School of Engineering and Design

Brunel University

Pattern Recognition Systems Design on Parallel GPU Architectures for Breast Lesions Characterisation Employing Multimodality Images

A thesis submitted for the degree of Doctor of Philosophy

By

Konstantinos Pavlos Sidiropoulos

2014

LONDON, UK

[This page was intentionally left blank]

To my brother

and sisters

Table of Contents

Table of Contents .... *iv*

List of Figures .... *viii*

List of Tables .... *xii*

Abbreviations .... *xiv*

Acknowledgements .... *xvii*

Publications .... *xviii*

Abstract .... *xx*

Chapter 1. Introduction and motivation *1*

1.1 Overview **1**

1.2 Aims and Rationale **2**

1.3 Contribution **6**

1.4 Thesis Outline **7**

1.5 Research Funding **8**

Chapter 2. Breast Cancer *9*

2.1 Overview **9**

2.2 Breast Anatomy and Physiology **10**

2.3 Breast Cancer Epidemiology **11**

2.4 Imaging Modalities for detection of Breast Cancer **12**

2.4.1 Digital Mammography 12

2.4.2 Ultrasound 13

2.5 Digital Mammography characterisation of breast masses **15**

2.6 Ultrasound characterisation of breast masses **15**

2.7 Summary **18**

Chapter 3. Introduction to Statistical Pattern Recognition ***19***

3.1 Overview **19**

3.2 Basic principles of pattern recognition **20**

3.3 Segmentation **22**

3.4 Feature extraction **22**

3.4.1 Histogram features 23

3.4.2 Features calculated from the Co-occurrence Matrix 25

3.4.3 Features Extracted from the Run-Length Matrix 29

3.5 Data Normalisation **31**

3.6 Feature reduction – selection **31**

3.6.1 Scalar Feature Selection 32

3.6.2 Feature Vector Selection 33

3.6.2.1 Optimal Feature Searching 35

3.6.2.2 Suboptimal Feature Searching 36

3.7 Classification **38**

3.7.1 Bayesian 39

3.7.2 Linear Discriminant Analysis 39

3.7.3 k-Nearest Neighbours 40

3.7.4 Probabilistic Neural Network 40

3.7.5 Support Vector Machines 41

3.8 Evaluation strategies **41**

3.9 Summary **46**

Chapter 4. GPUs as Parallel Processing Accelerators ***47***

4.1 Overview 47

4.2 The dawn of the Multi-Core Era 48

4.3 Parallel Programming 49

4.4 The evolution of GPU as programmable parallel processor 52

4.5 GPU Architecture Overview 55

4.6 The CUDA Execution Model 59

4.7 Limitations of GPUs -Considerations 61

4.8 Summary 63

Chapter 5. System Design and Implementation 64

5.1 Overview 64

5.2 Data Collection and Clinical Material 65

5.3 Ethical Issues 66

5.4 Breast Lesions' Delineation 67

5.5 Feature Extraction 69

5.6 Classifier Selection 72

5.7 PR System Design 73

5.8 Feature Selection Methods 73

5.9 PR-System Performance Evaluation 75

5.10 Parallel Processing GPU Platform 76

5.11 Summary 84

Chapter 6. System Evaluation 85

6.1 Overview 85

6.2 Evaluation of System Accuracy 86

6.3 Evaluation of GPU acceleration on System Design 95

6.4 Discussion on the Results 98

6.5 System Limitations 103

6.6 Summary 104

Chapter 7. System adaptation to other Medical Imaging applications 105

7.1 Overview 105

7.2 Real time Decision Support System for diagnosis of rare cancers 106

7.3 Discriminating between patients with micro-ischaemic and multiple sclerosis lesions, using MRI images 107

7.4 Discriminating between injured normal and pathological knee cartilage 109

7.5 Differentiating between low and high grade laryngeal cancer cases 111

7.6 Summary 113

Chapter 8. Conclusions – Future Work 114

8.1 Overview 114

8.2 Conclusions 115

8.3 Future Work 117

References 118

Appendix A 127

Appendix B 129

List of Figures

Figure 2.1: A graphical illustration of the female breast anatomy. Reproduced from Marieb and Hoehn (2013).. 11

Figure 2.2: A typical DM image.. 13

Figure 2.3: A typical US image 14

Figure 3.1: The basic stages involved in the design of a classification system. Adapted from Theodoridis and Koutroumbas (2003) 21

Figure 3.2: Examples illustrating how Skewness and Kurtosis are indicative of the histogram's shape 24

Figure 3.3: Calculation of GLCM at angle 0° 25

Figure 3.4: Filter and Wrapper feature selection approaches 34

Figure 3.5: Pseudocode for SFS algorithm.. 36

Figure 3.6: Pseudocode for SBS algorithm 37

Figure 3.7: Pseudocode for SFFS algorithm.. 38

Figure 3.8: Pseudocode for SBFS algorithm 38

Figure 3.9: Different evaluation techniques.. 44

Figure 4.1: The two main memory models used in parallel programming. 51

Figure 4.2: Comparison of Floating-Point Operations per Second (FLOPS) and memory bandwidth for the CPU and GPU (Nvidia, 2014) 53

