The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)________________________
Summary

Barrett’s Oesophagus is a pre-malignant condition that is estimated to effect 1.6%-6.8% of the population with a potential risk factor for leading to the development of cancerous cells of 10%. Because of this, it is important that pathologists are able to correctly identify the level of severity of the condition in patient’s tissue samples by analysing the grade of dysplasia, the abnormal development of cells. However, pathologists are found to often disagree due to their differing years of experience and the difficulty of the task itself.

It is therefore desirable to find automated ways of enabling computers to capture general information about what it is pathologists look for to create systems that could assist pathologists in this grey area of disagreement.

This project investigates the implementation of two types of such systems. Both systems attempt to learn what features in digital pathology images characterize different grades of dysplasia. The first system uses features based on texture analysis, and the second uses features based on nuclear and spatial analysis.
Acknowledgements

I would like to thank my supervisor, Dr. Andy Bulpitt. He has supported me throughout the completion of this project and his guidance has taught me several useful skills and enabled me to achieve what I have. He was professional in his conduct towards meetings to which he was always fully engaged. He was also always willing to give help if I needed it and gave useful feedback when I justified my decisions to him.

I would also like to thank Dr. Darren Treanor, this projects consultant pathologist, for meeting with me on several occasions and helping me understand the medical background to the project. I also want to thank him for the feedback he gave me on the work I had done as well as spending time to go through the falsely classified test images and giving his feedback on those.

Thanks also have to go to Alexander Wright, who works with Darren at the Leeds Institute of Molecular Medicine, for showing me around the lab there and giving me useful tips on Matlab and some of its different functions.

Duane Carey, a post-doc at the University of Leeds, deserves thanks for spending time with me trying to get the provided nuclear detection and colour normalisation methods working and for replying swiftly to emails regarding the different questions I had about them.

A final thanks goes to Afzan Adam, whose work forms a basis of part of this project. She kindly answered some of my questions regarding her work in a prompt and informative manner.
Contents

1 Introduction 1
  1.1 Overview .......................................................... 1
  1.2 Report Structure ................................................. 1
  1.3 Aims and Objectives ............................................. 2
    1.3.1 Aim .......................................................... 2
    1.3.2 Objectives .................................................. 2
  1.4 Requirements .................................................... 2
    1.4.1 Minimum Requirements .................................... 2
    1.4.2 Possible Further Extensions .............................. 2
  1.5 Deliverables ..................................................... 3
  1.6 Relevance to degree program ................................... 3

2 Background Reading 4
  2.1 Barrett’s Oesophagus ............................................ 4
  2.2 Image Data ........................................................ 6
    2.2.1 Tissue Slides .............................................. 6
    2.2.2 Staining ..................................................... 6
    2.2.3 Digitisation ................................................ 6
  2.3 Colour Deconvolution .......................................... 7
  2.4 Colour Normalisation .......................................... 8
  2.5 Texture Analysis Techniques .................................. 8
    2.5.1 Grey-Level Co-occurrence Matrix (GLCM) .................... 8
    2.5.2 Local Binary Pattern (LBP) ................................ 9
    2.5.3 Gray Level Run Length Matrix (GLRLM) ..................... 10
    2.5.4 Gray Level Difference Method ............................. 11
    2.5.5 Conclusions ................................................ 11
  2.6 Spatial Analysis Techniques .................................. 12
    2.6.1 Delaunay Triangulation (DT) .............................. 12
    2.6.2 Voronoi Tessellations (VT) ............................... 13
    2.6.3 Minimum Spanning Trees (MST) ............................ 13
    2.6.4 Conclusions ................................................ 13
  2.7 Machine Learning Aspects ..................................... 14
    2.7.1 Supervised Learning ....................................... 14
## Evaluation

6.1 Giving context to statistics .................................................. 38
6.2 Confusion Matrix Statistics ................................................... 39
6.3 Texture Analysis Evaluation ................................................... 40
6.4 Nuclear Analysis Evaluation .................................................. 41
6.5 Stability ................................................................................. 42
  6.5.1 Stability Conclusion .......................................................... 45
6.6 Texture Analysis Evaluation ................................................... 45
  6.6.1 Textural Analysis Decision Tree for Grade 1 ......................... 46
  6.6.2 Textural Analysis Decision Tree for Grade 3 ......................... 47
  6.6.3 Textural Analysis Decision Tree for Grade 5 ......................... 48
  6.6.4 Analysis of Tree-G1Tex ....................................................... 49
  6.6.5 Analysis of Tree-G3Tex ....................................................... 50
  6.6.6 Analysis of Tree-G5Tex ....................................................... 50
6.7 Nuclei Analysis Evaluation .................................................... 51
  6.7.1 Nuclear Analysis Decision Tree for Grade 1 ......................... 52
  6.7.2 Nuclear Analysis Decision Tree for Grade 3 ......................... 53
  6.7.3 Nuclear Analysis Decision Tree for Grade 5 ......................... 54
  6.7.4 Analysis of Tree-G1Nuc ....................................................... 55
  6.7.5 Analysis of Tree-G3Nuc ....................................................... 55
  6.7.6 Analysis of Tree-G5Nuc ....................................................... 56
6.8 Pathologist Feedback ............................................................ 56
6.9 Data Remark ............................................................................ 56
6.10 Methodology Evaluation ........................................................ 57

## Conclusion

7.1 Summary ............................................................................... 58
7.2 Aims and Objectives .............................................................. 58
7.3 Limitations .............................................................................. 59
7.4 Future Work ........................................................................... 60

Bibliography .................................................................................. 61

A Personal Reflection ...................................................................... 65

B External Materials ....................................................................... 67

C Ethical Issues .............................................................................. 68

D Consultant Pathologist Responses ............................................. 69

E Schedule ...................................................................................... 71
F  Code Evaluation  
  F.1  Improvements  ................................................................. 73  
  F.2  Code Timings ................................................................. 73  

G  Node Labelling Method Texture Analysis  
  G.1  Decision Tree Node Labelling ........................................ 75  
    G.1.1  Tree-G5Tex Node Labelling ........................................ 75  
    G.1.2  Tree-G3Tex Node Labelling ........................................ 75  
    G.1.3  Tree-G1Tex Node Labelling ........................................ 76  

H  System Design  ................................................................. 77
Chapter 1

Introduction

1.1 Overview

Barrett’s Oesophagus (BO) is a condition that affects the digestive tract. It causes the cells in the tissue lining of the oesophagus to change into abnormal forms. BO has been described as a pre-malignant (pre-cancerous) although treatable condition and has been linked to increased risks of developing of cancerous cells such as Adenocarcinoma [37], which has been estimated to have a risk factor of 10% in BO sufferers.

When a patient is suffering from concerning symptoms, they will visit a Gastroenterologist who will give them an endoscopy examination at which time they may have a biopsy, the medical removal of a sample of cells or tissues for observation to determine the presence or severity (grade) of a disease. This tissue is typically observed under a microscope by a pathologist, who uses their own experience and knowledge in conjunction with sets of diagnostic features they are trained to look for. The fact that the diagnosis of cancer is not just a clean cut set of rules and relies greatly on the pathologists individual experience makes the whole process subjective and prone to error [42]. Eye tracking is one possible way of learning the cognitive process an expert pathologist uses, but this requires specialist equipment not readily available as well as constricting the standard use of a microscope such as zoom and panning capabilities. The latter problem can be addressed by using glass slides, which allows video recordings to be taken of what is being viewed down a microscope, which can be paired with the pathologists commentary to learn their diagnostic process.

Virtual slides are used to address the same problem but in a more powerful way. A glass slide is scanned at high resolution to produce an image which can be viewed on a computer, which has various advantages as observed in [42]. Digitization allows annotations to be made, such as highlighting certain regions along with a diagnosis, which can then be extracted at a later date. This latter data is what forms the basis of this project.

1.2 Report Structure

The rest of this chapter lays out the aims and objectives of the project, and specifies an outline of the problem that is to be tackled. The second chapter contains background research that
has been carried out to gain an improved understanding of the problem that has to be tackled, and appropriate methods and their alternatives that could be used. The third chapter specifies any key design choices that were made as well as summarises the project plan and methodology that will be adopted to solve the problem. The fourth and fifth chapters contain details for the implementation phase. The sixth chapter contains an evaluation of the produced solution.

1.3 Aims and Objectives

1.3.1 Aim

The overall aim of this project is to develop a methodology to support pathologists in the diagnosis of Barrett’s Oesophagus, to help shed light on the grey areas where pathologists disagree. The methodology will analyse digitised annotated regions extracted from virtual slide images and learn what sort of textural and spatial characteristics contribute towards the classification of the grade of dysplasia present, and apply this knowledge to new images.

1.3.2 Objectives

- Understanding the diagnosis of Barrett’s Oesophagus in histopathology
- Classification of patches based on clustering image textures
- Evaluation of the frequency of textures for grading Barrett’s Oesophagus
- Investigation of use of spatial information

1.4 Requirements

1.4.1 Minimum Requirements

- Produce a methodology that is capable of taking in an input image of a region of tissue from a virtual slide and producing a new image that describes the clusters of textures within it
- Implement machine learning techniques to train three classifiers for distinguishing between whether a new image belongs to a certain grade or not for three different grades
- Evaluate results in comparison to those produced in previous experiments [2]

1.4.2 Possible Further Extensions

- Investigate link between spatial relationships of nuclei and the grade of the tissue
- Investigate link between spatial relationship of areas of texture types and the grade of the tissue
1.5 Deliverables

- Implementation of software, including Matlab code
- Evaluation of software classification against ground truth classes as well as evaluation of code

1.6 Relevance to degree program

This project makes use of knowledge and skills taught as part of a MEng Computer Science degree program at the University of Leeds in the following modules:

- Machine learning skills were taught extensively in the year 2 module Artificial Intelligence (COMP2240). These sort of skills will need to be applied to learn patterns within data
- General project management skills, particular the issues with the estimation of time requirements, were taught in the first year module Project Management (COMP1945).
- The management of deliverables and code throughout the development of a project including version control were covered in Software Systems Engineering (COMP2540).
- Understanding images and how it’s possible to manipulate them was covered in the third year Computer Vision module (COMP3300). This also taught Matlab skills which will be relevant here.
Chapter 2

Background Reading

This section contains the background reading that was done in order to acquire knowledge that would facilitate the completion of this project. We start by introducing Barrett’s Oesophagus and the necessary understanding of biology required for this project, as well as introduce some key characteristics that form potential candidates of features to analyse. We then give background information on how images of tissue are obtained from a patient. Specific information regarding the data used by this project can be found in Section 3.7.

Images containing biological structures could be described by their texture, as the structures make a constant image a textured one. Or these structures could be described by a set of points in 2-D space, and their spatial relations could be used. These are two methods of trying to describe same thing. The rest of this chapter is split into two sections. One gives information on various texture analysis techniques that could be employed in order to analyse digital pathology images, the second section does the same for spatial techniques. We then finish by looking at ways we could observe patterns in the extracted data.

2.1 Barrett’s Oesophagus

The Oesophagus is the internal tube that connects the mouth and stomach. Barrett’s Oesophagus (BO) is a condition often caused by bile/acid coming up from the stomach into the lower oesophagus, whose corrosive properties cause damage to the cells after repeated exposure. It is estimated to affect 1.6-6.8% of people [10].

Normally the oesophagus lining is described as normal squamous (mean scale-like, flat) lining formed from flat round squamous cells, but due to the exposure of acid from the stomach they become mucus secreting columnar (less round and more rectangular) cells, goblet cells. This is actually known as metaplasia, which is the change of cells from one form to another, whether it be due normal maturation or due to reacting to some external effect. This term is not to be confused with dysplasia which refers more to the abnormal development of cells.

Initially this mucus serves to protect the lining, but the continued effects of the stomach acid can cause the changes to spread into deeper layers, causing further cytological changes to cells. This can then go on to increase in severity, ultimately progressing through six stages starting with non dyplastic Barrett’s Oesophagus then going on to: indefinite for dyplasia
(probably negative for dysplasia), indefinite for dysplasia (probably positive for dysplasia), low grade dysplasia (LGD), high grade dysplasia (HGD) and then intramucosal carcinoma (cancer).

As the tissue progresses through these grades different architectural and cytological (of relating to cells) changes undergo. The tissue surface transforms from a flat surface to a rough surface going up and down but in a more regular way then to a progressively more irregular rough pattern, Figure 2.1(a). The nuclei become more irregular in shape and size, Figure 2.1(b), and potentially develop hyperchromatism which is the darkening of nuclei due to an increase in the coarseness of the DNA within them [39]. The cells also become more bunch/crowded, going from an ideally single lined formation to grouped formations. Tissue examples of some of these transformations can be seen in Figures 2.2(a) and 2.2(b). It is important to note that this is an ideal high level description, and as mentioned in Section 1.1 it is not as clean cut as this.

![Figure 2.1: High-level 'ideal' feature progression](image)

Once a diagnosis has been made, if the conclusion is that it's BO the patient will attend regular check ups to monitor its progression, as not all BO sufferers will develop oesophageal cancer. When the condition is observed to have the potential to develop into a dangerous level of dysplasia, more direct treatments may be necessary such as surgery or chemotherapy.
2.2 Image Data

This section gives a brief background to the techniques used by the Leeds Institute of Molecular Medicine (LIMM) to obtain digital images for the analysis of the presence and severity of BO.

2.2.1 Tissue Slides

The first stage to obtaining the high resolution images is to create the physical slide that the image is taken from. A biopsy is performed on a patient. The extracted tissue is encased in wax which is then solidified to keep the tissue sample in a rigid state, enabling a very thin slice of wax and tissue to be cut. The wax from this slice is then removed using hot water and the tissue is placed on a glass slide, and passed on to be stained.

2.2.2 Staining

Staining is the process in which certain chemicals are applied to tissue samples in order to highlight certain aspects. This works because the different chemicals react in different ways to different biological matter, examples of which are proteins, cytoplasm and nuclei. This facilitates the analysis of the tissue by a pathologist, allowing for an easier examination. It also makes it easier to apply certain computer vision techniques, such as segmentation and texture analysis, as the contrast in colour makes it easier to differentiate between structures.

*Haematoxylin and Eosin* (H&E) staining is a staining method whose name is derived from the chemical substances used. It is a popular method that has been adopted for many years and is one of the ones used at LIMM. Haematoxylin reacts with nucleic acids to stain them a dark blue-purple colour whereas Eosin reacts with the cytoplasmic proteins and stains them a lighter pink colour [11] (see Figure 2.3).

![Figure 2.3: Example of Haematoxylin and Eosin staining showing oesophageal squamous cells](image)

2.2.3 Digitisation

The next stage is to put the images onto computers, a process called digitisation. LIMM uses high resolution scanners provided by Aperio, which produce high resolution virtual slide images at various magnifications and can be billions pixels in size, reaching into the many GB's of storage. Consequently, the images reside on a server which can then be accessed remotely.
2.3 Colour Deconvolution

Colour deconvolution is a method in which we determine the contribution of different colours within an image, and in this case can be used to find out the contribution of specific stains within tissue samples. Whereas staining was used before to create a greater contrast between the different biological matter, colour deconvolution creates another image that gives an improved impression to where each stain is present, which may be desirable for various analysis purposes.

