775	18.27
3x5	0.0291	0.0000	0.0000	0.0291	17.26

Table 7.3: Fischer's Discriminant Results for Class Pairs: Regularity

as the more appropriately chosen 2x3 distribution. The sub-image distribution of 2x3 was chosen as it provided the most separability between classes. Although a faster extraction time for 3x5 was observed, the FDR results were greatly hindered.

7.1.6 Roughness

As presented in table 7.4, an alternate model to Tamura's model of Roughness is tested.

Tamura's model provided very little in terms of the ability to distinguish between classes. The definition recommended by Bianconia et al provided a considerable improvement to Tamura's model [5], however, as to whether the alternate definition accurately responds to a visually 'rough' image remains untested.

Additionally, the extraction time for Tamura's model of roughness is generated based on the assumption that Coarseness and Contrast have already been computed. If Tamura's

Model	FDR				Extraction Time (s/image)
	NB	CN	BC	Σ	
Tamura	0.0000	0.0017	0.0018	0.0035	0.40
ISO:25178-2 S_q	0.2153	0.1367	0.0051	0.3571	0.65

Table 7.4: Fischer's Discriminant Results for Class Pairs: Roughness

model of roughness were to be considered individually, the extraction times would be increased considerably.

7.1.7 Optimal Set

The six features corresponding to visual perception have been optimized as to provide data most beneficial to the two class support vector machines. Table 7.5 serves to rank the six features according to the amount of separability each feature provides. The features are therefore ranked according to FDR_{Σ} .

Feature	FDR				Extraction Time (s/image)
	NB	CN	BC	Σ	
Directionality	0.3098	0.1496	0.0165	0.4759	1.49
Roughness	0.2153	0.1367	0.0051	0.3571	0.65
Coarseness	0.2108	0.0616	0.0493	0.3217	12.88
Line-likeness	0.1894	0.0972	0.0041	0.2907	5.15
Contrast	0.1299	0.0364	0.0223	0.1886	1.21
Regularity	0.0525	0.0181	0.0069	0.0775	18.27

Table 7.5: Fischer's Discriminant Results for Class Pairs: Optimal Feature Set

As demonstrated in table 7.5, Directionality provides the most overall separability as indicated by FDR_{Σ} . Surprisingly, the definition of Roughness recommended by Bianconia et al performed considerably well given the simplicity of its definition. These top two features take very little time to extract, however provide very little in terms of separating Benign and Cancer images.

The two outliers, Coarseness and Contrast, provided the highest results for FDR_{BC} . Although Coarseness does take considerably long to compute, it does provide the most separability for Benign and Cancer images, when other features are incapable of doing so.

A common trend across all of these features is that FDR result for Normal and Benign images is considerable larger than FDR results for all other class pairs. For all features:

$$FDR_{NB} > FDR_{CN} > FDR_{BC}$$

Seeing as the generalization ability of a two class support vector machine is directly affected by the separability of the feature data it is trained with, this hierarchy of FDR results is expected to be mirrored in the two class support vector machine performance results.

The separability of data may be better visualized through the the following probability distribution curves for each of the six features. The PDF curves were plotted using the feature data generated from the optimal set of features. To better explain what these