Figure 4.3: Typical CPU and GPU architectures... 53

Figure 4.4: Hardware structure of a workstation featuring a GPU 56

Figure 4.5: Block diagram of Tesla-class GPUs (Wolfe, 2010) 57

Figure 4.6: Block diagram of Fermi-class GPUs (Wolfe, 2010) 57

Figure 4.7: Streaming Multiprocessor of Tesla (left) and Fermi-class (right) GPUs (Wilt, 2013) 58

Figure 4.8: Example of a GPU-based program that multiplies a set of data by the number 2... 59

Figure 4.9: Thread hierarchy in the CUDA programming model (Nvidia, 2014) 60

Figure 5.1: Percentages of benign and malignant breast cancer cases in the collected dataset. 66

Figure 5.2: From each case Digital Mammography and Ultrasound ROIs were delineated. First row presents the ROIs from a benign case while the second row presents a malignant case 68

Figure 5.3: Screenshot of the custom made software developed for the purposes of this research. The software enabled the physician to browse for a case and delineate the region of interest.. 69

Figure 5.4: The flow diagram of the procedure followed for creating the three datasets... 71

Figure 5.5: Block diagram describing the training of the proposed PR system, steps 1, 2, 4, 8 and 9 are sequentially executed on the CPU; steps 3, 5 and 7 refer to memory transactions between CPU and GPU; and step 6 is executed in parallel on the GPU, employing numerous threads.. 76

Figure 5.6: The flow diagram of the PR-system design task. Left part illustrates the sequentially executed task on CPU. Right part illustrates the same task as it has been modified to run in parallel on GPU 77

Figure 5.7: The task of each thread, running concurrently, was to train the PNN classifier with a unique feature combination and evaluate its classification accuracy by means of the leave-one-pattern-out technique. In every case the output of each thread was the overall accuracy achieved for the specific feature combination. 78

Figure 5.8: Pseudocode illustrating the PNN training in each CUDA thread *t* for a single feature combination. 78

Figure 5.9: Pseudocode illustrating Host implementation not including the ECV method.. 80

Figure 5.10: The kernel used to design the proposed GPU-based PR system 83

Figure 6.1: Overall accuracies achieved by the two sub-optimal and one optimal features-selection methods, and the LOO evaluation method using the three datasets (US, DM and US & DM combined) 89

Figure 6.2: Partial and overall accuracies achieved via the optimal feature selection method and LOO, applied on GPU, using features from US images, features from DM images, and combined features from US and DM images 90

Figure 6.3: Quadratic LS-mappings of benign and malignant data, employed in the optimal PNN design using (a) US, (b) DM, and (c) combined US and DM features 91

Figure 6.4: Box-plots of the six textural features that formed the best feature combination in the optimal PNN design. The features' values are not normalised. (STD: Standard Deviation, SOQ: Sum of Squares, SAV: Sum Average, SRE: Short Run Emphasis, DENTR: Difference Entropy, where a: average, r: range, _DM: features from DM images and _US: feature from US images. 92

Figure 6.5: Comparative evaluation of processing times for 2 different NVidia GPU cards (GeForce 8800GT and 580GTX) and standard CPU system (Intel Core 2 Quad). Times refer to the processors (GPUs or CPU) executing the same task, by forming all possible combinations of 6 features within a pool of 20, 30,..,or 80 features 96

Figure 6.6: Comparative assessment of CPU (single and multi-core) and GPUs (8800GT and 580GTX) against the number of patterns in the dataset, for the task of classifier training with up to 20 feature combinations from a pool of 20 features 97

Figure 7.1: (a): An example of a high grade pleomorphic glioblastoma multiforme tumour, (b): Segmentation of Figure1a for isolation of nuclei from surrounding tissue 106

Figure 7.2:Custom-made software tool for extracting lesions........ 108

Figure 7.3: Flow diagram of parallel processing performed on the graphics processing unit........ 109

Figure 7.4: Snapshots of the three PD-FSE FS MRI planes. In these planes and in their corresponding slices, the following ROIs were extracted: from the axial planes: (a) patella and (b) femoral trochlea, from the coronal planes; (c) lateral and medial of femoral and tibial condyles, from the sagittal planes: (d) patella and femoral trochlea, and (e) lateral and medial of the MFC and MTC cartilages........ 110

List of Tables

Table 2.1: Representative lesions of benign cases employing DM and US................ 16

Table 2.2: Representative lesions of malignant cases employing DM and US........... 17

Table 5.1: A summary of the verified breast cancer cases comprising the collected dataset. 66

Table 5.2: List of textural features employed in the current study along with their abbreviations for DM and US 70

Table 5.3: The list of candidate classifiers with their respective accuracy 73

Table 5.4: Specifications of the GPUs employed in this research 81

Table 6.1: Classification accuracy results for discriminating between benign and malignant lesions from ultrasound images, using the PNN classifier, two sub-optimal and one optimal features-selection methods, and the LOO evaluation method 86