Colour deconvolution is described by Ruifrok and Johnston [35]. They describe how the transmission of light from the tissue is related to the concentration of a stain in a nonlinear way. This consequently causes the grey values of each channel to also exhibit nonlinear behaviour with respect to stain concentration. This means separation of the stains cannot be achieved by analysis of intensity values in the images, and requires a linear relation to concentration to be defined. This is known as the optical density (OD) of a stain, whose calculation with respect to the intensity of light entering and leaving a specimen for a specific detection channel is explained in Ruifrok and Johnston’s paper. By calculating the OD for each of the RGB channels of each of the stains, an OD matrix is formed:

\[
\begin{bmatrix}
R & G & B \\
0.18 & 0.20 & 0.08 \\
0.01 & 0.13 & 0.01 \\
0.10 & 0.21 & 0.29 \\
\end{bmatrix}
\]

In this case Ruifrok and Johnston used a combination of staining techniques, using H&E with DAB (diaminobenzidine, a brown stain). This matrix is then normalized to give the matrix \( M \). Multiplying the inverse of this matrix by the OD levels at a particular pixel gives a 3x1 vector representing the amounts of the three stains at that pixel. These three values across all the pixels can be used to form new images representing each stain, Figure 2.4. Figure 2.4(b) and Figure 2.4(c) show these new images for Hematoxylin and Eosin stains respectively. In this instance, the darker the colour means the higher the concentration of the stain.

Figure 2.4: Colour Deconvolution Example
2.4 Colour Normalisation

Colour normalisation is a common pre-processing technique to carry out on pathological images when doing image analysis due to the subtle changes that can occur in the biopsy process. Two main factors that contribute to these changes are varying thicknesses of the tissue samples that are taken which effect how well the sample absorbs the dye: a thicker sample will typically appear darker, and the actual concentration of the dye used which will have a greater chance of varying if the samples were dyed at different times in different sessions. One method developed by Reinhard et al. [34] normalises a reference image based on the colour characteristics of a target image. However, this method has a problem in which it introduces noise into the image background, which ideally should be ignored by the normalisation process.

An alternative implementation developed by Magee et al. [26] overcomes some of the limitations associated with this method, and is suggested to be more robust to variable staining and less prone to the introduction of background artefacts.

2.5 Texture Analysis Techniques

In order to automate the grading of Barrett’s Oesophagus using computers, a property/feature of the extracted images of the tissue needs to be chosen in order to create some measure of differentiation between the different grades. Image texture is one such quality that this report investigates. There has been several attempts at using image texture in relation to the location/grading of dysplastic tissue, for example [17] [8] [32] [29] [47]. A review of the literature highlighted four texture analysis techniques in particular: Grey-Level Co-occurrence Matrices; Local Binary Patterns; Grey-Level Run Lengths; and Grey-Level Difference Methods.

2.5.1 Grey-Level Co-occurrence Matrix (GLCM)

A GLCM is a matrix that measures how often different combinations of grey levels at a given distance and angle occur in an image. If an image has \( N \) unique grey levels, consider each pixel as being labelled with an integer from \( 1...N \) corresponding to the grey level it possesses. The sample image in Figure 2.5 shows this labelling procedure for a simple 16-pixel image with 3 unique grey levels. The four other tables show the GLCM’s for different angles using a pixel offset, how far away in pixels we consider pairs of pixels, of 1. 0\(^\circ\) means we are looking at pixels immediately to the right of the reference pixel, 45\(^\circ\) pixels along the upward right bearing diagonal and so on. Note how the tables are \( N \) by \( N \) in size. The tables are generated by considering two pixels at a time and by looping through all the pixels in the image considering each one as the reference pixel and incrementing the i,j cell in the GLCM by 1, where i and j are the gray-level labels for the reference and neighbour pixel respectively.

A GLCM can be symmetrical or asymmetrical. The ones shown in Figure 2.5 are symmetrical: for every pair of pixels, i.e. (2,3), they only increment the cell in the table for one direction, in this case just (2,3). A symmetrical GLCM would increment the cells for (2,3) and (3,2).

There are a variety of statistical measures that can be derived from GLCM’s [19] [18]. The most popular are commonly termed \textit{Contrast}, \textit{Correlation}, \textit{Energy} and \textit{Homogeneity} (Table
Figure 2.5: Four different GLCM’s using the same pixel offset but different angles

<table>
<thead>
<tr>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td>( \sum_{i=1} (i-j)^2 p(i, j) )</td>
</tr>
<tr>
<td>Correlation</td>
<td>( \frac{\sum_{i=1} (i-\mu)(j-\mu) p(i, j)}{\sigma_i \sigma_j} )</td>
</tr>
<tr>
<td>Energy</td>
<td>( \sum_{i,j} p(i, j)^2 )</td>
</tr>
<tr>
<td>Homogeneity</td>
<td>( \sum_{i,j} \frac{p(i, j)}{1+</td>
</tr>
</tbody>
</table>

Table 2.1: Four statistics derived from GLCM’s

2.1. In this figure, \( p(i, j) \) corresponds to the \((i, j)\) value in the GLCM.

2.5.2 Local Binary Pattern (LBP)

In 1990, Li Wang and Dong-Chen He [20] proposed the Texture Spectrum model. They introduced the concepts of a texture unit and texture spectrum. A texture unit is considered as the smallest complete unit that best characterizes the local texture of a particular pixel and its neighbours and a texture spectrum is the frequency distribution of texture units for a particular image. They consider an 8-direction square raster which they use to transform a neighbourhood of grey levels into a texture unit (Figure 2.6). Figure 2.6(a) shows the criteria under which a neighbourhood is transformed into a texture unit where \( TU = \{E_1, E_2,...,E_8\} \). This gives \( 3^8 = 6561 \) possible texture units which can be labelled using a labelling method described in [20].

Taking a two-level approach to this method gives the LBP method [30]. Instead of labelling each neighbour of the centre pixel with 0, 1 or 2, they are labelled 0 or 1 by instead using the centre pixel as a threshold value. From doing this, an 8-digit binary number can be obtained which can be chosen to be converted to its decimal equivalent.

This process could be repeated for each pixel in a window, and even for different neighbourhood sizes, and a histogram of the frequency of each binary/decimal number calculated can be formed. This histogram can then be used as a feature vector from which various analysis
methods can be applied such as clustering and classifying.

In [30] it was shown that LBP could be combined with two different features, a contrast measure and a covariance measure, to form two new adapted methods. This resulted in more powerful methods that resulted in lower error rates than using LBP alone. A downside to LBP is that it is rotation variant, which may not be desirable in certain situations, for instance a vertically striped pattern might want to be classified the same as an equivalent horizontally striped pattern, but it is possible to derive a rotation invariant version [31]. It is also possible to derive a grey scale invariant version, which may be desired for some applications.

### 2.5.3 Gray Level Run Length Matrix (GLRLM)

A **gray level run** is a set of consecutive pixels in an image that have the same grey level value [13]. The number of pixels in a particular run represents its **length**. As in section 2.5.1, the same grey level labelling procedure is adopted. A Gray Level Run Length Matrix (GLRLM) is constructed where an \((i,j)\) element in the GLRLM corresponds to the number of times the image contains a run of length \(j\) of pixels having grey level \(i\). Similarly to a GLCM, different directions based on angles can be specified (see Figure 2.7).

Similar statistical measures to those used by [19] for GLCM’s can also be used for GLRLM’s. These are specified in [13] that uses a similar notation for presenting values in the table as was used in section 2.5.1.

![Image](image_url)

**Figure 2.7:** Example image with labelled grey levels and corresponding GLRLM’s in two directions

It is possible to group gray-levels and run lengths into ranges. This allows smaller matrices or matrices of a desired size to be generated, which may be particularly useful when the number
of different possible lengths of runs is high and memory usage wants to be kept to a minimum.

2.5.4 Gray Level Difference Method

The final method this report will consider is the Gray Level Difference Method. Let an image \( I \) have \( N \) gray-levels and let \( I(i, j) \) be the gray level of the pixel at coordinates \((i, j)\) in \( I \). For a given displacement \( \delta = (\Delta i, \Delta j) \), we can define the function \( f_\delta(i, j) = |f(i, j) - f(i+\Delta i, j+\Delta j)| \)[23][44], so the absolute difference between pixel \((i, j)\) and the pixel at the given displacement, hence the method name. For an image with \( m \) grey-levels it is possible to create an \( m \)-dimensional probability distribution \( p_\delta \) from which several statistics can be derived [44].

2.5.5 Conclusions

Previous work that used both GLCM and GLRLM features to help develop a computer aided classification system to classify between benign and malignant tissue masses from digital mammograms [29] concluded using GLCM texture features yielded greater accuracy results than GLRLM. However, the work also concluded that a combination of both sets of features out performed both individually in terms of accuracy (predicted up to 94.9%). For their work they also listed what they classified as important texture features, amongst which was the standard GLCM features: Energy, Contrast (Inertia), Homogeneity (Inverse Difference Moment).

Doyle et al. [8] uses first order grey level statistical features such as the average, mode and standard deviation across all pixels in the image as well as second-order co-occurrence texture features, GLCMs, extracted from images obtained from hematoxylin and eosin stained breast biopsy tissue in order to distinguish between low and high grades of breast cancer. They used a support vector machine, see section 2.7.1, to classify between cancerous and non-cancerous images, and then low grades and high grades of cancer. For the latter, the second order features performed to a higher accuracy than the grey level features, 0.767 vs. 0.700, and also a combination of all textural features, 0.733.

Zulpe and Pawar [47] achieved a 97.5% classification rate for classifying four different brain tumor types using GLCM based textural features of each class into a two-layered Feed forward Neural Network. The features they used for analysis included all four of the standard classic features mentioned in section 2.5.1.

A comparative study of texture measures used for terrain classification was carried out by [44]. They considered three techniques: Fourier power spectrum, second order grey level statistics (GLCM’s), and first order differences of gray-level differences. For their data they concluded that Fourier features gave a poor performance in classification, but GLCM’s (using Contrast, Angular Second Moment (related to Energy), Entropy and Correlation) and grey level difference methods performed similarly.

After analysing some of the popular choices for texture analysis, Gray-Level Co-occurrence Matrices will be the method of choice. Their wide adoption across the different literature makes it clear their applicability is credible, and their conceptual simplicity combined with this fact makes it a wise choice given the scope and time-frame of the project. Local Binary Patterns are another widely adopted approach, and sometimes in studies they are shown to give more accurate results for classification than GLCM [30], though [3] states that LBP’s produced data
overhead for this particular type of problem which is apparent by the 256 valued feature vectors formed as opposed to the 4 valued feature vectors for GLCM’s (though this number can be increased by considering multiple directions and distances at once, which may be required to get the most out of the method).

When observing other peoples results it is always important to consider what data they were using, and whether the power of their solution will only be apparent to that data. In this case, GLCM’s seemed to have been used on a variety of different medical tissue types and images and produced reasonable results in all. This suggests that the method will be reliably transferable to this projects data.

2.6 Spatial Analysis Techniques

It has already been discussed how the spatial relationships of nuclei as well as the different tissue types can be used to determine the grade and progression of dysplasia. After doing a preliminary literature review, it was concluded that Delaunay Triangulations, Voronoi Tessellations and Minimum Spanning Trees were the three most popular methods used for spatial analysis. Gurcan et al. [16] conducted a review of histopathological image analysis and identified the three same methods, so they are what will be examined further here.

2.6.1 Delaunay Triangulation (DT)

A Delaunay Triangulation on a set of points connects them up in such a way to form triangles such that no point is inside the circumcircle of any triangle. A circumscribed circle is a circle whose edge passes through all three vertices of the triangle. Another characteristic of a DT that drives the creation of the final triangulation is that when creating triangles the minimum angle of the internal angles is maximized. This is implemented via a technique known as edge swapping (Figure 2.8) in which an edge is deemed illegal if the minimum of the angles in the current possible layout is less than that of the minimum of the angles in the other possible layout.

Figure 2.8: Edge Swapping

A simple algorithm for generating a DT is considered as a demonstration:

Given a set of points $P$, initialise the triangulation by choosing three points that form a bounding triangle of $P$. Select any point $p$ from the remaining points in $P$. Determine the triangle that contains $p$. Divide this triangle into three new triangles using $p$ as a vertex. Swap any edges until the triangulation is legal. Repeat until all points of $P$ at included.

Delaunay Triangulation may be useful for analysis on small structures such as nuclei to analyse patterns and crowdness.
2.6.2 Voronoi Tesselations (VT)

A Voronoi Tesselation divides a set of points into segments by considering lines that are equidistant between pairs of points. In fact, by taking the dual of this graph we actually arrive at a Delaunay Triangulation. A dual graph of a graph $G$ is a graph that is formed from placing a vertex in each face of $G$ and placing edges between pairs of these vertices if their corresponding faces are neighbouring in $G$.

2.6.3 Minimum Spanning Trees (MST)

Firstly, an introduction to some basic graph theory. A connected graph is a graph in which there is always a possible path from any point to any other point. An undirected graph is a graph in which we have no concept of orientation in terms of edges. A cycle in a graph is a sequence of vertices starting and ending at the same vertex such that if any two vertices are consecutive in the sequence, they are adjacent (connected by an edge) in the graph. A tree is a connected undirected graph with no cycles. A spanning tree of a graph $G = (V,E)$ is any subgraph of $G$ that is a tree and contains all elements of $V$.

In a graph, weights can be assigned to each edge $E$ that represent the length of $E$. From this, a minimum spanning tree MS can be defined as a spanning tree that minimizes the total weight $w$ of edges in MS such that $w$ is less than or equal to the total weights for every other possible spanning tree.

2.6.4 Conclusions

Doyle et al. [9] used voronoi tessellations to describe the spatial arrangement of nuclei in histological tissue samples of prostate cancer in order to train a classifier to distinguish between different grades. They extracted the following features: the average area of polygons; the disorder of the area; the average roundness factor of each polygon. Using the AdaBoost algorithm [12] they assigned a higher weight to voronoi features in the machine learning process to achieve an accuracy of 89.7% for distinguishing between grade 4 and stroma.

They also used delaunay triangulations derived from the previously formed voronoi tessellations to connect nuclear centroids and calculated the average length of the edges in the resulting mesh. They achieved a classification accuracy of 76.9% for distinguishing grade 3 and grade 4 of prostate cancer when giving a higher weighting to delaunay features.

They also implemented minimum spanning trees on the nuclei and extracted the average length of all edges and computed the edge length disorder. By giving the highest weighting to the average edge length for MST, they achieved 92.8% accuracy for distinguishing between grade 3 and stroma.

Keenan et al. [21] used delaunay triangulations on the arrangement of nuclei in epithelium tissue to train a machine vision system to automatically grade new instances for the level of cervical intraepithelial neoplasia. They used the mesh to derive 18 features such as: mean triangle area; mean triangle edge length; mean nuclear area.

Landini and Othman [25] compartmented the epithelial tissue into exclusive areas associated with each nucleus and then characterised these by constructing graph networks which
included delaunay triangulations and minimum spanning trees. They extracted the following features from both: the number of edges; the total length of the graphs; the mean edge length; the standard deviation; skewness; kurtosis of the distribution of edge lengths (measure of the peakedness). They extracted these features along with several others from training data labelled with normal, pre-malignant and malignant, in order to train a linear model to distinguish between the three. Their highest accuracy for distinguishing between normal and malignant tissue sample was 68% and their best neighbourhood discrimination rate was 75% for normal versus carcinoma.

In conclusion, it is possible to extract several structural features from these diagrams such as roundness factor, number of sides, area of polygons, edge length, distance with neighbouring cells and so on. It is also a common theme to apply these using nuclei, which makes sense given the background research in section 2.1 which discussed the effects of BO on nucleic structure. Therefore, as part of an extension to this project, structural analysis on nuclei will be examined. Furthermore, if certain tissue texture can be grouped up and identified as dysplastic it may be worth investigating the spatial relationship between these groups, such as distance from the epithelium layer, to predict how far it has progressed. As the different methods provide different aspects that can be analysed and there is a close relation between delaunay and voronoi diagrams anyway, as shown by the theory and the work by Doyle et al., this project will not choose a specific one but apply them as is appropriate.