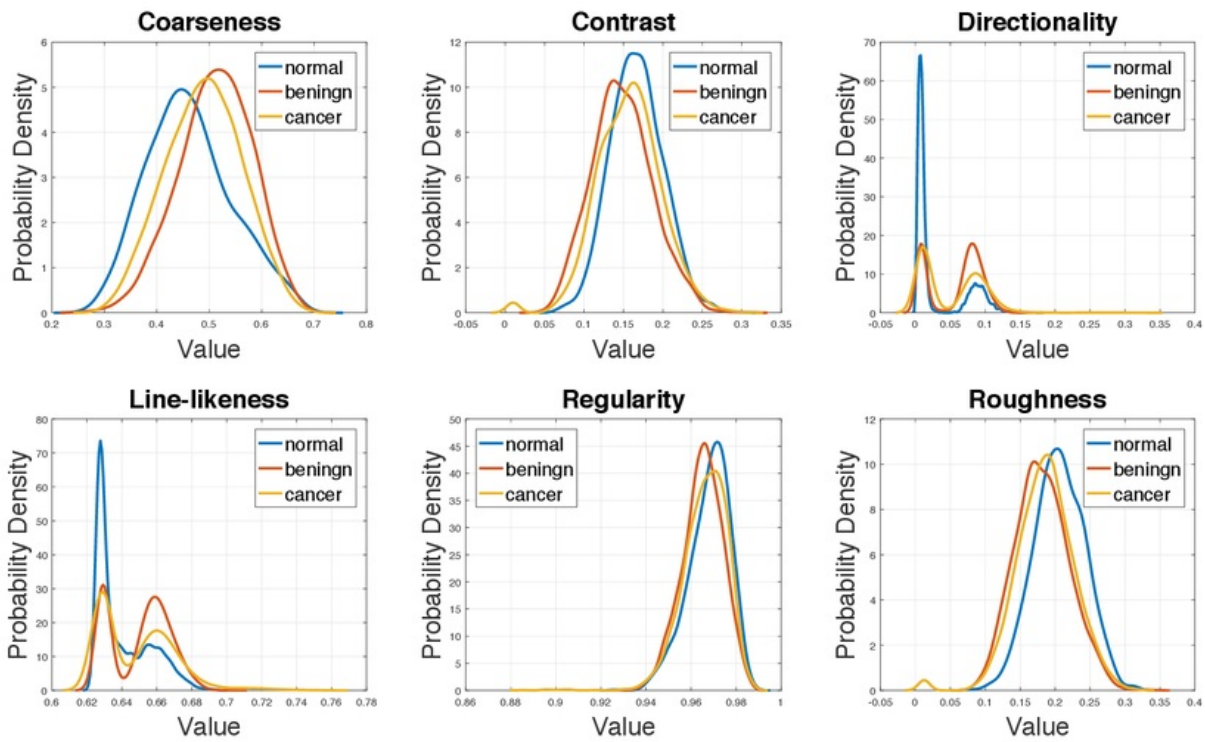


Figure 7.1: PDF Curves for Optimal Feature Models

curves are representing, the FDR results for Directionality is considered. The PDF for Directionality demonstrates a high probability that a Normal breast image will generate a low measure of Directionality, in comparison to the results for Benign and Cancer

images. Such a distinction is represented by the comparatively ‘high’ results for FDR_{NB} and FDR_{CN} . However, as demonstrated by the similarities in PDF curves generated by Benign and Cancer images in terms of shape and peak positions, the result for FDR_{BC} is comparatively low.

7.2 Two Class Performance

The following performance measures assess the performance of the two class support vector machines. For each class pair, the performance of each kernel type was investigated.

As mentioned in section , k-fold cross validation is applied. Such process aims to measure the performance of the two class support vector machine, based purely on the training data. The following classification rates are based on the average performance over 10-folds.

Kernel	10 Fold Average Performance %			Training Time (s)
	NB	CN	BC	
Linear	77.21	70.47	60.66	1.87
Polynomial, p=3	77.38	70.95	60.61	1.93
Polynomial, p=5	77.88	71.33	63.09	4.71
RBF	79.28	71.63	63.91	2.81

Table 7.6: Two Class SVM Performance: Kernel Types

As presented in table 7.6, the positive classification rates associated with the RBF kernel is the highest over the 10 folds. The performance of the Polynomial kernel with polynomial order 5 is comparable however the training time associated with such a kernel is higher. For the given results, there is no reason not to choose the RBF kernel.

At this point, the two class classification rates may be properly assessed as to evaluate the performance of Tamura’s features as a means of distinguishing between image class pairs.

As predicted in section 7.1.7, the performance of the two class classifiers correlated highly with the separability of feature data for each class pair. The results for the classification for NB and CN class pairs is quite impressive, with positive classification rates of 79.28% and 71.63%. However, the classification rate for BC is considerably lower with result of 63.91%.

The statement may be made that Tamura’s features is suitable for making the distinction between Normal and abnormal images, where the abnormal set consists of Benign and Cancer images. However, to assess the severity of the abnormality, Tamura’s features

does not provide data capable of making a clear distinction between Benign and Cancer images.

To better understand the the specific source of classified or misclassified data, the following confusion matrices are produced.

		Predicted Class		Positive Rate (%)	Negative Rate (%)
		N	B		
Actual Class	N	130	55	70.27	29.73
	B	28	206	88.03	11.97

Table 7.7: Confusion Matrix: NB

		Predicted Class		Positive Rate (%)	Negative Rate (%)
		C	N		
Actual Class	C	212	44	82.81	17.19
	N	79	105	57.07	42.93

Table 7.8: Confusion Matrix: CN

As demonstrated by the confusion matrices for NB and CN class pairs, there exists a common trend of Normal images being misclassified as either Benign or Cancer images, respectfully. Particularly as a result of the CN classifier, the rate in which Normal is misclassified as a Cancer images is substantially high at a 42.93%.

		Predicted Class		Positive Rate (%)	Negative Rate (%)
		B	C		
Actual Class	B	139	103	57.44	42.56
	C	67	161	70.61	29.39

Table 7.9: Confusion Matrix: BC

As indicated by the performance results in table 7.6, the classification rate for the BC classifier is considerably low at 63%. As demonstrated in the confusion matrix for the BC classifier, the rate of which Benign images is correctly classified is very low at 57.44%. We may conclude that Tamura's features do not provide the relevant feature data to properly aid the separation of the Benign and Cancer images.