Table 6.2: Classification accuracy results for discriminating between benign and malignant lesions from DM images, using the PNN classifier, two sub-optimal and one optimal features-selection methods, and the LOO evaluation method 87

Table 6.3: Classification accuracy results for discriminating between benign and malignant lesions using combined features from ultrasound (US) and digital mammography (DM) images, and employing the PNN classifier, two sub-optimal and one optimal features-selection methods, and the LOO evaluation method 88

Table 6.4: Truth table of GPU results using PNN, exhaustive search, and LOO for features from US images, features from DM images, and combined features from US and DM images 89

Table 6.5: Classification accuracies in discriminating benign from malignant lesions, achieved by employing the ECV method on GPU, and by designing the PNN classifier by combined features from ultrasound and digital mammography features............ 93

Table 6.6: ECV Wilcoxon-rank-features Classification accuracies in discriminating benign from malignant lesions, achieved by employing the ECV method on GPU, and by designing the PNN classifier by combined features from US and DM features 94

Table 6.7: ECV SBS Classification accuracies in discriminating benign from malignant lesions, achieved by employing the ECV method on GPU, and by designing the PNN classifier by combined features from US and mammography features 94

Table 6.8: Comparative performance assessment varying the total number of features in the dataset (all time measurements have been averaged over 10 runs) . 95

Abbreviations

Abbreviation	Description
API	Application Programming Interface
BC	Breast Cancer
BI-RADS	Breast Imaging Reporting and Data System
CAD	Computer-Aided Diagnosis
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CV	Cross validation
DCIS	Ductal Carcinoma In Situ
DICOM	Digital Imaging and Communications in Medicine
DM	Digital Mammography
DMA	Direct Memory Access
DSS	Decision Support System
ECV	External Cross Validation
FLOPS	FLoating-point Operations Per Second
GLCM	Grey Level Co-occurrence Matrix
GPGPU	General Purpose Computation on Graphics Processing Units
GPU	Graphics Processing Unit
IDC	Invasive (or infiltrating) Ductal Carcinoma
IDE	Integrated Development Environment
JAI	Java Advanced Imaging
k-NN	k-Nearest Neighbour

LCIS	Lobular Carcinoma In Situ
LDA	Linear Discriminant Analysis
LOO	Leave One Out
LSO	Leave Some Out
MFC	Medial Femoral Condyles
mIS	micro-ischemic
MPI	Message Passing Interface
MRI	Magnetic Resonance Imaging
MS	Multiple Sclerosis
MTC	Medial Tibial Condyles
OOP	Object Oriented Programming
PNN	Probabilistic Neural Network
PR	Pattern Recognition
QLS	Quadratic Least Squares
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
SBFS	Sequential Backward Floating Selection
SBS	Sequential Backward Selection
SFFS	Sequential Forward Floating Selection
SFS	Sequential Forward Selection
SFU	Special Function Unit
SIMT	Single Instruction, Multiple Thread
SM	Streaming Multiprocessor

SP	Stream Processor
SVM	Support Vector Machines
TIFF	Tagged Image File Format
US	Ultrasound
WHO	World Health Organisation

Acknowledgements

I would like to thank my professors and colleagues and to express my deep gratitude. First, I would like to express my deepest appreciation to my supervisor, Professor *John Stonham,* for his encouragement and advice he has provided throughout my time as his student. I would also like to express my gratitude to Professor *Dionisis Cavouras* for his support and guidance during the course of my studies.

I would also like to thank *Dr. Dimitris Glotsos, Dr. Spyros. Kostopoulos*, and *Dr. Ioannis Kalatzis*, for the scientific support they provided by reviewing journal and conference papers, that came as offspring of this study, and by influencing my thinking in approaching scientific writing.

Special acknowledgement must be reserved for Radiologist M.D. *Nikos Dimitropoulos*, for his collaboration in collecting medical data and for helping me understand the clinical problem at hand.

Finally, I would like to thank my parents and close friends for their support throughout this endeavour.

Publications

This is the complete list of conference and journal publications derived from the work performed for the purposes of this thesis.

- **Sidiropoulos, K.**, Glotsos, D., Kostopoulos, S., Ravazoula, P., Kalatzis, I., Cavouras, D., Stonham, J. "*Real time decision support system for diagnosis of rare cancers, trained in parallel, on a graphics processing unit*", (2012), Computers in Biology and Medicine, 42 (4), pp. 376-386, ISSN: 00104825.

- **Sidiropoulos, K.P.**, Kostopoulos, S.A., Glotsos, D.T., Athanasiadis, E.I., Dimitropoulos, N.D., Stonham, J.T., Cavouras, D.A. "*Multimodality GPU-based computer-assisted diagnosis of breast cancer using ultrasound and digital mammography images*", (2013), International Journal of Computer Assisted Radiology and Surgery, 8 (4), pp. 547-560, ISSN: 18616410.

- Solomou, E., Kostopoulos, S., **Sidiropoulos, K.**, Athanasiadis, E., Lavdas, E., Glotsos, D., Sakellaropoulos, G., Zampakis, P., Stonham, J., Cavouras, D. "*Designing a pattern recognition system on GPU for discriminating between patients with micro-ischaemic and multiple sclerosis lesions, using MRI images*", (2013), International Journal of High Performance Computing Applications, 27 (3), pp. 348-359, ISSN: 10943420.