### 2.7 Machine Learning Aspects

As discussed in Section 1.1, when a pathologist is analysing a tissue sample it is not just a matter of counting some quantity to come to some conclusion, but requires drawing upon years of experience and the learning of unquantifiable indicators. Any good computer related solution to this problem will require the use of machine learning to emulate pathologist-like diagnosis. Some basic theory of machine learning is introduced here, as well as specific methods.

#### 2.7.1 Supervised Learning

Supervised learning is a type of machine learning in which we have a set of training data (ie vectors of features) that are examples of our input data, in which the target class is known for each of them. We train a machine on this training data with the intention that it learns patterns that correspond to the desired target classes, and is then able to apply this learnt model to some new input data in order to classify it. This is predictably known as a classification problem [5].

**Support Vector Machines (SVM)**

The basic idea of an SVM is to form a linear model of the form $y(x) = w \cdot x + b$ (this is a linear kernel) such that new data points are classified according to the sign of $y(x)$ [5]. The training data consists of $N$ input vectors each with corresponding class $t_n \in \{-1, 1\}$. Figure 2.9 shows a support vector (the black line) between two classes of data. The aim is to position the line such that the margin (width of the yellow area) is as large as possible. The margin is bounded
by the first point it touches (no points lie within the margin) and is parallel to and centred on the black line.

![Figure 2.9: Example of a simple case of a linear support vector machine](image)

This is a simple case, where the data points are clearly linearly separable. In the cases where it is not linearly separable, it is possible to use non-linear kernels such as a Gaussian Kernel [40]. This also shows a simple case of two classes, but in practice it is often desirable to distinguish between multiple classes. Bishop [5] describes the one-versus-the-rest approach for $K$ classes in which $K$ separate SVM’s are trained and the $k^{th}$ model $y_k(x)$ is trained on training data in which data from class $C_k$ is marked as 1 and everything else is marked as -1. They also describe the one-versus-one approach in which $K(K-1)/2$ 2-class SVMs are trained on all possible pairs of classes, and a new un-seen class is put through each SVM and classified as the one with the highest number of votes. The literature also describes the pros and cons of both of these.

SVMs are a popular choice of supervised learning in medical imaging. They were used for grading benign and cancerous cells which 89.5% accuracy in [9]. They were used alongside colour co-occurence matrices in [36] to distinguish between normal, polyp and tumour images to facilitate discrimination between cancerous and non-cancerous slides, with promising results. Yinhai et al. used a combination of the classic GLCM features in describing textures along with SVM’s in order to segment squamous epithelium from cervical histological virtual slides to 92.1% accuracy.

**Decision Trees**

A decision tree is a tree where each node represents a single rule based on a single attribute and branches to another node depending on how a particular instance responds to that rule i.e., at a particular node, ”if $x1 < 0.2$, go to left node, else go to right node”. A decision tree learns rules from some training data, where each observation in the training data has a set of attributes and a corresponding class, the predictor class. The tree will attempt to learn what maps these attributes to the known class in order to classify a new unseen instance based just on its attributes. The leaves of these trees represent a single class which is described by the rules along the path down to it, and a path stops branching when all the observations that fall under it are of a single class.

It is clear that for a given set of training data, several different trees exist as different
attributes can be chosen at each stage. The can also have different depths. A smaller tree is
certainly more user friendly to read and is more explanatory, as it distinguishes between classes
at a higher level using less information, and is actually also like to have a higher accuracy in
predicting an unseen observations class [33]. This introduces the idea that we should identify
the attributes that best classify the training data, i.e. give the most information, and use these
high in the tree. Statistical equations such as information gain exist that create a numeric
measure of the amount of information in a given attribute at a given level in the tree [2].

Decision trees are advantageous for several reasons. They can be used for both categorical
(i.e. a variable temperature with possible values cold, warm or hot) and numerical data, and
typically can easily be interpreted by inspection (although this is not always the case). Furthermore,
sometimes an attribute might not be used which suggests that a decision tree can still be
generated when some observations don’t have values for certain attributes.

Decision trees also have disadvantages. They can be prone to overfitting which is when they
produce complex models resulting from trying to describe the training data too closely and end
up not generalising well to new unseen instances. This occurs because the training data can
contain noise/random error, thus making the data not a completely accurate description of the
general cases of the classes. This can be avoided by pruning, a process in which sections
of the tree are cut off if they provide not much influence towards classifying. Furthermore,
decision trees can be sensitive to differences in the training data: a small variation can cause
a completely different tree to be generated. Not only does this question the suitability of a
particular tree trained when the data is in a certain state, as essentially the same data with a
small variance may be better, but requires a completely new tree to be made as we evolve the
data we are training on.

Random Forests (RF)

Random Forests are described as a machine learning ensemble method, meaning they use mul-
tiple models in attempts to attain improved predictive performance. In general, they act in
a similar manner to decision trees but combine them with two new ideas which will now be
introduced.

In the previous section, a key problem with decision tree’s, overfitting, was discussed, and
so was a DT’s sensitivity to changes in the training data. A technique known as bootstrap
aggregating, or bagging, was proposed by [6] which he argued increased accuracy of classification
by counteracting this sensitivity and could decrease over fitting to noise/random error in the
training data. Given a training set \( D \), \( m \) new training sets \( D_i \) are formed by sampling from \( D \)
uniformly (equal chance for each observation to be chosen) with replacement (an observation
can be chosen multiple times). Each sample \( D_i \) is known as a bootstrap sample, and \( m \) models
are trained on these bootstrap samples, one for each, creating \( m \) decision trees, although it is
important to note that bagging can be applied to different methods as well.

Bagging is combined with random feature selection to form the general algorithm for creating
Random Forests. If there are \( V \) input variables in each observation, a number \( c << V \) is chosen,
then at each node of a decision tree \( c \) variables are selected at random and the best split is chosen
based on these nodes. The value of \( c \) is kept constant throughout forest growing. Furthermore,
individual trees are grown to their fullest extent and are not pruned.

Once all trees have been grown, a new unseen instance is passed through each of the \( m \) models, and each model votes for an output class for the instance. Votes are accumulated and the class receiving the most votes is what the new instance is classified as.

### 2.7.2 Unsupervised Learning

Unsupervised learning contrasts from supervised learning in the sense that the machine still receives inputs but does not use training data with labelled target classes or receive rewards for produced actions as is the case with machine learning related to game theory [14]. Unsupervised learning will find structure within the data and use this to come to some conclusion.

**K-means**

Suppose we have a dataset of \( N \) vectors all of some length \( D \) and we wish to cluster these into \( K \) clusters; k-means clustering gives us the ability to do this. Clusters are defined by cluster centroids and justified through minimizing some distortion measure such as minimizing the total distance from all the points to their designated centroid.

Bishop [5] describes the two phases of the k-means algorithm considering an objective function value labelled \( J \) that equals the sum of all the distances between each data point and its cluster centroid. In the first phase we simply minimize \( J \) with respect to assignment of points to a single centroid without changing the location of those centroids, so effectively just picking the centroid each point is closest too. In the second phase we minimize \( J \) again but with respect to each centroid location whilst keeping the assignments fixed, so effectively just finding the mean of the data points in each cluster. These phases are repeated until convergence.

There are several aspects/problems with k-means that have to be considered. It is possible that the algorithm forms an empty cluster [28], in which case we can take one of three actions: throw an error; remove the empty cluster; create a new cluster containing a single point that is currently the furthest point from its centroid. The number of clusters is an input parameter to the algorithm, but it may not be known what a suitable amount of clusters is. In the case above, a reasonable assumption could be made on what classes would be contained in the data but the number of classes isn’t always known. Therefore, it is important to run several test runs and analyse the results in order to find a good number of clusters to use for classifying the set.

### 2.7.3 Conclusion

In order to enable pathologist like thinking, there is a need to be able to learn patterns in the data and how these map to known classes for that data. SVM’s and decision tree’s are useful tools in doing this. Although SVM’s have the power to model non-linearities in data by using complex mathematical functions as their kernels and can be adapted to apply to multiple classes, they are what is commonly described as a ‘black box’, meaning that the reasoning behind their decisions isn’t apparently clear which can be disadvantageous in the diagnosis of medical issues as the medical personnel would want to clearly explain their conclusions and
decisions. Although this project is about creating a methodology to support a pathologist rather than replacing their diagnostic input, it would still be beneficial to relay justifications to them as it could help them realise new observations that help grade dysplasia as well as support their diagnostic explanations rather than just conclusions. This project will therefore use decision trees rather than SVM’s. The decision trees gives you the rules it has learnt thus showing how the software came to a particular decision, facilitating the previous comments, as well as potentially being useful for evaluation and debugging purposes. Adam [2] conducted experiments and concluded that SVM’s failed to differentiate G1 and G3 in BO, and that RF and BDT (binary decision trees) had similar results to each other, and were better than SVMs, due to their similar nature.

K-means is a standard clustering adopted throughout many machine learning systems. In the interest of time, and due to its almost standardized use, it will be adopted in this research.

2.8 Summary

This section intends to summarise some of the key points made in the background research. We have established that the lower Oesophageal tissue responds in increasing severity to acidic stimulus from the stomach. This causes it to undergo cytological changes that effect the tissues appearance and structure. These changes can be better visualised by exploiting the fact that the cytological changes cause differing responses to staining methods. These differing responses are reflected in the pixel values of the resulting image when the tissue is digitized. From this, varying computer vision techniques can be deployed. These derive features, and can be applied to images of different grades in order to derive a feature set. Machine learning algorithms can then be used to find patterns within these sets to learn what is representative of the original grades. This knowledge can then be applied to unseen images in a similar manner.
Chapter 3

Project Planning and System Design

3.1 Methodology

This project is very much an exploratory research based project, which already suggests a methodology in which minimal planning should be adopted. Furthermore, since this project requires no usability testing which is often required in projects that have a design aspect such as GUI development or an external client that has a specific set of requirements, there is little need to define specific milestones in a more structured sequential approach.

The methodology that has been adopted for this project is Rapid Application Development [4]. This approach gives favour to rapid prototyping whilst using minimal planning, resulting in a possibility to write software much faster and giving greater flexibility in terms of changing requirements.

Since this field of research is so fast and experimental, the full project requirements and exactly how they are designed are not realised at the very start of the project, requiring the ability to design and develop ‘on the fly’. For example, several weeks into an implementation of some classifier preliminary results may show that a particular choice of technique is not working, requiring a consolidation of time left and a possibility to change what has been implemented.

More specific details of how this methodology was adopted can be read in Section 6.10.

3.2 Schedule

As with an extensive project, planning is an important aspect to ensure any deliverables and write ups are developed and submitted on time. A schedule is often created, a Gantt Chart being a common form. This puts things into perspective and acts as a helpful reminder to how far into the project the current state of implementation is which helps evaluate whether progress is being made at an acceptable rate. Even in a project such as this one as outlined in the previous section, this will be a useful tool given the relatively short time frame of the project.

Therefore, a schedule was created early on to give an overview of when things would be done and how much time would be allocated to them. This can be seen in Appendix E.1.
3.3 Changes to the Schedule

An initial schedule is only an estimation, and as is common in a lot of projects eventually it was required to revise the schedule. This was done for several reasons. The implementation of the texture analysis part of the software was initially meant to be completed by 14th of March, but due to small details such as the overlapping of patches and the implementation of this part of the software in general an extra week was allocated. The time it took to run small simulations to check the capabilities of the implementation at this stage also took time, as lots of features have to be gathered from relatively small patches of large images which also contributed to this deadline extension, see Appendix F.

Another critical time delay also occurred. Initially it was planned to have just over four weeks to implement spatial classifying. This was intended to be split up into half nuclear spatial relations and features and half region spatial relations. However, due to technical reasons discussed in 5.3 as well as the loss of a week due to the texture classifier, it was decided there would not be enough time to investigate the region features. The remaining time was therefore spent ensuring the nuclear classifier was delivered on time, which was important as it had been learnt just how long it can take to extract features, of which the nuclear ones were identified to take even longer, see F. The early evaluation stage was also removed, and replaced with less formal smaller evaluations. This is discussed in Appendix A. All other milestones were completed on time.

3.4 Technology

There are a multitude of different programming languages and technologies available to implement the proposed deliverables, several of which are C/C++, Java, MATLAB, Python and OpenCV. They all have their own advantages and disadvantages that put their appropriateness for this project into question.

The most commonly used languages for Computer Vision projects are C/C++ and MATLAB.

C/C++ are widely adopted low-level/intermediate programming languages. There level reflects there 'closeness' to the underlying hardware allowing them to produce very fast and efficient software solutions. However, the pay-off of this is that they require a steeper learning curve in comparison to other higher level languages. The longer development time caused by this and the increased time spent debugging suggests it wouldn’t be very appropriate for the approach outlined in 3.1. However, given that the extraction of features and training of machine learning techniques will potentially take a long amount of time a lower-level language could potentially make it more effective.

In the end though, MATLAB was the language of choice for implementing the proposed deliverables. This is for several reasons:

- MATLAB is a scientific programming language and interactive environment developed by MathWorks [27] that specialises in numerical computation which makes it efficient when dealing with matrix operations. This will be particularly useful in this project as it
deals with the analysis and feature extraction of images, which are just large matrices of
numbers representing pixel values.

• MATLAB also specialises in visualization, and provides an extensive set of built-in func-
tions for this very purpose with regards to the plotting of data and generation of decision
trees which will be useful in the evaluation stage of this project.

• MATLAB, being one of the most popular languages for Computer Vision purposes, has
an abundant amount of built-in functions for various image processing goals. This in-
cludes single simple function calls for the reading and displaying of images, which is both
practically useful as well as making it easy to view images for confirmation and debugging
purposes.

• The latter two points and the high-level nature of MATLAB means similar solutions
to those that could be derived in other languages can be arrived at in a much shorter
timespan, which is important given the timespan of this project.

3.5 Previous Work

The project supervisor highlighted previous work done by Adam [2] who also explored textural
and spatial features from tissue regions, looking at three approaches. The first was to extract
textural features of sub-images along the epithelial layer of the tissue. The second approach
was to analyse textural features of whole tissue regions using GLCMs, rather than just patches
along epithelial layer. They used this information to form new representations of the images,
to which further analysis was done to obtain features for a binary decision tree classifier. The
third approach was to look at the spatial arrangement of tissue types.

For the second approach they achieved kappa values, a measure of agreement (see Sec-
tion 6.2), of 0.75, 0.5 and 0.63 for G1, G3 and G5 respectively- which out performed current
pathologist agreement, discussed in Section 6.1. Since their adoption of GLCMs and decision
trees were in-line with the background research that has been carried out, it was decided to
adopt their optimal experiments within this project, discussed at the appropriate stages of the
implementation chapters.

3.6 Grading Scale

As discussed in section 2.1, annotated regions are graded with a six level grading scale. Adam
et al. [3] describes a review of the literature, and their own experiments, in which tissue samples
were classified into 5 grades, 3 groups of grades and 2 groups of grades. The review found that
classifying into 2 groups of grades, best AP of 81.47, yielded better results than classifying into
3 groups of grades, best AP of 47.41, which out-performed classifying into the original 5 grade-
scale, best AP of 31.13. Furthermore, higher agreement between the classifier and ground truth
values is achieved when the classes are grouped more. This predictably occurs as the distinction
between 5 individual groups will be less defined and more clues will have to be used to grade a
new instance. The use of two groups, although effective, is too general and doesn’t give enough
information. Therefore a balance between accuracy and information is achieved by adopting the 3 group scale. This grouping for the 6-grade scale is shown in table

<table>
<thead>
<tr>
<th>Vienna Classification</th>
<th>3-grouped Classification</th>
<th>2-grouped Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1 - Barrett's Oesophagus</td>
<td>G1: G1-G2</td>
<td>No Dysplasia</td>
</tr>
<tr>
<td>G2 - Probably Negative for Dysplasia</td>
<td>G3: G3-G4</td>
<td>Dysplasia</td>
</tr>
<tr>
<td>G3 - Probably Positive for Dysplasia</td>
<td>G5: G5-G6</td>
<td></td>
</tr>
<tr>
<td>G4 - Low Grade Dysplasia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G5 - High Grade Dysplasia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G6 - Intramucosal Carcinoma</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Grading Scales

3.7 Ground-truth Data and Data Filtration

The data used for this project was provided by the Leeds Institute for Molecular Medicine. There are two sets of data that are available for this project. The first is an excel sheet consisting of the details of whole slides and the diagnostic classification given from a set of experts. The second is in excel file consisting of details about regions highlighted by pathologists, which we refer to as annotated regions, as they were in the process of diagnosing a whole virtual slide. The goal of this project is aimed at classifying given regions of tissue based on their textural or nuclear/spatial features, and time is not allocated for developing techniques to locate regions of interest from the whole virtual slide. Therefore the second excel file is the one this project uses.