7.3 Multiclass Performance

The following table represents the performance results for the multiclass implementations outlined in section 6.2.2.

Multiclass Method	Classification Rate (%)	Classification Time (s)
Standard	57.7677	5.57
Fuzzy Average	59.3422	5.63
Fuzzy Min	58.8523	5.68
DDAG N	58.1875	5.66
DDAG B	58.0126	5.46
DDAG C	58.5724	5.58

Table 7.10: Multiclass SVM Performance

As demonstrated in table 7.11, the Fuzzy multiclass solution with Average voting performed the best with a positive classification rate of 59.3422% over the three classes. As a result of the two class classification rates, the performance of any multiclass solution was not expected to provide any spectacular results. As clearly defined, the multiclass solution does not merely provide classification rate equivalent to the average two class classification results. The majority vote is highly susceptible to votes cast from any sub-performing two class classifier.

Chapter 8

Conclusion

The objective of this research project was to assess the performance of Tamura's features in the context of Breast Image Classification using Machine Learning Techniques.

To be able to make such an assessment, the computational models defined by Hideyuki Tamura in his paper 'Textural Features Corresponding to Visual Perception' were practically implemented in MATLAB.

The six features were applied to a Mammographic Image set that was provided by Macquarie University, as to extract information relevant to each Mammographic Image class: Normal, Benign or Cancer. The extracted feature data was used to train Support Vector Machines as to generate models capable of distinguishing between Mammographic Image classes.

The following classification rates best summarize the performance of the the outlined experimental process:

(i) Normal versus Benign	79.23%
(ii) Cancer versus Normal	71.63%
(iii) Benign versus Cancer	63.91%

It may be concluded that Tamura's features are effective in making the distinction between Normal and Abnormal Mammographic classes, as indicated by classification rates (i) and (ii). Although the process does not provide an absolute classification, it may serve to aid the decision making process undertaken by a healthcare professional.

Chapter 9

Future Work

As stated in the conclusion, experimental method outlined throughout this report is capable of making the distinction between Normal and Abnormal Mammographic classes quite effectively. However, the ability to assess the severity of the abnormality remains substandard. As an extension to the current experimental method the following areas may be investigated.

Feature Optimization

Although a feature optimization process was undertaken, the extent to which it taken was limited due to time constraints. Particularly the results generated by alterations in the Coarseness model proved to give promising results.

Introduction of New Features

Given the circumstance that the six features are exhausted, that is, no further performance benefits are witnessed — new features may be introduced. Research into existing solutions will be beneficial to this process.

Method of Feature Application & Alternative Image Sets

As mentioned in chapter 3, the particular application of Tamura's features is such that the entire image is analyzed. Reference to applications involving 'regions of interest' (ROI) is discussed, although ruled-out due to the limitations of the provided image set. Further investigation into ROI implementations is recommended as a direct extension to the current experimental setup, in combination with an image set capable of providing relevant meta-data.

Usability

The experimental setup outlined in this project may better serve an application that does not involve Breast Images. By improving the usability of the MATLAB code, particularly the implementations of Tamura's features, the experimental setup may be applied to other image classification scenarios.



Chapter 10

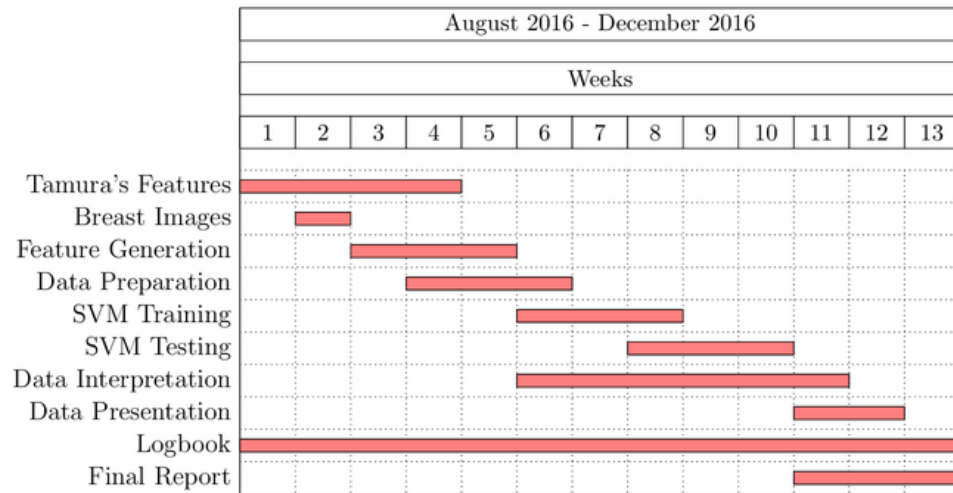
Abbreviations

SVM	Support Vector Machine
RBF	Radial Basis Function
OAA	One Against All
OAQ	One Against One
DDAG	Decision Directed Acyclic Graph
CRS	Coarseness
CON	Contrast
DIR	Directionality
LIN	Line-likeness
REG	Regularity
RGH	Roughness
FDR	Fishers Discrimination Ratio
PDF	Probability Density Function
ROI	Region of Interest