- Kostopoulos, S., **Sidiropoulos, K.**, Glotsos, D., Athanasiadis, E., Boutsikou, K., Lavdas, E., Oikonomou, G., Fezoulidis, I.V., Vlychou, M., Hantes, M., Cavouras, D. "*Pattern-recognition system, designed on GPU, for discriminating between injured normal and pathological knee cartilage*", (2013), Magnetic Resonance Imaging, 31 (5), pp. 761-770, ISSN: 0730725X.

- Ninos, K., Kostopoulos, S., **Sidiropoulos, K.**, Kalatzis, I., Glotsos, D., Athanasiadis, E., Ravazoula, P., Panayiotakis, G., Economou, G., Cavouras, D. "*Computer-based image analysis system designed to differentiate between low-grade and high-grade laryngeal cancer cases*", (2013), Analytical and Quantitative Cytology and Histology, 35 (5), pp. 261-272, ISSN: 08846812.

Proceedings in international conferences:

- **Sidiropoulos K.**, Cavouras D., Pagonis N., Dimitropoulos N., Stonham J., "*Accelerating the Design of Probabilistic Neural Networks for Computer Aided Diagnosis on Mammography, Employing Graphics Processing Units*", 3rd International Conference on Experiments/Process/System Modelling/Simulation/Optimisation (EpsMsO), Athens, Greece, July 2009.

- Pagonis N., Cavouras D., **Sidiropoulos K.**, Sakelaropoulos G., Nikiforidis G., "*Improving The Classification Accuracy of Computer Aided Diagnosis through Multimodality Breast Imaging*", 3rd International Conference on Experiments/Process/System Modelling/Simulation/Optimisation (EpsMsO), Athens, Greece, July 2009.

Abstract

The aim of this research was to address the computational complexity in designing multimodality Computer-Aided Diagnosis (CAD) systems for characterising breast lesions, by harnessing the general purpose computational potential of consumer-level Graphics Processing Units (GPUs) through parallel programming methods. The complexity in designing such systems lies on the increased dimensionality of the problem, due to the multiple imaging modalities involved, on the inherent complexity of optimal design methods for securing high precision, and on assessing the performance of the design prior to deployment in a clinical environment, employing unbiased system evaluation methods.

For the purposes of this research, a Pattern Recognition (PR)-system was designed to provide highest possible precision by programming in parallel the multiprocessors of the NVIDIA's GPU-cards, GeForce 8800GT or 580GTX, and using the CUDA programming framework and C++. The PR-system was built around the Probabilistic Neural Network classifier and its performance was evaluated by a re-substitution method, for estimating the system's highest accuracy, and by the external cross validation method, for assessing the PR-system's unbiased accuracy to new, "unseen" by the system, data. Data comprised images of patients with histologically verified (benign or malignant) breast lesions, who underwent both ultrasound (US) and digital mammography (DM). Lesions were outlined on the images by an experienced radiologist, and textural features were calculated.

Regarding breast lesion classification, the accuracies for discriminating malignant from benign lesions were, 85.5% using US-features alone, 82.3% employing DM-features alone, and 93.5% combining US and DM features. Mean accuracy to new "unseen" data for the combined US and DM features was 81%. Those classification accuracies were about 10% higher than accuracies achieved on a single CPU, using sequential programming methods, and 150-fold faster. In addition, benign lesions were found smoother, more homogeneous, and containing larger structures.

Additionally, the PR-system design was adapted for tackling other medical problems, as a proof of its generalisation. These included classification of rare brain tumours, (achieving 78.6% for overall accuracy (OA) and 73.8% for estimated generalisation accuracy (GA), and accelerating system design 267 times), discrimination of patients with micro-ischemic and multiple sclerosis lesions (90.2% OA and 80% GA with 32-fold design acceleration), classification of normal and pathological knee cartilages (93.2% OA and 89% GA with 257-fold design acceleration), and separation of low from high grade laryngeal cancer cases (93.2% OA and 89% GA, with 130-fold design acceleration).

The proposed PR-system improves breast-lesion discrimination accuracy, it may be redesigned on site when new verified data are incorporated in its depository, and it may serve as a second opinion tool in a clinical environment.

Chapter 1

Introduction and motivation

1.1 Overview

This research harnesses the general purpose computational potential of consumer-level Graphics Processing Units (GPUs) aiming to address the computational complexity in designing multimodality Computer-Aided Diagnosis (CAD) systems, for breast lesions characterisation, by employing optimal classifier training and unbiased system evaluation methods. The following chapter provides an introduction to the research performed, by first defining the problem it addresses and by describing its motivation and rationale. Additionally, the research aims and contribution are clearly identified, while, at the end of the chapter, the structure and organisation of this thesis are outlined.