This file consists of details about a set of 640 unique annotated regions highlighted and graded by two expert and two trainee pathologists. To ensure the reliability of the ground-truth classifications which will be used to train and evaluate the classifiers the trainee data is discarded, leaving 474 annotated regions. The information provided for each of these annotated regions includes its ID, slide number, the pathologist who highlighted the region (Expert_B, Expert_E), his/her grading for that annotation (according to the six-grade scale, chosen one), the zoom level used, his/her comment about the annotation, various fields describing the coordinates of the four corners of the rectangular annotation, information about the size of the region in pixels and a url address that can be used to access an image of that particular annotation.

Before this data is separated into training and test sets the data needs to be filtered for varying reasons highlighted in the technical report that accompanied the data, written by Adam [2] whilst doing her research. The reasons that were highlighted included misleading annotations, images that were not H&E staining, annotations that were 0 pixels in size, lack of quality on the tissue slide itself and blurred images. If these were not removed they could introduce noise and have negative effects on the classification results.

Images below 250*250 pixels in size were also removed, as these were considered too small to contain useful information. Images larger than 5000*5000 pixels in size were also removed.
In this case this was done because it was deemed they contained too much of the tissue and could give misleading information as it increases the chance of including extra tissue that was not characteristic of the grade the pathologist has annotated it as i.e. a pathologist may have highlighted multiple regions characteristic of G5 in a single rectangular region, which may contain bits of lower grades in-between.

When looking through the data, it was found that the pathologists grading would occasionally conflict with their comments: they may grade a region as high grade but comment on it being low grade. The consultant pathologist told us that this was due to the grade being the pathologists overall impression: he may have annotated four regions from the same slide, commented that two were low grade and two high grade, then given them all a grading of high grade in the grade field, to account for the worst-case scenario. Rather than manually going through and manually changing the hand written comments into a fixed labelling system or just using the grading field we inspected the data and removed any fields that contained conflicting information in the grading and comment section.

This left us with 420 annotated images. After reducing the grades to the three grade scale, the data was split into training and test sets for each grade like so:

<table>
<thead>
<tr>
<th></th>
<th>G1</th>
<th>G3</th>
<th>G5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>232</td>
<td>116</td>
<td>164</td>
</tr>
<tr>
<td>Test</td>
<td>60</td>
<td>28</td>
<td>40</td>
</tr>
</tbody>
</table>

In each set, there was an equal balance of the grade and not that grade. For example, the train set for G1 contained 116 G1 and 116 NG1. Each not grade set contained an equal balance of entries for the other two grades. So for train set G1: 58 G3 and 58 G5.

### 3.8 System Design

Based on the project methodology, a detailed system design would not be appropriate as it would limit the ability to explore. But to bring the different aspects: computer vision, machine learning and biology together simple pipeline diagrams were developed to envision what the texture and nuclear analysis systems would act like. They follow the simple design of extract features, train, classify unseen instances based on the same features. 

As discussed in Section 2.8, the tissue exhibits behaviour which is reflected in the images which allows computer vision techniques to be applied to extract features. For the nuclear analysis, we need to process the images (i.e. locate nuclear centroids) so we have the means to apply feature extraction.

For the texture analysis, an additional stage is required in which we use all images to describe texture types which can then be used to perform further analysis on the types of textures within images. This is based on the new image representations mentioned in the previous work, Section 3.5. Features are extracted from this analysis.

In both cases, patterns can be learnt in the data to attempt to learn what corresponds to a particular grade. All this is explained in more detail in the subsequent chapters.
Chapter 4

Implementation - Texture Analysis

This chapter discusses the stages that had to be taken in order to extract the textural features. We look at reading in the images, splitting them into patches, performing texture analysis, creating new image representations and doing a higher-level texture analysis.

4.1 Colour Normalisation

As discussed in Section 2.4, colour normalisation should be investigated due to the variable staining that can occur. Previous work done by Adam [2] evaluated similar methodologies to those that were explored in [26] and concluded that RVM (relevance vector machine) bimodal colour normalisation worked best. This method was investigated. This method suffers from situations where the staining is very dark or very weak. For example, a classification model is used to determine reference staining colours for colour deconvolution, as part of the normalisation process, and biased estimates (caused by the strong staining) can affect normalisation results.

In this instance, this method and others were attempted to be implemented on the images stored on the image server but no single classification model was found to be suitable, often giving black images in response. Multiple rvms were used, and example of good staining were obtained by request from the consultant pathologist, but to no avail. Since a considerable amount of time had already been spent, and given the time-scale of the project, colour normalisation was not adopted and instead images are converted to the grey scale required for the GLCMs in order to minimize variations in colour intensities.

4.2 Reading in the Images

The full virtual slides are stored on an image server at University of Leeds digital Pathology Project. Due to the fact these images are scanned in at a high resolution, 40x zoom, they are in the billions of pixels in size, and take up GBs in memory. Therefore it is important to be able to read in relatively smaller parts of the image, particularly those highlighted by the pathologist, rather than the whole image itself. This is done by manipulating the image server URLs using several parameters:

http://129.11.65.162/Barretts/10592.svs?012693+019130+412+382+8+100
xPos and yPos are the x and y coordinates of the top left pixel of the annotation in the image. The height and width are in pixels, which can have a maximum of 2000 each. The zoom level is the Aperio zoom level and has a correspondence with the standard convention of zoom as shown below:

<table>
<thead>
<tr>
<th>Aperio</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convention</td>
<td>40x</td>
<td>20x</td>
<td>10x</td>
<td>5x</td>
<td>2.5x</td>
<td>1.25x</td>
</tr>
</tbody>
</table>

Images can be extracted into Matlab using these URLs with a simple \texttt{imread} call. As we zoom out, the real-life distance in microns per pixel increases. This means that at different zoom levels a feature, i.e. nuclei, will be represented by a different number of pixels. Since the different regions in the dataset were extracted at different zoom levels, features that are actually the same in different images may not be detected as so. Therefore the images need to be converted to the same zoom level to be comparable. However, simply changing the zoom level changes the image that is extracted, adding more or less of the tissue depending on whether the zoom level was increased or decreased. This is because the height and width are kept the same and now correspond to different real life distances. Therefore the height and width need to be multiplied by the inverse of the scale by which the zoom level was changed i.e. changing the zoom level from 8 to 2 means the height and width need to be multiplied by 4, 1 to 16 multiplied by 1/16 etc. This is done by string splitting on delimiters and string concatenation to extract parameters, change them, then form new URLs.

However, this causes some of the images to go above 2000*2000 in size. To counteract this, Wright [45] developed an image block concatenation technique in which an image was extracted in 500*500 blocks then concatenated together. The number of blocks required in each dimension was decided by dividing the new width and height by 500 and rounding up (to ensure more rather than less of the image was retrieved). Then, using information about the required image size, the image is cropped.

### 4.3 Image Patches

It is obviously not going to be sufficient to just perform texture analysis on an annotated region as a whole as this will produce many different possible clusters for each grade due to the varying ways in which each region was highlighted, the positioning in the tissue, the amount of background, and the fact that the visual differences between the grades are not clean-cut and simple to categorize. This results in a lack of a singular texture descriptor for each grade. We also know that there are several possible visual signs pathologists are trained to look for, not just one. Therefore, as with many computer vision programs, the images will have to be split into patches in order to analyse different textures that appear within the tissue.

Rather than just split images into patches, this project adopts overlapped patches. The reason for this is because features that are good indicators for a particular grade could be lost when a patch doesn’t sufficiently contain that feature in order to recognise it (see figure 4.1(a)).

By observing diagrams like that of Figure 4.1(b), which shows three overlapped patches, it was identified that for a particular dimension:
Figure 4.1: Observing overlapped patches

\[ d = (p \times n) - ((n - 1) \times o) \]

where \( d \) is the distance in pixels in that dimension, \( p \) is the patch size in that dimension, \( n \) is the number of patches in that dimension, and \( o \) is the overlap of patches in pixels. This equation was then re-arranged to put it in terms of \( n \) which gave:

\[ n = \frac{d - o}{p - o} \]

This new equation allows us to calculate the number of patches required in order to split an image into patches of a given size and overlap. When applied to both the \( x \) and \( y \) dimensions it makes it easier to manage the for loops that loop through the image to extract each patch. However, for images that cannot be evenly split up \( n \) will be non-integer. Because of this, it was decided to round \( n \) down. This results in a potential to cut off of parts of the image, shown by the grey area of figure 4.2.

Figure 4.2: Information loss due to image patches

Initially this may seem to be discarding potentially useful information, which technically it is, but it was decided the effects would be minimal as when a pathologist highlights a particular region they are likely to click and drag round a region of interest and not spend a lot of time making sure the box lined up with completely relevant information. If there was anything important on these boundaries, they would have roughly centred a region on it or extended the region to include it more. Furthermore, it can be easily calculated that the cut-off will have a maximum width of \( p - o - 1 \) in each dimension which for a patch size of 100 is at most 89 pixels (if a minimum overlap of 10 was enforced). Given that a lot of the images are in the several of thousand of pixels in each dimension, this shouldn’t be significant. In this research we use an overlap of 20.
4.4 Cluster Co-occurrence Images (CCIs)

As has already been mentioned in this report, the diagnosis and grading of BO isn’t based on a simple quantification of some abnormal feature or lack thereof and therefore it isn’t good enough to just identify some type of cell or structure in an image. Furthermore, an abnormal feature may be present in two different gradings and the difference between the two could be based on where that abnormal feature is in relation to other features. This means a higher level of analysis needs to be carried out introducing a shift from pixel-level analysis to patch-level analysis, requiring that each image be transformed into a new representation and the final analysis carried out on that instead. CCIs are this new representation.

They are created by splitting a set of images into patches, see Section 4.3, and performing clustering on these patches based on their textural features. GLCM analysis using the Matlab function `graycomatrix` is carried out on each patch, and the four popular statistics, see Section 2.5.1, are retrieved. However, preliminary tests and research suggested that these four numbers alone would not suffice in giving a detailed enough description of each patch in order to generate natural clusters amongst them. In addition to this, due to the nature of tissue and the virtual slide creation process, the patches don’t contain tissue in a standardized orientation which implies multiple GLCM directions need to be used to achieve a degree of rotational invariance. To describe the patch in even more detail multiple directions are also used, ensuring the right level of detail in describing the texture is captured. 10 different distances (offset of 1 to 10) were considered for each direction which produced 160 features in total (4 directions $\times$ 10 distances $\times$ 4 features) for each patch, following the previous work.

In order to keep the code manageable and maintainable a separate module of code was developed to manage this feature extraction separately. This module accepts an input image, in this case a patch, the distance from 1 to use for the GLCMs, a boolean saying whether the GLCM will be symmetric or not, and a boolean saying whether we want to include correlation from the GLCM’s or not. This latter parameter is needed to address the problem that a patch may have no pixel correlation value, NaN, in which case clustering will ignore the entire feature vector. Adam [2] describes two ways of dealing with this, EXP1 and EXP2, which is to either replace NaN correlation values with 0.0001 or to remove all the correlation values from all feature vectors to enable clustering. EXP1 is adopted for grading G3 and G5 whereas EXP2 is adopted for G1.

Once a list of features vectors for all patches from all input images is obtained k-means clustering using Matlab’s `kmeans` function is carried out to cluster the patches into groups. The optimal clustering found by the previous work was five clusters, but since they used particular choices of images that included the epithelium lining and the lamina propria (the supporting tissue around it), there may be more texture types in our less discriminative dataset. A simple experiment was carried out which varied the number of clusters between 3 and 8 and plotted the total euclidean distance between the feature points and their corresponding centroid. This graph is show in Figure 4.3. Choosing too few number of clusters costs more and may fail to capture the different texture types within the data. Choosing too many clusters may cost less, but may be over-fitting. A compromise is chosen by choosing the point roughly half way between. This number is five. Since this is in-line with the previous work, confidence was taken.
in the value and it will be used in subsequent experiments.

![Figure 4.3: Cost for clusters](image)

Figure 4.3: Cost for clusters

In order to capture a single generalized set of cluster centroids that are not specific to each experiment, the training data from each experiment is combined into a single set and the patch creation, texture analysis and clustering is carried out on all of these. This is done twice, to obtain a set of centroids for both EXP1 and EXP2 as their feature vectors are of a different size. These sets of centroids are then saved and re-used in each experiment.

The centroids are then used to generate the CCIs by labelling each patch in an image with a label corresponding to the cluster it belongs to, based on its textural features. This produces a new matrix of numbers representing the image which is $numPatches \times numPatches$ in size. An image of a region of stained tissue and a corresponding CCI can be seen as an example below in Figure 4.4.

![Figure 4.4: CCI example](image)

Figure 4.4: CCI example
4.5 Cluster-coded Cooccurrence Matrix (CCM)

As discussed at the beginning of section 4.4, there needs to be a shift from pixel-level analysis to patch-level analysis in order to effectively capture the relationship between different types of textures and the correspondence of these relationships to the different grades of dysplasia. To explore these relationships GLCM analysis is carried out again but on the \(numPatches_x \times numPatches_y\) sized CCIs of each input image. This should theoretically work as a description because GLCMs analyse the co-occurrence of pixels, and in this case each patch is represented as a single 'pixel'. Furthermore, since each cluster \(C_1, ..., C_5\) is assigned a different grey-level value the typical GLCM features can be used to intuitively deduce a degree of spatial relations between texture clusters. For example, if \(C_1\) is assigned a grey-level of 51 and \(C_5\) is assigned 255, an image with a high contrast could imply there is a lot of texture type \(C_1\) next to texture type \(C_5\). Taking this a level further, the multiple directions and distances further imply similar deductions. Figure 4.5 shows that if GLCM analysis gave a moderate value for \(135^\circ\) contrast and \(90^\circ\) contrast then a high value for \(0^\circ\) contrast and \(45^\circ\) contrast it could imply that where you find \(C_1\) in the image you often find \(C_2\) above and to left and \(C_5\) above and to the right.

![Figure 4.5](image)

Due to their much smaller size and the fact that we wish to consider relationships across the entire region, the CCIs are not split into patches. Four directions are still considered for the GLCM as well as multiple offsets however four offsets (1 to 4) are used instead of 10. This was the optimal value found by [2], and most likely works the best due to the CCIs relative size, as larger offsets will count the co-occurrence of texture types far apart in the original image which probably won’t be as relevant as a patches relations with those closer to it.

To reduce the effects of noise from texture outliers, before the CCM features are extracted patches far from the existing centroids are removed. This is carried out by calculating the standard deviation in distances between centroids and all their patches based on the texture feature vectors. Any new patches more than three standard deviations away from the centroids they’re assigned to are the ones discarded. The largest connected component of a texture corresponding to that of the image background is also removed. This is done by creating a binary image where each foreground pixel represents patches belonging to the cluster of mostly blank patches, performing connected component analysis, and using this information to change the CCI.