Appendix A

Scheduling & Planning

A.1 Project Timeline



A.2 Consultation Meetings Attendance Form

Consultation Meetings Attendance Form

Week	Date	Comments (if applicable)	Student's Signature	Supervisor's Signature
1	1/8/16	Plan/Schedule	Amir Alcin	MM
2	8/8/16	Breast Images	Amir Alcin	MM
3	19/8/16	Tomera Features	Amir Alcin	MM
4	26/8/16	Data prep for SVM	Amir Alcin	MM
5	30/8/16	SVM	Amir Alcin	MM
5	01/9/16	MULTICLASS SVM	Amir Alcin	MM
7	13/9/16	progress report comments	Amir Alcin	MM
7	16/9/16	code usage	Amir Alcin	MM
MID SEM	28/9/16	performance	Amir Alcin	MM
8	5/10/16	performance and final report	Amir Alcin	MM

Figure A.1: Consultation Meetings Attendance Form

Bibliography

- [1] S. Abe, "Support vector machines for pattern classification," *London : Springer*, 2005.
- [2] B. C. C. WA. (2013, February) About breast cancer: Statistics. [Online]. Available: <http://www.breastcancer.org.au/about-breast-cancer/statistics.aspx>
- [3] T. Y. H. Tamura, S. Mori, "Textural Features Corresponding to Visual Perception," *IEEE Transactions on Systems, Man. and Cybernetics*, June 1978.
- [4] S. R. P. Howarth, "Evaluation of Texture Features for Content-Based Image Retrieval," *CIVR 2004*, 2004.
- [5] A. F. Francesco Bianconia, Alberto lvarez-Larrnb, "Discrimination between tumour epithelium and stroma via perception-based features," *Neurocomputing*, April, 2015.
- [6] S. Y. H. L. B. Wang, H. Bao, "Crowd Density Estimation Based on Texture Feature Extraction," *Journal of Multimedia*, August 2013.
- [7] S.-N. Y. Hsin-Chih Line, Chih-Yi Chieu, "Texture Analysis and Description in Linguistic Terms," *Asian Conference on Computer Vision*, January 2002.
- [8] A. F.-G. S. A. M. G. Orru, W. Petterson-Yeo, "Using Support Vector Machine to identify imaging biomarkers of neurological and psychiatric disease: A critical review," *Neuroscience and Biobehavioural Reviews*, 2012.
- [9] M. Pal, "Multiclass Approaches for Support Vector Machine Based Land Cover Classification," *National Institute of Technology , Kurukshetra*, February 2008.
- [10] B. Aisen. (2006, December) A comparison of multiclass svm methods. [Online]. Available: <http://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/>
- [11] V. Vapnik, "The nature of statistical learning theory," *London : Springer-Verlag*, 1995.
- [12] V. Vapnik, "Statistical learning theory," *Wiley-Interscience*, 1998.
- [13] U. of Essex. (2012, December) The mini-mias database of mammograms. [Online]. Available: <http://peipa.essex.ac.uk/info/mias.html>

- [14] F. D. Francesco Bianconi. (2013, February) Discrimination between tumour epithelium and stroma via perception-based features. [Online]. Available: <http://www.breastcancer.org.au/about-breast-cancer/statistics.aspx>
- [15] MathWorks. Histogram bin counts. [Online]. Available: <https://au.mathworks.com/help/matlab/ref/histc.html>
- [16] ISO, "Geometrical product specifications (gps) — surface texture: Areal," International Organization for Standardization, Geneva, Switzerland, ISO 25178-2, 2012.
- [17] WolframMathWorld. Root-mean-square. [Online]. Available: <http://mathworld.wolfram.com/Root-Mean-Square.html>
- [18] s. M. Mozos, "Semantic labeling of places with mobile robots," *Springer-Verlag Berlin Heidelberg*, 2010.
- [19] MathWorks. Statistics and machine learning toolbox. [Online]. Available: <https://au.mathworks.com/products/statistics/>
- [20] —. Predict labels using discriminant analysis classification model. [Online]. Available: <https://au.mathworks.com/help/stats/compactclassificationdiscriminant.predict.html>