1.2 Aims and Rationale

Breast cancer is the most common malignancy among women, both in the developed and developing countries (Laine et al., 1996; WHO, 2013). Although death rates have been decreasing because of early detection, the global death burden of breast cancer is estimated to 508,000 women in 2011 alone (WHO, 2013). Early detection of breast cancer is considered of paramount importance as it significantly increases the chances for successful treatment. This is mainly due to the fact that breast malignancies, detected during their early stages, are more likely to be smaller in size and still confined to the breast, while malignancies detected at more advanced stages, because they are causing symptoms, tend to be larger and are more likely to have metastasized, rendering treatment less likely to succeed (Hayes, 2007; Tabar and Dean, 2008).

In addition to self-examination, medical imaging constitutes an important asset assisting in the early diagnosis of breast cancer. A number of imaging modalities, including Digital Mammography (DM), Ultrasound breast examination (US), and Magnetic Resonance Imaging (MRI), has been employed in breast cancer detection (Tang et al., 2009). Although DM plays a crucial role in early breast cancer detection, it has been claimed that over 20% of breast lesions are missed (Humphrey et al., 2002; Majid et al., 2003) while an accountable number of biopsies is still performed on benign lesions (Bird et al., 1992; Beam et al., 1996; Harms, 1999; Evers, 2001). DM’s fundamental limitation is that non-calcified breast cancers are often obscured by surrounding and overlying dense parenchyma (Berg et al., 2008).

Consequently, alternative medical imaging modalities such as US and MRI are employed in a complementary manner, especially since they involve non-ionizing radiation and they do not burden the patient with radiation dose. Moreover, according to (Sardanelli et al., 2007; Lee et al., 2009; Grunberg and Domingo, 2011; Houssami and Ciatto, 2011) the combination of information from different imaging modalities may increase the radiologists’ diagnostic accuracy in breast cancer. Special interest, as complementary to DM, has been placed on US breast examination, since it is of low cost and it has been additionally shown to be of value as an adjunct method to DM examination due to the fact that it has the potential to depict early, node-negative breast cancers, not seen on DM, and its performance is improving in dense parenchyma (Laine et al., 1996; Berg et al., 2008).

Recently, computer-aided diagnosis (CAD) has been embedded in the daily clinical routine as several studies (Freer and Ulissey, 2001; Gur et al., 2004; Birdwell et al., 2005; Cupples et al., 2005; Dean and Ilvento, 2006; Morton et al., 2006) suggest that utilisation of CAD systems seems to increase the detection rates of breast cancer (Doi, 2007). In addition, a few studies have employed computer aided analysis for combining information from different imaging modalities as an aid to the discrimination between malignant and benign breast lesions. It has been shown that the diagnostic precision of CAD systems has increased by combining image information from different breast imaging modalities, as compared to using information from individual modalities. Yuan Y et al (Yuan et al., 2010) have shown the value of multimodality computer aided cancer diagnosis by combining lesion features from full field digital mammography and contrast enhanced MRI to discriminate between benign and malignant breast lesions; classification accuracy increased when features from both modalities were combined as compared to single-modality accuracies. Karen Drukker et al (Drukker et al., 2005) have developed a computer-based system by combining features from mammography and ultrasound to improve system classification precision in discriminating between benign and malignant lesions. In another study by Berkman Sahiner et al (Sahiner et al., 2009), a system was designed that combined data acquired from 3D US and X-ray mammography as an aid to the radiologists' performance in discriminating malignant from benign masses. Horsch K et al (Horsch et al., 2006) proved the value of multimodality computer aided diagnosis systems in improving the radiologist's diagnostic accuracy in the task of differentiating between malignant and benign breast lesions using mammography and sonography.

Nevertheless, in those studies, the design of CAD systems was based solely on the processing power of CPUs. It must be noted that one of the challenges encountered during the design of a CAD system is the time required to optimally train the pattern recognition system which typically lies in its core. Hence, designing of a classification scheme on a normal computer may take hours, or even days, while, once designed, the characterisation of a case takes infinitesimal time. Accordingly, in all the aforementioned studies, the design of multimodality CAD systems was far more challenging. The main problem was that they had to deal with increased image information, since more than one modality was involved. As a result, some had to employ sub-optimal system design methods for achieving manageable processing times for CAD design, thus resorting to feature reduction methods prior to training the classifiers by evaluation biased methods, such as the leave-one-out method (LOO) (Theodoridis and Koutroumbas, 2003). Consequently, these compromises in the CAD systems' design may have provided biased estimates of the classifier's performance. Accordingly, one of the solutions that have been proposed in tackling the CPU's processing power limitations is by employing parallel processing methods, typically involving powerful supercomputers or computer clusters. Unfortunately, this kind of hardware is prohibitively expensive and therefore accessible only to few people. However, a new promising development in this regard is the emergence of consumer-level Graphics Processing Units (GPUs) as a mainstream computing platform (Xu and Mueller, 2007).

Over the past few years, GPUs have evolved from the traditional fixed-function 3D graphics pipelines used as image-synthesis devices, into powerful, programmable, highly parallel computing devices, becoming an increasingly popular tool in many research fields including image analysis. This dramatic shift was the inevitable consequence of consumer demand for videogames, advances in manufacturing technology, and the exploitation of the inherent parallelism in the graphics pipeline (Luebke and Humphreys, 2007).