As discussed in Section 3.5, the optimal experiments from their work are adopted here. The three experiments for extracting features to feed to the decision tree are as follows. A1 for G5: full CCM features are used (contrast, correlation, homogeneity, energy). A4 for G3: CCM features without correlation. A7 for G1: frequency of clusters in each CCI.
4.6 Testing Stage

Once a decision tree has been trained on a set of training instances using MATLAB’s ClassificationTree class based on a set of extracted features and the corresponding classification from each, the same features are extracted from each of a set of test instances and passed to the tree to see what they get classified as. Therefore, the testing stage of the texture classifier is effectively the same as has been described thus far. All that had to be done was ensure the same analysis and parameters were used from the test set of a specific train set so the same feature vectors are extracted. Again, for maintainability and manageability, a separate module was made that accepted the decision tree from the training phase, the name of the test file and an array containing the specific parameters to be used i.e. EXP1/EXP2, distance for the GLCM, exp A1/A4/A7 etc.

4.7 Code Testing

Some artificial data was generated in order to test the patch creation, texture analysis and clustering aspect. This data involved multiple images of obviously different textures. Since it was known how many texture types were in the images, the appropriate amount of clusters could be chosen and the outputted CCI was inspected. No concerns were raised.
Chapter 5

Implementation - Spatial Analysis

5.1 Nuclear Segmentation

Since this is such a large area in itself, it was advised not to try to reimplement some new method of nuclei detection but instead use an existing in-house approach. This approach is described in the subsequent section.

5.1.1 In-house Approach

The best approach currently available at the Leeds University Vision Group as recommended by a member is one developed by Chomphuwiset et al.\cite{7} which uses a Hough transform based technique that uses Hough voting to form an accumulation array from which seeds points for elliptical fitting are chosen based on local maxima within the array. These local maxima are formed by centering circles of a defined radius on the edge responses given from the Sobel edge detection on a de-convoluted image showing the hematoxylin stain contribution. Peaks of intersections of these circles are taken as further seed points. There are three parameters that determine the output of this approach: a size threshold which is the radius used in voting; an edge threshold (only edges of a certain strength (based on their gradient) are retained; a smoothing constraint used to form the edges.

However, a problem with this approach was discovered. Although the program produced promising results when tested on a selection of images from the different grades, it fell-down when it came to identifying nuclei that were clumped together. This is show in Figure 5.1, where clumped nuclei are completely ignored even though it is fairly visible that there are nuclei present.

One way to get around this was to attempt to optimise the three thresholds for the set of annotated regions or on a per-image basis, but it was advised by the individual who provided the code that the thresholds provided were those used by the original research and it would be best to keep them how they are. Furthermore, the technological limitations and lack of a C compiler of the machine that was available meant it would be difficult to make changes to the code in order to derive information for which a basis for optimisation could be created. It might have been possible to study the code and produce separate code that would act as a pre-processing
Figure 5.1: In-house nuclear detection on non-clumped nuclei and failure on clumped nuclei

unit, returning optimal threshold values for each image. Given the ratio of complexity of this with the potential that it might not yield any improved results (even potentially yielding worse ones) meant a more intuitive and time-effective approach was chosen. This is described in the next section. It was decided to only adopt this new approach for extracting the average edge length and average triangle area features, and use the previous method for the other features. The justification for this was that for the other features, the information would still be captured by the detection of the other nuclei, which would in effect be acting as a sample. However, for the spatial features the loss of closely bunched nuclei would significantly alter the results as the whole reason for detecting these features relates to this quality being more present in higher grades.

5.1.2 An alternative approach

The consultant pathologist highlighted previous work done by James Swainston [41], who used delaunay triangulations on nuclei centroids to help automatically grade dysplasia in Barrett’s Oesophagus. The code for this was unavailable at the time of this project so had to be reimplemented from scratch, though the key concepts were extracted from Swainston’s work.

The initial stage of this approach involves converting an input image from its RGB colour space into its HSV colour space and performing thresholding on the hue and saturation channels. The optimal thresholds were found by plotting pixels’ hue and saturation values on a graph then identifying a bounding region around the pixels corresponding to those of nuclei. This simple approach effectively produces a binary mask that highlights colours in the image corresponding to that of the nuclei, Figures 5.2(a) & 5.2(b).

However, before this is carried out the images are pre-processed by applying Guassian smoothing, determining new pixels values by averaging the values within a neighbourhood [38], using a 7x7 Kernel with a standard deviation of 9. This results in the nuclei becoming more defined, as is evident in comparing Figures 5.2(b) and 5.2(c), the latter of which shows the HSV thresholding on a pre-processed image. The smoothing connects individual nuclei internally when they would otherwise be disconnected.

As is also evident in Figure 5.2(c), nuclei that are close together appear as a single connected component, meaning they will be recognised as a single nuclei. Therefore, a way of separating
these nuclei has to be chosen. The way Swainston did this was to first form a graph that
represents each connected component, then loop through each pixel in the connected component
and consider its 4-connected neighbourhood. If a neighbour was in the connected component
an edge was formed between the two. Initially this project used a \( m \times m \) matrix to store
the connectivities of \( m \)-pixel components but this became infeasible for larger components as
MATLAB only uses a limited amount of memory and will return an 'out of memory' error when
this is exceeded. Because of this, and since for each graph node there will only be a maximum
of 4 outgoing edges, sparse matrix storage is both required and used.

Now that a graph representation has been generated, graph partitioning is carried out to
split this graph into smaller ones to represent each nuclei. To do this, a graph partitioning
software package called Metis is used [1]. Metis uses several mutli-level partitioning algorithms
to separate an input graph into a user-specified number of partitions.

Traditionally Metis accepts a text file of a particular format that contains all the relevant
graph information such as nodes and connectivity, and outputs a text file containing the nodes
and the new partition they belong to. Since the project at this point was already segregated
between two different machines, the output and processing of text files would be another step
that would slow the development of the system. Therefore a short time was spent considering
alternatives, ideally a way of using metis directly within MATLAB without the exchange of
external text files. Metismex is a mex interface for allowing this to happen [15].

Before the connected components are divided up, regions smaller than 100 pixels in size are
discarded. This can be observed in Figure 5.2 where some small artefacts have been removed.
This is in-line with analysis done by Swainston who determined that these were often found to
not be nuclei, and which also suggested the average nuclei size was 421 pixels and since they
were looking at tissue samples of the same medical condition, these will be adopted. Metis
requires the number of desired partitions to be specified, so this was calculated by dividing the
size of a connected component by 421 and rounding up .

The output of the metismex program also specifies what partition each pixel belongs to,
but this is relative to a particular connected component i.e. two different components will both
contain label '1', even though they are not connected which each other. Therefore a relabelling
procedure is adopted that loops through all the pixels and uses information about the total number of connected components along with the metis partition label in order to come up with a globally unique new label with respect to individual connected components, as shown in Swainston’s work. The output of new approach thus far along with relabelling is shown in Figure 5.3(b).

5.1.3 Finding Nuclei Centroids

The theoretical nuclei centroids are found by simply finding the mean of the x and y coordinates of each pixel within a particular connected component.

Figure 5.3: Connected components after graph partitioning and relabelling

By plotting these points on the original image, the output of the chosen alternative approach can be seen in Figure 5.4.

Figure 5.4: Output of the alternative nuclear detection approach
5.2 Features

Now that the nuclei have been detected and their centroids and subset of boundaries have been found between the two methods, the project can explore some of the features that were highlighted in the literature research to in order to begin classifying different grades of dysplasia based on spatial and nuclear analysis. The following five features will be explored: Nuclear Darkness; Nuclear Size; Nuclear Irregularity; Average Edge Length; Average Triangle Area. In each sub-section a brief reminder of why a particular feature is significant is given along with the details of how this information was captured.

5.2.1 Nuclear Darkness

Nuclei are stained a dark blue because of the type of DNA inside them. As the cells become more malignant, they divide more often and produce more of this DNA which causes them to become darker, hyperchromatism. Therefore one of the features this projects looks at is nuclei darkness (ND). To extract ND information the coordinates of the boundary points of a nuclei are used. These form a polygon in which every pixel inside this polygon is inspected to find its value. To limit the area of search from every pixel in the image, a bounding box around these points is formed and only the pixels inside this are considered for inspection.

Using a small set of ten images for each grade, a small experiment was carried out to see whether distinguishing patterns occurred in the distribution of nuclear darkness values. The distributions for G5 and G1 showed no outstanding characteristics and formed definable peaks, but the distribution for G3 had a less defined peak, Figure 5.5(a).

![Nuclear Darkness Distributions for Different Set Sizes](image)

Figure 5.5: Nuclear darkness distributions for different set sizes

This behaviour initially suggested that two peaks may be forming. This reflects what could be deduced theoretically: that G3 contains some lighter nuclei and some darker nuclei due to it sitting in between the transition from no dyplasia to high grade dysplasia, and that some but not all of the nuclei have become malignant. Due to this, it may be appropriate to analyse particular bins of the distribution histogram rather than just taking a mean to produce several features. To identify potential bins of interest, the same distribution was formed but for a larger
set of images, 50 in this case. This however yielded a more defined peak, Figure 5.5(b), and therefore suggests the behaviour shown before is not generalisable.

5.2.2 Nuclear Size

As discussed in Section 2.1, the nuclei at a high level description exhibit enlargement as high grades of dysplasia are occur.

Using the coordinates of points along nuclei boundaries returned by the original nuclear detection method, Section 5.1.1, a polygon could be formed using MATLAB’s built-in function `polyarea` which takes two vectors of ordered \( x \) and \( y \) coordinates respectively and outputs the area. After verifying by inspection of the provided code and from test plots of the points that the outputted points of the nuclear detection program were ordered, the area could be easily calculated.

5.2.3 Nuclear Irregularity

As cells become more malignant they divide more often, and produce more of the characteristic DNA inside of their nuclei— which is initially organised into neat packets. The more often they divide, the less control they have over the organisation and shape of their nuclei which subsequently become more irregular. In order to measure the irregularity of nuclei a novel approach was devised using the resources available at this late stage in the project. Using the output of the first nuclei detection algorithm a list of potential radii for each nuclei was calculated by taking the distance between the centroid and each point along the boundary of said nuclei. The standard deviation was taken for this set of potential radii to give a number representing the spread. Before this was done, the distances were scaled to be between 0 and 1 as nuclei of different sizes but the same shape should be deemed to have the same irregularity, Figure 5.6. From this, the mean nuclei irregularity can be calculated.

![Figure 5.6: Nuclei with different sizes but the same shape](image)

5.2.4 Average Edge Length

The average edge length (AEL) of all the triangle edge lengths in the derived delaunay triangulation can be derived. The distance between pairs of centroids that have an edge between them in the derived triangulation for that set of points is considered as a feature in attempts to capture information regarding the crowdness of nuclei as discussed in Section 2.1.

\[
d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}
\]

It is suspected that G5 will have more shorter edges than G1, as the nuclei are closer, and that G3 will be somewhere in between as the transition from no dysplasia to high grade dysplasia is only partially complete.
5.2.5 Average Triangle Area

Once the centroid locations have been captured using the method discussed in section 5.1.2, and a delaunay triangulation has been derived from them, the average triangle area (ATA) can be calculated. This feature hopes to capture the progressive crowdedness action nuclei typically take in higher grades of dyplasia, Section 2.1. Given the irregular shape of the resulting triangles and the fact they are rarely right angled triangles or of a defined class of triangle, a formula based purely on the lengths of the edges is used. This is Heron’s formula, and for a triangle with sides of lengths a, b and c is defined as follows:

\[ s = \frac{a+b+c}{2} \]

\[ \text{Area} = \sqrt{s(s-a)(s-b)(s-c)} \]

Similarly to the last feature, it is suspected G5 will have more smaller triangles than the other two grades, and G1 will have larger ones.

5.3 Technical Issues

The in-house nuclear detector was written in C++. To be able to use this in Matlab, a MEX-File, an interface between Matlab and other subroutines, was also given. This however was not compatible on Linux machines, and given the code had a lot of dependencies it was decided it wouldn’t be viable to write a new one. Therefore, after a day and a half delay, the nuclear analysis aspect of the project was carried out on Windows machines available at the University of Leeds. For various reasons, such as these machines being in a more public location, they were accessed via the remote software accessing application Desktop Anywhere. These were slower than the linux machines, and could not be reliably left running over night as they would disconnect. On one instance this caused a file holding the bottleneck nuclear features to become corrupt, which delayed the project by a day so we could re-extract the features.

5.4 Tesing Stage

Similarly to the testing stage for textural features, the nuclear features are simply extracted from the set of training images and used to create another ClassificationTree decision tree to train unseen instances.

5.5 Code Testing

As with the texture analysis, preliminary testing to check that the code was working were carried out. This involved creating images of nuclei exhibiting different characteristics using different shades of colours that were compatible with the original nuclear detection algorithm. Tests naturally followed that checked whether the extracted features were as expected. The new second nuclear detection algorithm was easily testable by observation.
Chapter 6

Evaluation

This section outlines the approach that was taken in order to evaluate the success of the project. We will evaluate the performance of both approaches individually on the three sets of training and test instances outlined in Section 3.7. Performance is evaluated by comparing the output classification of unseen instances with their ground-truth classification and deriving objective quantitative statistics. Performance is also evaluated by comparing to the current agreement rates given by pathologists, and contrasting some of the tree structures and results patterns to previous work. Since one of the reasons decision trees were chosen was due to their readability, their nodal structure based on the fact that features holding more information appear near the top will be discussed to see if it lines up with the theory. Given the potential sensitivity of decision trees, any results obtained may not be generalisable to different data. Therefore this will be investigated by looking at using different training and test sets. In addition to this, we will perform a brief evaluation on the performance of the algorithm in terms of run-time in Appendix F, and an evaluation of our chosen project methodology.

It is important to note that some of the values are not directly comparable to the previous work [2], as they used cross-validation, Section 6.5, in order to find optimum decision tree models- which was validated with a set of 24 unseen images. In our case, since we used a larger amount of data, it was decided to stick with choosing a single training and test set.

6.1 Giving context to statistics

It is not immediately obvious how good a percentage/statistic is. In the context of University degree classifications, 75% is considered to be in the top tier and is looked upon favourably. In the context of recognising the presence of high-grade cancer the same percentage may be concerning, as 25% of the patients may not get urgently needed treatment. Therefore it is not good enough to just compare to previous work, and some way of measuring the classifier performance against the pathologists is required. One of the key motivators for this project was the low scoring inter-observer agreement between pathologists, due to their differing years of experience. Therefore if a measure of agreement between the produced classifiers and the ground-truth data and a measure of agreement between sets of pathologists can be calculated, they can be compared with each other to see if there has been any improvement in the current
level of agreement. A quantitative measure of the level of agreement between two observers can be derived in the form of a Kappa Value, Section 6.2.

However, there is a problem with this: agreement between the pathologists on the given dataset cannot be derived. This is because each annotated region is unique, as highlighted by the pathologist who graded it, and there is no union of sets of regions graded by both experts. Agreement between experts for the whole virtual slides containing the different annotated regions could be used, but it was felt that this wouldn’t give a fair basis for comparison as pathologists would be more likely to agree in this situation, i.e. for a given virtual slide containing three highlighted regions one pathologist may say that two of the regions are G1 and one is G5, and the other pathologist may say that two of the regions are G3 and one is G5. Overall they may both pick the worse case scenario and classify the whole slide as G5, showing disagreement on the region-level but not at the whole slide level. This line of thinking was mentioned in Section 3.7.

It was decided to use previous research into the inter-observer agreement between pathologists to obtain agreement values to compare to. Since a full formal analysis of the different studies that had been carried out was out of scope of this project, it was concluded to use kappa values and percentages reported in [2] with reference to [22]. The values chosen for the current agreement levels were a kappa value of 0.33 for G1, an accuracy percentage of 0.72 for G3 and a kappa value of 0.6 for G5.

6.2 Confusion Matrix Statistics

A confusion matrix is an \( N \times N \) matrix, where \( N \) is the number of possible classes, where each row corresponds to the ground-truth classification and each column corresponds to the predicted class. In the case of the experiment for tissue samples corresponding to that of G1, the confusion matrix would take the form of that below:

<table>
<thead>
<tr>
<th></th>
<th>G1</th>
<th>NG1</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>TP</td>
<td>FN</td>
<td>( m_1 )</td>
</tr>
<tr>
<td>NG1</td>
<td>FP</td>
<td>TN</td>
<td>( m_0 )</td>
</tr>
<tr>
<td>Total</td>
<td>( n_1 )</td>
<td>( n_0 )</td>
<td>( t )</td>
</tr>
</tbody>
</table>

True positives (TP) are images correctly classified as G1, false negatives (FN) are images pathologists graded as G1 but have been incorrectly classified as NG1, and so on.

The overall Accuracy Percentage (AP), the number of correctly classified instances, is simply calculated equation 6.1

\[
AP = \frac{TP + TN}{TP + FN + FP + TN} \tag{6.1}
\]

The Precision, the amount of positive predictions that are correct, is calculated by equation 6.2. This will be important to look at as for higher grade classifiers, a low precision will mean that negative cases are actually being classified as positive which would be distressing to the patient. Equally, for a lower grade classifier it would mean high grade cases would be going unnoticed.
\[
\text{Precision} = \frac{TP}{TP + FP}
\] 

(6.2)

The \textit{Recall}, the amount of actually positive instances that were predicted as positive, is calculated by equation 6.3. This will be important to look at for the higher grade decision trees, particularly G5, as in those cases it is especially important to be able to correctly identify positive cases as they would more imminently need to be treated.

\[
\text{Recall} = \frac{TP}{TP + FN}
\] 

(6.3)

The \textit{Kappa Value} (KV) shows the amount of inter-observer agreement. In this case it would calculate how much the classifier agrees with the pathologists. It is calculated by equation

\[
KV = \frac{AP - p_e}{1 - p_e}
\] 

(6.4)

where:

\[
p_e = \left( \frac{n_1}{n} \times \frac{m_1}{m} \right) + \left( \frac{n_0}{n} \times \frac{m_0}{m} \right)
\]

Viera and Garrett [43] a kappa value of <0 is less than chance agreement, 0.01-0.20 is slight agreement, 0.21-0.40 is fair agreement, 0.41-0.60 is moderate agreement, 0.61-0.80 is substantial agreement and 0.81-0.99 is almost perfect agreement. Its possible for agreements to be negative, if the observers completely disagree.

### 6.3 Texture Analysis Evaluation

<table>
<thead>
<tr>
<th>G1</th>
<th>NG1</th>
<th>G3</th>
<th>NG3</th>
<th>G5</th>
<th>NG5</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>22</td>
<td>8</td>
<td>G3</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>NG1</td>
<td>8</td>
<td>22</td>
<td>NG3</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G5</td>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NG5</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 6.1: Confusion Matrices for Tree-G1Tex, Tree-G3Tex and Tree-G5Tex

<table>
<thead>
<tr>
<th></th>
<th>G1</th>
<th>G3</th>
<th>G5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy Percentage</td>
<td>0.73</td>
<td>0.57</td>
<td>0.73</td>
</tr>
<tr>
<td>Precision</td>
<td>0.73</td>
<td>0.6</td>
<td>0.76</td>
</tr>
<tr>
<td>Recall</td>
<td>0.73</td>
<td>0.43</td>
<td>0.65</td>
</tr>
<tr>
<td>Kappa Value</td>
<td>0.4667</td>
<td>0.1429</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 6.2: Statistical results for nuclear feature classification

We have achieved accuracy percentages of 0.73 for both G1 and G5, with kappa values of 0.4667 and 0.45 respectively. This significantly outperforms the current pathologist agreement for G1, 0.33, although does not achieve as highly for G5, pathologist kappa score of 0.60. However, a score of 0.45 is still considered moderate agreement. An AP of 0.57 with a kappa value of 0.1429 was achieved for G3, showing poor agreement between the classifier and the
pathologist. This suggests that the classifier often fails to recognise that features shared from G1 and G5 corresponded to G3, and instead often allocates a G3 instance to one of the other two grades thinking that the presence of characteristic G1 and G5 features means it must be one of them. The fact that the recall is less than the baseline (random binary classification should achieve a recall of 0.5) shows that the majority of true G3 instances were classified as NG3- supporting the previous remark. In contrast the precision is 0.6, meaning the majority of instances classified as G3 were actually G3- suggesting once the classifier had finally identified this grade the majority of the time it was sure it was correct. In the light of the other statistics though, and its proximity to the baseline, this conclusion should be considered lightly as it is potentially achieved by chance.

The G5 classifier has the highest precision, 0.76. This actually outperforms the previous precision achieved by Adam, albeit by a small amount, which is 0.75, though as said before we will not look too much into direct comparisons. This shows that about three quarters of the time if we recognise something as G5 we can be certain that it is G5. However, for the same experiments the previous work obtained a recall of 0.86 for G5 whereas we obtained 0.65. This shows that although we are equally as confident that once we have concluded something is a G5 it actually is, we are not as good at actually recognising G5. In the context of patient diagnosis, this is not the most ideal situation.

Since anything that is not G1 and not G5 must be G3, we repeated the experiments but passed the G3 training instances through both trees. This gave us an AP of 0.71 which is almost exactly the same as the current pathologist agreement score of 0.72, and a KV of 0.43. This improved score is probably to do with the idea that it is easier to categorize something as G1 or NG1 and G5 or NG5 than it is for G3 or NG3.

### 6.4 Nuclear Analysis Evaluation

The following tables summarise the output of the nuclear feature classifiers for the three different grades.

<table>
<thead>
<tr>
<th></th>
<th>G1</th>
<th>NG1</th>
<th>G3</th>
<th>NG3</th>
<th>G5</th>
<th>NG5</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>18</td>
<td>12</td>
<td>5</td>
<td>9</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>NG1</td>
<td>11</td>
<td>19</td>
<td>NG3</td>
<td>5</td>
<td>NG5</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 6.3: Confusion Matrices for Tree-G1Nuc, Tree-G3Nuc and Tree-G5Nuc

<table>
<thead>
<tr>
<th></th>
<th>G1</th>
<th>G3</th>
<th>G5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy Percentage</td>
<td>0.62</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>Precision</td>
<td>0.62</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>Recall</td>
<td>0.6</td>
<td>0.35</td>
<td>0.7</td>
</tr>
<tr>
<td>Kappa Value</td>
<td>0.23</td>
<td>0</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 6.4: Statistical results for nuclear feature classification

For grades G1, G3 and G5 we have achieved APs of 62%, 50%, 70% and kappa values of
0.23, 0 and 0.4 respectively: chance, fair and borderline moderate agreement. These values suggest that nuclear analysis isn’t as powerful for grading dysplasia in BO as texture analysis is. Furthermore, they fail to achieve the current agreement by pathologists. Similarly to texture analysis, Tree-G3Nuc was the worst performing classifier, achieving a kappa value of 0 which is equivalent to chance agreement. This again confirms the difficulty in classifying G3 as its lies in the transition period from no dysplasia to high grade dysplasia and therefore has overlap with both of the other grades.

Tree-G1Nuc shows a significantly worse performance than Tree-G5Nuc and Tree-G1Tex, whereas Tree-G1Tex and Tree-G5Tex performed similarly. This may suggest that the changes in our nuclear features don’t immediately occur and aren’t a constant spectrum of increasing severity, resulting in the changes from G1 to G3 not being as noticeable as those from G3 to G5. When we asked the consultant pathologist to comment, he said it was nothing he was familiar with.

There are no massively significant differences between the relative patterns in the precision and recall values between the texture analysis and nuclear analysis. For both G1 experiments, their respective precision and recall values were the same or similar to each other. In both G3 experiments, the recall was around 70% of the precision value. For Tree-G3Nuc the low recall value is particularly concerning as it shows the majority of G3 cases are going unnoticed. In the case that the G3 cases are being classified as G1, it would say that the patient doesn’t need treatment that could potentially stop the development of high grades of dysplasia whilst its still in its early development. In the case of Tree-G5Nuc, it was less precise than Tree-G5Tex, showing less confidence in its correctness of being right about a G5 being a G5, but has a higher recall- showing it could identify actual G5s at a better rate. The differences are not that significant though.

For the G1 and G3 experiments, both the precision and recall values were lower in the nuclear tests showing that the classifiers based on our nuclear features don’t just under perform overall but also do so when looking at the classification of positive classes.

Like the additional G3 experiment for textures, we passed the G3 test instances through Tree-G1Nuc and Tree-G5Nuc. This gave an AP of 0.54 and a KV of 0.07. This gives improved results, likely for the same reason as the texture experiment, but minimally. This is likely due to the fact the G1 and G5 classifiers for nuclear features were not as powerful on our train-test sets as for textural.

Overall, Tree-G5Nuc is the best performing classifier for the nuclear feature experiments.

6.5 Stability

As discussed Section 2.7.1, decision trees can be very sensitive to changes in the input data. This implies that the choice of training and test data could have an adverse effect on the resulting accuracy, and vice-versa. Therefore it was decided it was important to carry out an experiment to assess how sensitive the resulting classifiers are. This was done by taking the features extracted for the train and test sets, for both nuclear and texture analysis, then performing k-fold cross validation on them.
K-fold cross validation takes a set of $N$ observations and divides them up into $k$ groups, or folds. It does this as equally as possible. It then runs the classifier using the first fold as a test set then the other $k-1$ folds as a training set. From this an AP is derived. The process is repeated using each fold as a test set and the others as a training set and an average AP is calculated.

There are a variety of adaptations of this method, including leave-one-out cross validation in which $N$ folds are taken. Analysis done by Kohavi [24] suggested the best form of cross-validation for their model selection was ten-fold stratified cross-validation. The stratified part refers to the algorithm trying to get the same proportions of each of the classes as was in the dataset. Furthermore, they did their testing on a form of decision tree on data that involved breast cancer data, though the specifics of what this data represented were not discussed. Finally, 10-folds are a common practice to use in cross-validation.

Figure 6.1 and 6.2 show the box-plots for all three trees for both textural and nuclei analysis. For each box-plot on each set of axis, the green icons represent KVs and the red icons present APs. Diamonds are average values and boxes are maximum values. The red lines in each box represent the median AP of that experiment, and the top and bottom edges of the box are the 75th and 25th percentile respectively. Red crosses represent outliers, and the whiskers represent the range of data values not considered outliers.

![Box Plots for 10-fold cross-validation of Textural Features](image)

**Figure 6.1**: Box plot for textural features

Note: There is an outlier beneath the red square of G3 and the red square of G5.

All average APs are greater than 0.5, giving support to the fact it is not just random classification. The diagram shows that, particularly for G3 and G5, there is a large spread
of results suggesting that the process chosen is unstable for textural features. Comparing the average APs, 0.6846, 0.58 and 0.5940 for G1, G3 and G5 respectively show that although G3 has the lowest average AP, its proximity to that of G5 makes it not fair to say that it is the hardest grade to classify, as was with previous experiments and expectations. Furthermore, the average AP calculation takes into account outliers from the box plot. If these were excluded, the average AP for G3 would remain relatively unchanged whereas the one for G5 would decrease.

The box-plot for G1 is tightly packed in comparison to the others, suggesting those 5 features are not as sensitive to the choice of training and test as the others are. The fact it has less features than the G3 and G5 experiments, which have 48 and 64 features respectively, is potentially the reason why.

The plots show that there is choices of train test sets that yield high results (max APs of 0.8, 0.9286 and 0.8). This shows that its possible to separate the same data in such a way that high APs and agreements can be achieved, even though the same data can yield poorer results. However, for the latter two these maximum values are considered outliers and could be viewed as miscellaneous results. The next largest values for those experiments are actually 0.67 and 0.7.

![Box plot for nuclear features](image)

**Figure 6.2: Box plot for nuclear features**

Like before, all average APs are greater than 0.5 suggesting non-random classification. Based on the average (0.69, 0.57 and 0.61) APs and KVs the difference in performance between the texture features and nuclear features isn’t as great as the initial experiments suggested, in fact they are rather the same. Interestingly, for G1 the nuclear features give very similar average APs and KVs and max APs and KVs to the G1 box-plot for texture experiments. The box-plots themselves don’t differ too much either, which suggest that both sets of features generalise to similar levels. The G3 box-plot stats for both analysis’ exhibit similar behaviour, though not
to quite the same extent as for G1.

One final note is that the for the nuclear feature box plots, none of the AP values are considered outliers whereas there were some in the case of the texture box plots suggesting relatively the data spread is more likely to fluctuate for textural features. Adam’s work also features outliers for cross-validation of textural, showing more fluctuating behaviour.

6.5.1 Stability Conclusion

In conclusion, it is clear that the extracted features can be unstable. Given the feedback in Section 6.8 and the sensitive nature of decision trees, this instability may be due to the machine learning method rather than the data extracted. A range of APs and KVs have been extracted from 10-fold validation, suggesting if different training and test sets from the initial experiments were taken better classification results may be achieved without cross-validation.

6.6 Texture Analysis Evaluation

The following sections display the decision trees for the texture analysis of the three different grades. In order to understand the variable labels on each of the nodes, a simple mathematical approach was devised, see Appendix G.
6.6.1 Textural Analysis Decision Tree for Grade 1

Figure 6.3: Texture feature decision tree for G1 vs NG1
Figure 6.4: Texture feature decision tree for G3 vs NG3
6.6.3 Textural Analysis Decision Tree for Grade 5

Figure 6.5: Texture feature decision tree for G5 vs NG5
6.6.4 Analysis of Tree-G1Tex

Tree-G1Tex is a relatively balanced tree, suggesting no one straightforward rule in distinguishing a G1 tissue sample. In order to perform a discussion on the node structure, the texture that corresponds to each cluster has to be visually analysed. Each row of Figure 6.6 corresponds to one of the five textures used.

![Figure 6.6: A selection of patches from the 5 clusters, each row is a cluster](image)

It was discovered that the textures that were grouped together didn’t exhibit as similar behaviour as expected. The work done in [2] showed different image patches being grouped based on their textural features like was done in this project, where each group was easily visually distinguishable from the other groups. The lack of such behaviour here initially throws into question the correctness of the clustering method implemented. However, the fact that method captured a cluster that clearly corresponds to that of the image background/other mostly blank patches suggests that the clustering is working and there is another reason for the difference in behaviour. Looking back at the choice of data as discussed in Section 3.7, the previous work selected data that specifically contained the lamina propria and epithelium tissue whereas this project was not as discriminative. This choice could have potentially introduced more of a certain type of tissue/(s) that rather than formed new potential clusters instead ‘diluted’ the current ones.

Nonetheless, there was some minor observable behaviour. As said, cluster one contains mostly blank patches, most likely the image background. Cluster two contained several curved...
structures, though its most convincing quality was its darker nuclei and cytoplasm. Cluster three was a rougher texture, owing to often containing internal white structures, but as can be observed this was totally unique to this cluster. Cluster four exhibited no distinguishable behaviour. Cluster five often contained fine-grained textures that faded structures.

Given this lack of confident clustering, a proper analysis on Tree-G1Tex cannot be carried out. However one remark would be that feature $x5$ is the root, which corresponds to the ratio of texture five that was in the image. Other than texture one, texture five was the cluster to most consistently exhibit the characteristics attributed to it previously, suggesting its a trusted feature. Comparing to the equivalent decision tree in [2], the feature corresponding to the blank textures appears quite low down- whereas they found it to be the most distinguishing feature. Their decision tree also has a lot less nodes, showing that for the data used in this project it was harder to divide the data up based on texture cluster ratios. Again, the use of significantly larger datasets that didn’t discriminate against what was inside them may be what has attributed to this.