Today, graphics processing units constitute a low-cost, low-power (watts per flop) very high performance alternative to conventional microprocessors. For example, back in 2006, a GeForce 8800 GTX with a theoretical peak 520 GFLOPs (1 GFLOP equals 1 billion floating point operations per second), and dissipating 150 watts, cost about $500. This was an order of magnitude faster than ordinary CPUs. In a more up-to-date comparison, a GeForce 580 GTX with the same release price had a theoretical peak performance of 1,581 GFLOPS. This cannot compare to the 96 GFLOPS of Intel's Xeon Westmere X5670 CPU that cost about $800.

Nevertheless, the use of GPUs for general purpose computations in various scientific fields did not begin to gain momentum until the introduction of specialised programming frameworks, such as Stanford University's BrookGPU language (Buck et al., 2003), NVidia's Compute Unified Device Architecture (CUDA) (Nvidia, 2014), Microsoft's AP (Microsoft, 2006), and University of Waterloo's Sh Embedded Metaprogramming language (McCool et al., 2002), which provided an easy way to harvest the GPU's tremendous parallel computation potential.

Previous studies have employed similar measures to tackle processing time demanding image processing procedures, such as implementations of neural networks (Oh and Jung, 2004), Support Vector Machines (Ohmer et al., 2005), K-Nearest Neighbour (Beliakov and Li, 2012), tomographic reconstruction algorithms (Xu and Mueller, 2007; Pang et al., 2011), image registration methods (Lapeer et al., 2010; Shams et al., 2010), and dose simulation (Santhanam et al., 2012)

The aim of this research is to harness the general purpose computational potential of consumer level GPUs, in order to address the challenging problem of designing multimodality CAD systems for breast lesions characterisation by optimal classifier training and unbiased system evaluation methods.

Thus, the objectives of this thesis are:

i) to utilise parallel processing software methods and multicore GPU architectures with purpose to accelerate the training of CAD systems,

ii) to combine multimodality imaging information in CAD systems, and

iii) to implement and evaluate a low cost and optimum design GPU-based CAD system that will increase the precision of characterising breast lesions employing ultrasound and digital mammography images.

1.3 Contribution

The contribution of this thesis is in the design of high precision CAD systems. In particular, by transferring computer processing on the powerful processors of the GPU and by employing parallel processing programming techniques it has been made possible:

i. to employ optimal classifier design methods, by searching exhaustively for best feature combinations in a large feature space, augmented by the contribution of two imaging modalities (Sidiropoulos et al., 2012; Sidiropoulos et al., 2013).

ii. to optimise the classifier parameters for best performance, since retraining of the classifier was possible due to significant reductions in processing times (Solomou et al., 2012).

iii. to estimate the true error rate of the system to unknown data, by means of the 10-fold external cross validation (ECV) method, which is computationally demanding, since it requires the retraining of the system a multiple of times. This provides a reliable assessment as to how such systems, once designed, would perform in a clinical environment when presented with new “unseen” by the system data (Kostopoulos et al., 2013; Ninos et al., 2013; Sidiropoulos et al., 2013).

iv. to propose a CAD system that can be adapted on site, when additional imaging data are made available, which is also a computationally challenging procedure (Sidiropoulos et al., 2012).

Although, GPU technology has been previously employed in studies on image processing and analysis (Xu and Mueller, 2007; Ruiz et al., 2009; Dai et al., 2010; Lapeer et al., 2010; Shams et al., 2010), however, there appears to be no previous studies that employ GPU technology to deal with the problem of optimally designing stand-alone evolving CAD systems. Such use of technology may offer engineers the platform for building systems with the ability to incorporate new verified data to their depository and remodel themselves on location.

1.4 Thesis Outline

This manuscript is organised into eight chapters. Chapter 2 focuses on the clinical problem of breast cancer. In particular, this chapter presents the fundamentals of breast anatomy and physiology and provides information regarding the epidemiology and the taxonomy of breast cancer. In addition, chapter 2 describes the main imaging modalities employed for the diagnosis of the disease.

Chapter 3 provides an introduction into the basic concepts and methods of statistical pattern recognition. Key ideas and the basic algorithms of all stages of pattern recognition, including feature extraction and reduction techniques, methods for classification and feature selection, along with evaluation approaches, are discussed.

The use of GPUs as parallel processing hardware accelerators is the topic of chapter 4. Hence, this chapter begins with the history and evolution of GPUs over the last 2 decades. Architectural differences between CPUs and GPUs are identified and the main reasons behind the ever growing popularity of GPU-accelerated computing and its limitations are analysed. This chapter also includes a description of the CUDA programming model, thus providing a theoretical foundation for the subsequent chapter.

Chapter 5 describes in detail and in a systematic way the steps followed throughout the whole CAD design procedure such as protocols followed and software designed for multimodality data collection, methods of feature generation and reduction, classifiers designed and tested for execution on GPU and in parallel using the CUDA framework, and evaluation methods employed for assessing system precision and acceleration of system design.

In chapter 6 results are presented regarding the achieved precision and design acceleration introduced by the employment of GPUs. Furthermore, the textural features involved in the selected feature combination are analysed as to the information they convey and their significance in medical diagnosis of breast cancer.

Chapter 7 presents the employment of the designed GPU-based CAD system to tackle other medical problems. These included classification of rare brain tumours, discrimination of patients with micro-ischemic and multiple sclerosis lesions, classification of normal and pathological knee cartilages, and separation of low from high grade laryngeal cancer cases.

Finally, chapter 8 summarises the findings of the present thesis, draws conclusions and indicates directions for future research.

1.5 Research Funding

The current thesis was funded by the Greek State Scholarship Foundation (I.K.Y.). The support of NVIDIA Corporation with the donation of GPU hardware, used for the purposes of this research, is also gratefully acknowledged.

Chapter 2

Breast Cancer

2.1 Overview

This chapter focuses on the clinical problem of breast cancer. In particular, the fundamentals of breast anatomy and physiology are presented and information regarding the epidemiology and the taxonomy of breast cancer is provided. In addition, the main imaging modalities employed for the diagnosis of the disease are described, along with their advantages and limitations.

2.2 Breast Anatomy and Physiology

The breast constitutes one of the distinctive features of all mammals, and its biological role is the production and delivery of milk to nourish a newborn. In humans, the breast is a mound of tissue overlying the pectoralis major and is common in both sexes. Although both males and females develop breasts from the same embryological tissues, women's breasts become far more prominent than those of men, mainly due to female sex hormones that promote breast development (e.g. estrogen). This disparity in size is one of the key differentiation points between male and female anatomy.

The female breast consists of fibro-glandular and fatty tissue, in ratios that vary depending on the age and the genetic characteristics of the individual. In younger age, the female breast consists predominantly of fibro-glandular tissue, as fatty tissue is limited. In middle aged women, fibro-glandular and fatty tissues within the breast typically reach an equilibrium resulting in a 1:1 ratio, while in older ages and after menopause fibro-glandular tissue is gradually replaced by fatty tissue.

Each breast contains one mammary gland, which develops during pregnancy, remains active during lactation, and atrophies when a woman ceases to nurse. From a developmental perspective, mammary glands are modified sweat glands. The areola is the ring of pigmented skin located slightly below the centre of each breast, and surrounds a central protruding nipple. The latter can become erect when stimulated by tactile or sexual stimuli, or cold temperatures thanks to smooth muscle fibres controlled by the autonomic nervous system.

Inside the breast, each mammary gland comprises 15 to 25 lobes arranged radially around the nipple. Fibrous connective tissue and fat lie between the lobes, while suspensory ligaments, formed by this inter-lobar connective tissue, provide natural support for the breasts.

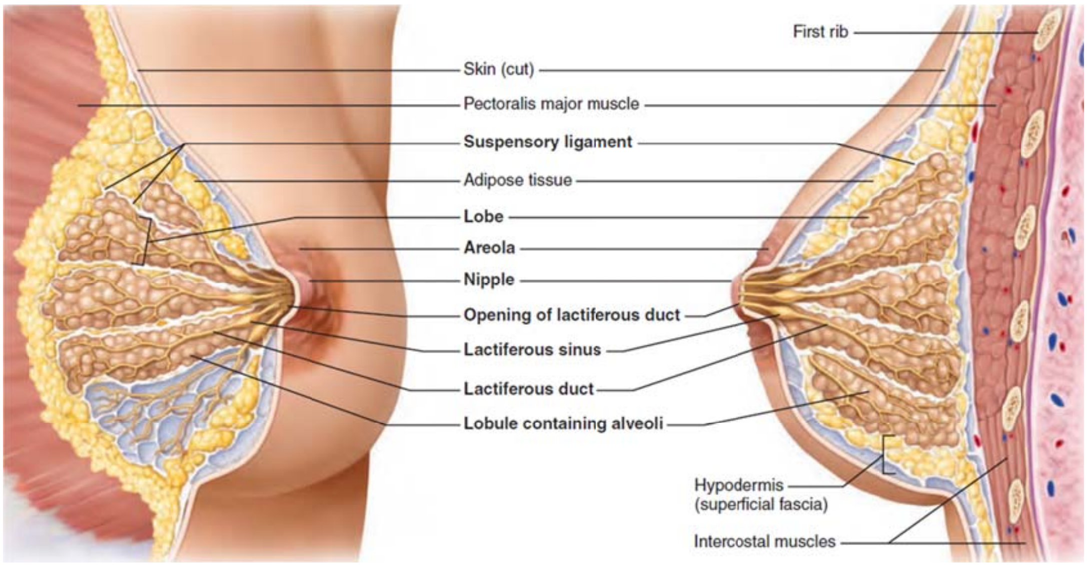


Figure 2.1: A graphical illustration of the female breast anatomy. Reproduced from Marieb and Hoehn (2013)

As illustrated in Figure 2.1, inside each lobe, there are smaller units called lobules. These lobules contain glandular alveoli, which in turn, produce milk, during lactation. Milk passes through the lactiferous ducts, which open to the outside at the nipple. Every lactiferous duct has a dilated region right under the areola, called lactiferous sinus, which is used to accumulate the milk. It must be noted that in non-pregnant women, the glandular structure of the breast is largely undeveloped and the duct system is rudimentary. For this reason, breast size is largely due to the amount of fat deposits (Martini et al., 2011; Marieb and Hoehn, 2013).