6.6.5 Analysis of Tree-G3Tex

Tree-G3Tex is significantly unbalanced at the root node, and given the fact that the third test node down is the root of a sub-tree that is also significantly skewed to one side implies that this tree has latched on to a single rule. This is surprising, as it was suspected that this tree would be balanced as G3 lies between G1 and G5 thus may share qualities with both, resulting in multiple combinations of features that together are characteristic of G3.

It appears that contrast is an important feature in distinguishing between G3 and NG3 based on the data used, since it appears as the first 5 nodes in the tree- at multiple GLCM offsets and angles. In particular, contrast at $0^\circ$ appears as the root node and another two consecutive nodes, as well as once further down the tree, suggesting this particular angle is relevant for distinguishing G3. The fact that contrast is important could be related to the idea that nuclei get darker in higher grades meaning that G3 may contain some of the darker features of G5 and lighter features of G1, creating a greater contrast. Adam’s [2] also found that contrast was an important feature for G3, particularly contrast at $0^\circ$ which appeared three times in their tree, twice in the top three nodes (though they had contrast at $135^\circ$ as the root).

Homogeneity also features quite prominently, appearing four consecutive times at the bottom of the tree and at two points further up. Five times out of six this is at $90^\circ$, implying that it is an important angle. This however could be misleading as it is quite far down the tree meaning that although it was better than other feature choices at that point, it is not necessarily a very good feature in general. Like before, this is similar to Adam’s work whose decision tree for grading G3 only featured homogeneity and contrast, with homogeneity being less significant.

Correlation does not appear much in the tree, suggesting it not important in defining a G3 implying the attribute didn’t produce a significant split between the grades.

6.6.6 Analysis of Tree-G5Tex

Tree-G5Tex is a balanced tree, suggesting there is no one-straight way of determining whether something is G5 or not. A contrast feature, contrast $45^\circ$ at an offset of 1, is the root of the tree,
suggesting that its importance is not only for distinguishing G3. The same feature also appears again further down, giving more attribution to its importance. However it is not as prevalent as it is for G3, and rather correlation appears more important, appearing ten times in the tree.

Homogeneity appears several times in the tree, albeit mostly further down. However, when it does appear it consistently follows the trend that less homogeneous images are not G5 and more homogeneous ones are. This could perhaps be due to the clumping of nuclei, which as observed in Figure 5.1 could be forming larger areas of homogeneous texture.

The tree features mostly the shorter distances from the GLCM analysis, d=1 and d=2. Since this analysis is done on the patch level, larger distances mean further apart in the real image than on the pixel level.

In comparison to Adam’s work, her decision tree contained equal amounts of contrast, homogeneity and correlation nodes, with correlation as the root node. This is partially reflected in our tree where although correlation doesn’t appear as the root, it is quite prevalent in subsequent nodes. Although our contrast doesn’t appear as much as our correlation like Adam’s, it is balanced by the fact that contrast is the root node for us. Then finally, homogeneity also appears quite prevalently.

6.7 Nuclei Analysis Evaluation

Given how the five nuclear features were extracted into feature vectors, in the resulting decision trees they are given the following labels: AD (x1); AI (x2); ANA (x3); AEL (x4); ATA (x5). The following subsections display the final decision trees followed by a discussion of their respective structures.
6.7.1 Nuclear Analysis Decision Tree for Grade 1

Figure 6.7: Nuclear feature decision tree for G1 vs NG1
6.7.2 Nuclear Analysis Decision Tree for Grade 3

Figure 6.8: Nuclear feature decision tree for G3 vs NG3
6.7.3 Nuclear Analysis Decision Tree for Grade 5

Figure 6.9: Nuclear feature decision tree for G5 vs NG5
6.7.4 Analysis of Tree-G1Nuc

Tree-G1, Figure 6.7, is relatively balanced, showing no significant skew in number of nodes to either side of the route node. This suggests that there is no single rule in grading tissue as G1. It also contains feature $x_5$ at the root suggesting that the ATA is the most distinguishing feature for that grade. Feature $x_4$ appears as the next node on the right-side of the tree which could be to do with the fact that the AEL and ATA are closely linked, so since ATA is an important feature AEL appears reasonably high in the tree. Both $x_4$ and $x_5$ appear appear several times in the tree on both sides, giving further indication of their importance for distinguishing G1 from the other grades.

Feature $x_1$ appears three times in the decision tree, each time being a final test node. Being further down in the tree it says the AD is not a characterising feature, which could be due to a lack of colour normalisation and the conversion to gray-scale not counteracting the potential variances between actually equivalent nuclei as discussed in Section 2.4. What is interesting to note is that in all three of these cases, the darker nuclei at that node in the tree correspond to NG1, which lines up with the theory. Similarly the two times $x_3$ appears as a final test node, it implies larger nuclei correspond to higher grades and the one time $x_2$ appears as a final test node implies that more irregular nuclei correspond to higher grades. Furthermore, the last test node on the far left, $x_4$ also supports the idea that larger edge lengths are present in grade 1 as the nuclei are less bunched together. However, it is important to note that some of the leaf nodes branching off further up in the tree do go against the theory.

6.7.5 Analysis of Tree-G3Nuc

Tree-G3 is not balanced at the root node, and given the fact that the fourth test node down is the root of a sub-tree that is significantly skewed to one side implies that this tree has latched on to a single rule. This is surprising, as it was suspected that this tree would be balanced as G3 lies between G1 and G5 thus may share qualities with both, resulting in multiple combinations of features that together are characteristic of G3. Feature $x_2$, AI, is the root node implying it is an important feature for distinguishing between this and the two other grades. In this case, it says more regularly shaped nuclei correspond to NG3 which can’t be over-fitting to G1, given the way the test set was chosen. If the high-level ideal description of the progression of grades was correct, then looking at one feature alone should not be enough for distinguishing whether something is G3 or not in a binary classification system, so the fact the root node has a leaf node is surprising.

One interesting observation is how going to the right child of the root, then the right child of the $x_3$ test node gives a classification of NG3. This is not surprising as the path to this leaf node corresponds to two features that sway to one side of G3, more irregular nuclei that are also larger, which would also make it not surprising if this leaf node contains a significant amount of G5 entries from the test set. On a similar note, by observing the path to the closest G3 from the root node there is a combination of responses to the tests that would be expected from both G5 and G1.

Like before, $x_1$ appears further down in the tree giving question to its relevance. In fact, every other feature appears in the tree before it appears for the first time.
6.7.6 Analysis of Tree-G5Nu

Tree-G5 is noticeably un-balanced, though not to the same extent as Tree-G3. The root node suggests that ANA is an important feature for determining G5. The next couple of nodes are a mix of $x_2$ and $x_5$, which were root nodes of the other two trees, suggesting their importance is still relevant here.

The lowest test node, $x_4$, suggests larger edge lengths are more prevalent in G5 than in lower grades- going against the theory. However, this node is far down the tree, with many leaf nodes before it, meaning it is probable that only a small subset of the test set reaches this point. This small subset might contain exceptions to the rules, as we know they are not clean-cut, and subsequently does not capture the general trend.

The decision tree also features feature $x_2$, AI consistently near the top of the tree. We would expect more irregular nuclei to appear in the decision tree, but a few of the AI nodes suggest that for the test instances that made it to that node- the less irregular ones were G5.

All paths to root nodes, with the exception for one, contain at least three different nuclear features, several even include four different features, suggesting that no single nuclear characteristic can distinguish a grade. This once again shows that lack of a clean-cut rules.

6.8 Pathologist Feedback

Early on in the project we raised a question about why a particular image was a particular grade. The pathologist explained why it was the given grade, but explained how they understood why it would be considered otherwise by other experts. Therefore it was decided that when classification was complete a sample of the image that was incorrectly classified would be sent to get feedback on why they were misclassified. A total of 30 image, 15 misclassified from the textural analysis and 15 misclassified from the nuclear analysis, were sent. Amongst these, there were 5 images from each grade from each analysis. The comments are given in Appendix D. The table shows once more how the current grading of dysplasia in BO is a subjective field as even with expert pathologists there is disagreement and uncertainty. 15 of the images contained comments that either said the ground truth value was wrong (in the opinion of the consultant pathologist), it could be another grade or it was understandable why it was classified otherwise. This potentially explains some of the falsely classified instances.

6.9 Data Remark

Although the data in this project used balanced amounts of each of the three grades in its training and test sets, Section 3.7, it did not use a balanced amount of each of the more detailed grades in each of the grouped grades. The justification for this, as confirmed by the project supervisor, was that in order to use the full extent of the data and to view the three grades as actual independent classifications rather than groups of classification we would not make such a discrimination. Nonetheless, a short time was spent adapting the training and test sets to test such data which for the nuclear experiments gave APs of 0.58, 0.67 and 0.58 for G1, G3 and G5 respectively. The fact the original G1 experiment was better, could be due to the
fact their being an abundance of detailed G1 data entries which could have skewed the grouped
G1 to the lower end of the spectrum but this isn’t the case for the other grades implying the
use of such data just didn’t work as well in our case.

6.10 Methodology Evaluation

The ability to not have to spend time developing a highly structured schedule and to be able
to re-define goals in a quick manner was very useful in this project. This is shown by actions
that had to be taken at the start. Initially we had some goals in mind: texture analysis and
spatial analysis with some rough understanding that spatial analysis could be applied to nuclei
or texture types. As part of the background research and before any real progress could be made
a meeting had to be arranged with the consultant pathologist in order to discuss our ideas, get
some feedback on them in addition to guidance on avenues to take, but most importantly to get
an initial understanding in Barrett’s Oesophagus as no previous knowledge was known. Whilst
waiting for the meeting, the data that would be used for the project was made available (see
Section 3.7). Time was spent understanding the fields in the data, and any questions were noted
in preparation for the meeting.

The meeting confirmed that three different avenues (texture analysis, spatial analysis of
nuclei and texture types), which also involved learning the medical side of things, were appro-
priate and further discussion was had which identified texture analysis to be the first thing to
look at. The meeting also answered the questions that were had about the data, which a short
amount of time was then spent making sure that these could be accessed appropriately. In-line
to the proposed methodology, it was now possible to do some initial implementation whilst still
keeping spatial analysis vague.

From this stage, the project involved weekly meetings with the project supervisor. Prelim-
inary results and outputs were taken to these meetings which caused certain decisions to be
made. For instance, the adoption of overlapped patches was taken in attempt to capture more
information when preliminary results did not act as expected. Also fundamental problems with
the current approach were identified and corrected.

Since this methodology adopts minimal planning, the initial meetings with the pathologist
and supervisor did not go into detail on the spatial aspects as they were not being carried
out at the time and as to retain the flexibility if different delays occurred. When the texture
analysis took longer than expected, it was easy to eliminate the broader spatial classifier and
focus just not nuclear feature classification. Close to that stage in the project, another meeting
was arranged with the pathologist to report on results so far and discuss features that could
be obtained, showing not all aspects of the system were initially planned. The pathologist
highlighted one approach for nuclear detection (Section 5.1.2), but this was disregarded in
favour for another approach (Section 5.1.1) which gave more detailed information. When this
approach was found to have some flaws, the RAD approach made it easy to decide to go back
and explore the first one even though it would require the whole thing to be reimplemented.
It was also easy to then decide to not do additional research into nuclei in BO and focus on a
simplified set of features, due to this increase in time implementing.
Chapter 7

Conclusion

7.1 Summary

To conclude, we have shown that textural analysis and nuclear analysis can be used in order to grade images of Barrett’s Oesophagus producing positive APs and KVs, more specifically when using decision trees, however the results depend quite significantly on the data sets used. We were only able to beat the current levels of agreement with pathologists for G1 textural features, but achieved moderate and borderline moderate agreement for G5 textural and nuclear features. However, our tests on the stability of our features suggested it may be possible in some cases to significantly improve our results, but on average our system will mostly not improve upon the current agreement.

Although the final tests were carried out slightly differently in comparison to the previous work, making it hard to form direct comparisons of classification results, we were able to find ways of comparing by looking at the decision trees and box plots and found that similar patterns occurred with regards to the best and worst performing classifiers based on the grades they’re classifying. We did not manage to achieve visually distinguishable clusters of textures like the previous work, but suggested this may be to do with the choice of our data.

Overall, our textural analysis performed better than our nuclear analysis in all cases but the box plots suggested they were not that dissimilar, and that the nuclear features may in fact contain less outlying cases.

7.2 Aims and Objectives

This project has achieved its aims, objectives and minimum requirements in addition to one of the two further extensions. All deliverables have been produced.

**Produce a methodology that is capable of taking in an input image of a region of tissue from a virtual slide and producing a new image that described the clusters of textures within it:** the images were extracted by manipulation of the server URLs in order to give the desired region at a desired resolution. We performed clustering on the texture features of image patches to form five texture types which are then labelled by a unique grey level.
Implemented machine learning techniques to train three classifiers for distinguishing between whether a new image belongs to a certain grade or not for three different grades: By creating balances training and test sets, we were able to extract features based on textural analysis and nuclear analysis that were characteristic of the two different grades and formed decision trees based on the importance of these features. We grouped a six grade scale into a three grade scale due to limited data for two grades and improve classification results whilst retaining an appropriate amount of output information.

Evaluate results in comparison to those produced in previous experiments [2]: this was achieved by extracting various statistics from the classification results as well as from visual comparison of decision trees, Chapter 6. Due to the different experiments direct comparisons were not always possible, but stability investigations closed this gap and comparisons were made between decision trees.

Investigate link between spatial relationships of nuclei and the grade of the tissue: By implementing two different nuclear detection techniques and from our background reading into the diagnosis of Barrett’s Oesophagus we were able to capture five features to which we used to classifier unseen instances. These were compared to the texture features in Chapter 6.

7.3 Limitations

Several limitations could have effected the classification power of the produced solution. The calculation of the nuclear irregularity could do with some improvement. Although the calculation worked, it was dependent on the quality of the nuclear detection program, which contained noise in which the detected boundary wouldn’t always follow the nuclei boundary.

Since we couldn’t get colour normalisation to work to good enough standards, the effectiveness of the average darkness calculation is somewhat limited. Furthermore, the pathologist feedback also highlighted that darker nuclei can sometimes come into the diagnosis of lower grades. Although our generalised view of nuclei being darker in higher grades was deemed appropriate by the pathologist for this level of project, this is obviously somewhere that needs further investigation and lower level less generalised assumptions should be made.

In Section 5.1.1 we justified that the use of the original nuclear detector to extract the features AD, AI and NA as it wouldn’t be effected by the loss of information regarding clumped nuclei. This is probably not the case.

The graph-cuts based method for segmenting clumps of nuclei was limited by the fact it had no knowledge of nuclei boundaries, and cut the graph based on an estimation of number of nuclei present. This method could be improved.

The nuclear features were quite limited to the information they captured as they were only means or means of standard deviations. Although some experiments were done to try and investigate whether statistics other than the mean would be useful, i.e. looking at histogram bins, they showed no promising patterns. It would be good if someone spent more time investigating different ways of representing features from images.
7.4 Future Work

In addition to the previous points, there is other future work that could be done.

- It was discovered that the clustering did not work as effectively on the larger dataset as it did for the previous works smaller dataset. Further investigation should be done into whether this method is an effective way of capturing information about clusters of textures. Alternative ways of describing textures in order to cluster patches could also be investigated.

- One possible further extension for this project that was not achieved due to time contraints was the investigation of the spatial arrangements of groups of texture. This was partially captured in the texture analysis section, when GLCM analysis was carried out upon the CCI’s to generate the CCM’s. However, grouping texture types that are the same into connected components and investigating their spatial relations may also yield useful information. This avenue could also make use of qualitative logical reasoning about space or other such methods to describe texture types engulfing other texture types, partially or fully, and various other descriptors. This information can be captured to learn ways of identifying important structures characteristics of different grades of dysplasia.

- Adam [2] investigated the spatial relations between clusters of texture and the epithelial layer, in particular the distance between the two. They achieved AP’s of 75%, 68.8% and 68.8% with KV’s of 0.5, 0.37 and 0.37 for grading G1, G3 and G5 respectively using binary decision tree models. These did not perform as well as their texture analysis techniques like implemented in this report, so it may be interesting to combine this research with the previous proposed extension and see what classification power could be achieved.

- The pathologist feedback highlighted that probable reasons for some of the annotated regions to be misclassified may be due characteristics of inflammation being mis-understood as being representative of G5. Perhaps some tool that identifies inflammation would be useful. Similarly, fragmentation in tissue samples was highlighted as a possible reason for confusing the classifier. Filtering these out either manually or by investigating the prevalence of texture types representing the image background may be a useful image pre-processor for future classifiers.

- It was decided to not optimize the thresholds for the original nuclear detector. This could be something that someone in the future could investigate.

- It would be good to see someone use the code we developed to split an image into patches to find the optimal overlap for this dataset.

- As was highlighted in Section 3.7, pathologists don’t just look at these annotated regions in individual sessions but look at several in a virtual slide and their impression of each affects the impression of the others. How our methodologies scale up to virtual slides would be an interesting avenue for exploration.
Bibliography


[26] Derek Magee, Darren Treanor, Doreen Crellin, Mike Shires, Kevin Mohee, and Philip Quirke. Colour normalisation in digital histopathology images.


Appendix A

Personal Reflection

I was aware of there being a big independent project in the final year of my degree from my first year. Back then, when I would see third year students working on it, it was a daunting prospect that I would one day be doing one as it was nothing I had done before. As I developed as an individual and a computer scientist throughout my first, second and third years the idea of the FYP became less concerning and after going through what was probably my most intellectually demanding term at University at the beginning of the third year I was actually looking forward to working on something that I could just dedicate all my work-time to. I went into the project feeling confident and excited, but upon reflection I was surprised to both a greater and lesser extent.

First of all, I spent a considerable amount of time thinking about what project I was going to choose. I feel it is very important to choose something that interests you as it will keep you motivated and help ensure you are giving more than minimal effort. I would advise all future students to not take this choice lightly, and spend time reading through the choices available and also, what I think is very important, go speak to the academic who would be supervising that project. This is an excellent opportunity to get a proper insight into what the project involves and whether it would actually live up to your initial expectations. Some project that I was actually very keen to take I ended up disregarding from doing this. I chose a Computer Vision project in a medical field as I found the content of the Computer Vision module very enjoyable, and I have a general interest in science so I thought it would be a good opportunity to learn something new.

I found the background research particularly difficult. I wasn’t used to having to justify my choices so formally, or to reading academic literature for that matter, and found this stage particularly stressful. I grew impatient with reading and wanted to get on with actually coding. My supervisor advised me to stagger my reading with sessions of implementation, to give me a chance to re-charge. This was very useful, and would be something I advise to other students who faced similar frustrations. It also allows you to test your understanding of what you have read.

I have learnt some transferable skills that I could take to a future career in both academia and industry. One such skill is my ability to analyse and critique academic literature, which is obviously transferable to academia but also would help in industry as I may have to read tech-
technical documents and try and highlight the potential draw back and advantages of its contents. One real application of this in this project was the fact that I had to be really conscious about what other data people were using in their experiments when doing my background research, as their potentially appealing looking results may just be suitable to their problem domain/data and not transferable to what I wanted to achieve. The experiments in my evaluation re-iterated the significance of slight variations in the data effecting results. This is something I would tell future students to be weary of when reviewing others and their own results.

When I experienced delays in the middle on my implementation phase, I didn’t carry out a weeks evaluation alongside further implementation. Rather than re-working this in, I just performed minor experiments that gave promising results, but were not thorough enough to be a formal evaluation. If I had re-worked in this early formal evaluation, it would have made me realise potential problems in my approach, such as the lack of a nice clustering of texture types, and given me time to investigate them and potentially improve them in time for my final evaluation.

I feel I have also developed a nice academic writing style. This was partially to my exposure to the academic literature throughout this project but also writing the different sections required in a research write-up, particularly the background research and evaluation chapters.

Another reflection I would make would be that since my work was based quite a lot on previous work, I would often feel disheartened when something wouldn’t happen as planned. On occasions I got quite stressed, thinking that I had done something wrong just because I had done something different to the previous work, due to misunderstanding their paper. However, I have gained confidence in myself and choices that I make and have learnt that it often advantageous to try something different to someone else even if they got really good results because even if you fail to achieve as highly, you can at least say why and show that a particular method may not be suitable.

Overall I am happy with the project, and have learnt some valuable skills and work ethics that I know I would be able to apply to future project, particularly research based ones.
There were several material used in this project that were not my own. The following materials were provided for us by members of staff of the Computing department at the University of Leeds. Any relevant papers are also referenced:

- Nuclear Detection Code: [7] [46].
- Code to draw the edge responses from nuclear detection as in Figure 5.1.
- Code to create color deconvoluted images: [35].
- Code to perform colour normalisation on images: [26]

Additional materials:

- Dataset of digital pathology annotations: Leeds Institute of Molecular Medicine

All toolboxes available for MathsWorks that are available on the University machines were also used. No additional toolboxes were used.
Appendix C

Ethical Issues

This work falls under Dr Treanors Local Research Ethics Committee Approval (Leeds West LREC 05/Q1205/220)

This project used data provided by the Leeds Institute for Molecular Medicine (LIMM). This data comprised of images of patient tissue taken from a biopsy, as well as the the grade a pathologist had given various regions and various values representing coordinates and dimensions of those regions. This data contained no metadata about the identification of the patients the tissue samples belonged too. Furthermore, the identity of the pathologists who gave a grade for each region is anonymised, they are only refer to as EXPERT B, TRAINEE A etc. Image data was strictly kept on machines within the School of Computing, and never transferred to personal devices such as home machines or laptops, or to cloud based remote storage. This also meant that whenever the report was worked on in a different location to the School of Computing, any images included in the write-up were temporarily removed. Other Ethical issues are discussed in the reference above.
Appendix D

Consultant Pathologist Responses

The first column of the table contains the number corresponding to the url of the annotated region in the list following the table. The second column contains the true classification, with the new grading scale and the traditional detailed classification in brackets. The third column contains the pathologists comments.

<table>
<thead>
<tr>
<th>True Classification</th>
<th>Consultant Pathologist Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 G1 (Negative)</td>
<td>Difficult. This area looks really dysplastic but was probably selected as an example of an area of G2 atypia (prob reactive) but others might call it G3 or G4</td>
</tr>
<tr>
<td>2 G1 (Negative)</td>
<td>Could be G2 or G3</td>
</tr>
<tr>
<td>3 G1 (Indefinite (Probably negative))</td>
<td>Its G1 or G2 - difficult because its inflamed which makes the nuclei more atypical</td>
</tr>
<tr>
<td>4 G1 (Negative)</td>
<td>Very large nuclei, could be G2 or G3</td>
</tr>
<tr>
<td>5 G1 (Negative)</td>
<td>Not sure why this was incorrect</td>
</tr>
<tr>
<td>6 G5 (High grade dysplasia)</td>
<td>Looks like G4 or G5 to me. The tissue is a little fragmented which might explain the incorrect response</td>
</tr>
<tr>
<td>7 G5 (High grade dysplasia)</td>
<td>Horrible looking nuclei, must be HG5</td>
</tr>
<tr>
<td>8 G5 (Intramucosal carcinoma)</td>
<td>EXCLUDE this area of inflammation is not representative of G5</td>
</tr>
<tr>
<td>9 G5 (Intramucosal carcinoma)</td>
<td>Same as above but background cells are G5</td>
</tr>
<tr>
<td>10 G5 (High grade dysplasia)</td>
<td>G5</td>
</tr>
<tr>
<td>11 G3 (Low grade dysplasia)</td>
<td>Tricky, others might call it G2 or G4</td>
</tr>
<tr>
<td>12 G3 (Low grade dysplasia)</td>
<td>Again controversial, could be G1 or G2</td>
</tr>
<tr>
<td>No.</td>
<td>Grade and Description</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>G3 (low grade dysplasia)</td>
</tr>
<tr>
<td>14</td>
<td>G3 (Indefinite (Probably dysplastic))</td>
</tr>
<tr>
<td>15</td>
<td>G3 (Indefinite (Probably dysplastic))</td>
</tr>
<tr>
<td>16</td>
<td>G1 (negative)</td>
</tr>
<tr>
<td>17</td>
<td>G1 (negative)</td>
</tr>
<tr>
<td>18</td>
<td>G1 (negative)</td>
</tr>
<tr>
<td>19</td>
<td>G1 (Indefinite ( Probably negative))</td>
</tr>
<tr>
<td>20</td>
<td>G1 (Indefinite ( Probably negative))</td>
</tr>
<tr>
<td>21</td>
<td>G3 (low grade dysplasia)</td>
</tr>
<tr>
<td>22</td>
<td>G3 (Low grade dysplasia)</td>
</tr>
<tr>
<td>23</td>
<td>G3 (Low grade dysplasia)</td>
</tr>
<tr>
<td>24</td>
<td>G3 (Indefinite (Probably dysplastic))</td>
</tr>
<tr>
<td>25</td>
<td>G3 (Indefinite (Probably dysplastic))</td>
</tr>
<tr>
<td>26</td>
<td>G5 (High grade dysplasia)</td>
</tr>
<tr>
<td>27</td>
<td>G5 (High grade dysplasia)</td>
</tr>
<tr>
<td>28</td>
<td>G5 (High grade dysplasia)</td>
</tr>
<tr>
<td>29</td>
<td>G5 (Intramucosal carcinoma)</td>
</tr>
<tr>
<td>30</td>
<td>G5 (Intramucosal carcinoma)</td>
</tr>
</tbody>
</table>

- G3
- G1
- G2
- G3/4
- G5
- G6
- G2/3
Appendix E

Schedule

![Initial Project Schedule](image)

Figure E.1: Initial Project Schedule
<table>
<thead>
<tr>
<th>Tasks</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background Research</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reading in images</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic impl. Of Texture Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic impl. Of Machine Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid Project Report Writing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Writing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milestones/Deadlines</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aim and Objectives form</td>
<td>31/01/12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid Project Report</td>
<td></td>
<td>10/03/12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re-implemented texture classifier</td>
<td></td>
<td>31/03/12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add nuclear classifier</td>
<td></td>
<td>15/04/12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Progress meeting form</td>
<td></td>
<td>30/05/12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Submission</td>
<td></td>
<td>31/05/12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix F

Code Evaluation

This project is very much an exploratory style of project, therefore the performance of the code with regards to running times is not a key component of the overall evaluation, and not an important aspect for the success of the project. However, during development it became clear that some of the different feature extraction methods did consume significant amounts of time. In anticipation of any problems that might be encountered during evaluation that may delay the project, a brief analysis of the code was carried out pre-evaluation to highlight any possible areas for re-factoring. Then, an analysis of the run times of the code is carried out to give the reader an impression of how long things took to process.

F.1 Improvements

This analysis was carried out only on the nuclear analysis side of the project. The motivating factor for this was the fact that the Desktop Anywhere sessions were unreliable, Section 5.3, and the texture analysis aspects could just be left running over night.

The following improvements were made:

- For loops removed in code to extract average nuclear darkness from an image. Functional programming style with smart indexing adopted instead. Time taken to run on an image with 1062 detected nuclei centroids improved from 235.4 seconds to 39.6 seconds. Similar results achieved for all images.

- Similar changes made to code that extracts average irregularity. Time taken to run on the same image improved from 16.6 seconds to 0.42 seconds.

At this stage in the project it was decided there was no particular concerns with any of the running times of other aspects of the code.

F.2 Code Timings

The images were ordered in order of size, in pixels. The average of this was taken, which was found to be roughly 7385200 pixels. Ten images around this average were extracted and use to test the running times of different aspects of the code.
• **Extract CCM features (EXP1, A1):** The total time was 458 seconds, with an average time of 46 seconds. The processing of this part of the code depends mainly on the number of patches resulting from patch creation.

• **In-house nuclear detection:** The total time was 2396 seconds, with an average time of 240 seconds. There was however variance in this, with the shortest time being 43 seconds and the largest being 785 seconds. The running time on this depends on the amount of nuclei present on the image.

• **HSV and graph-cuts based nuclear detection:** The total time was 2595 seconds, with an average time of 259 seconds. Once again, there was high variance in these timings as the quickest time was 19 seconds and the longest time was 1226 seconds. The running time on this depends not just on how many nuclei are present in the image but also on how large the connected components are. For example, the image that took 1226 seconds in this approach, only took 180 seconds (fifth slowest) in the other approach.

It is important to note that the timings for the nuclear features were relatively slower due to the windows machine they were running on.
Appendix G

Node Labelling Method Texture Analysis

G.1 Decision Tree Node Labelling

G.1.1 Tree-G5Tex Node Labelling

To find out which of the four GLCM features the label corresponded to, we first divide the label number by four. The resulting number shall be denoted \( \Phi \). The decimal part of \( \Phi \) corresponds to a GLCM feature like so:

<table>
<thead>
<tr>
<th>GLCM Feature</th>
<th>Decimal Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td>.25</td>
</tr>
<tr>
<td>Correlation</td>
<td>.5</td>
</tr>
<tr>
<td>Energy</td>
<td>.75</td>
</tr>
<tr>
<td>Homogeneity</td>
<td>.0</td>
</tr>
</tbody>
</table>

Divide \( \Phi \) by four again and inspect the decimal part of this number, which gives the angle used in the GLCM based on the following categories:

<table>
<thead>
<tr>
<th>GLCM Angle</th>
<th>.0625-.25</th>
<th>.3125-.5</th>
<th>.5625-.75</th>
<th>&gt;.8125 or 0</th>
</tr>
</thead>
</table>

Then rounding this same number up to the nearest integer gives the offset.

G.1.2 Tree-G3Tex Node Labelling

Since Tree-G3Nuc omitted correlation when extracting the CCM features from the CCI, it was described by a different sized feature vector to those from the G5 experiment. Therefore to label the features, a slightly different approach is adopted. To find out which of the three GLCM features the label corresponds to, we first divide the label number by three. This number shall be denoted \( \Upsilon \). The decimal part of \( \Upsilon \) corresponds to a GLCM feature like so:

<table>
<thead>
<tr>
<th>GLCM Feature</th>
<th>Decimal Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast</td>
<td>.33</td>
</tr>
<tr>
<td>Energy</td>
<td>.66</td>
</tr>
<tr>
<td>Homogeneity</td>
<td>.0</td>
</tr>
</tbody>
</table>

Divide \( \Upsilon \) by four and inspect the decimal part of this number, which gives the angle used in the GLCM based on the following categories:
Then rounding this same number up to the nearest integer gives the offset.

<table>
<thead>
<tr>
<th>GLCM Angle</th>
<th>0°</th>
<th>45°</th>
<th>90°</th>
<th>135°</th>
</tr>
</thead>
<tbody>
<tr>
<td>.083-.25</td>
<td>.33</td>
<td>.58</td>
<td>.75</td>
<td>&gt;.83</td>
</tr>
<tr>
<td>.833 or 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

G.1.3 Tree-G1Tex Node Labelling

Each node number simply corresponds to the cluster ratio.
Appendix H

System Design

Figure H.1: Basic Pipeline for Classifying a train-test set of images based on Nuclear Features

Figure H.2: Basic Pipeline for Classifying a train-test set of images based on Textural Features