2.3 Breast Cancer Epidemiology

Breast cancer (BC) constitutes the most frequent malignant tumour among women and occurs when cells in the breast divide and grow without their normal control. Incidence rates, vary greatly worldwide, but suggest that 12.3% of women will be diagnosed with BC during their lifetime (SEER, 2011). Although mortality, presents a constant reduced tendency, BC constitutes the second cause of death from malignancies after pulmonary cancer in the USA and is still the first cause of death from malignancies among females between 20 and 59 years old. In UK alone, 49,936 women were diagnosed with invasive BC in 2011 (Clamp et al., 2003; GLOBOCAN, 2008; Jemal et al., 2008)

There are several types of BC. Ductal Carcinoma In Situ (DCIS) is considered non-invasive BC. In cases of DCIS the cells have not spread through the walls of the ducts into the surrounding breast tissue. However some DCIS cases can become invasive cancers. Lobular Carcinoma In Situ (LCIS) is not considered a true BC, despite its misleading name. LCIS indicate areas of abnormal cell growth within the breast tissue. This is an indication of high risk for developing BC in the future.

Invasive (or infiltrating) ductal carcinoma (IDC) is the most common type of BC, accounting for 75-80% of BC cases. IDC originally forms in milk ducts and infiltrates the fatty tissue of the breast through the duct walls. Then, using the lymphatic system and bloodstream IDC is able to metastasize to other parts of the body. Another invasive type of BC is the Invasive Lobular Carcinoma (ILC). Similar to IDC, ILC can metastasize to other parts of the body, but originally develops in the lobules of the breast.

2.4 Imaging Modalities for detection of Breast Cancer

As mentioned in the previous chapter, in addition to self-examination, medical imaging modalities such as Digital Mammography (DM), Ultrasound breast examination (US), and Magnetic Resonance Imaging (MRI), have been employed in the early diagnosis of BC (Tang et al., 2009).

2.4.1 Digital Mammography

DM (see Figure 2.2) is one of the most recent advances in x-ray mammography. Although it uses doses of ionizing radiation to create images, exactly like standard mammography, DM utilises specialised digital detectors and computers instead of the traditional x-ray film to form breast images. DM is considered the method of choice for the early diagnosis of breast carcinomas. Even so, over 20% of breast lesions are missed, as DM fails to detect non-calcified BCs that are often obscured by surrounding and overlying dense parenchyma, commonly found among women under the age of 50 (Humphrey et al., 2002; Majid et al., 2003). Hence, invasive cancers, often manifested as non-calcified masses, cannot be detected by DM, particularly in cases of dense surrounding tissue (Berg et al., 2008). As a result, in many cases, US and MRI can be employed in a complementary manner and detect BCs that may not be visible with DM.

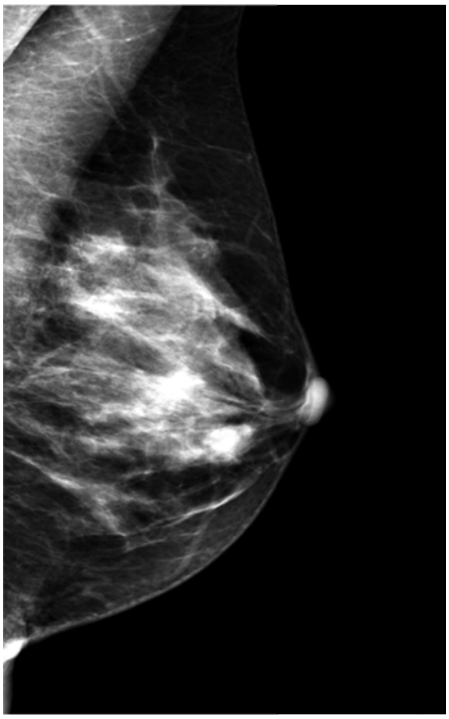


Figure 2.2: A typical DM image

2.4.2 Ultrasound

In breast US examinations, ultrasonic sound waves at a frequency of 8-16MHz are sent via a small handheld probe to move through the skin. As these sound waves travel through the breast, they bounce off the tissues and return to the probe just like an echo. The timing and intensity of these reflected waves are then used to form a detailed image of the breasts (see Figure 2.3). Due to its benefits, special interest has been placed on US breast examination.

US breast examination is painless for the patient, widely available, and less expensive than other imaging methods. In addition, US does not use any ionizing radiation and, thus, it does not pose any health danger, allowing pregnant women to use it. In contrast to DM, US provides a clear picture of dense tissues and it is the only non-invasive method to determine whether a breast mass is cystic or solid (Kossoff, 2000). Therefore, US has the potential to depict early, node-negative breast cancers not seen on mammography, and its performance is improving in dense breasts.

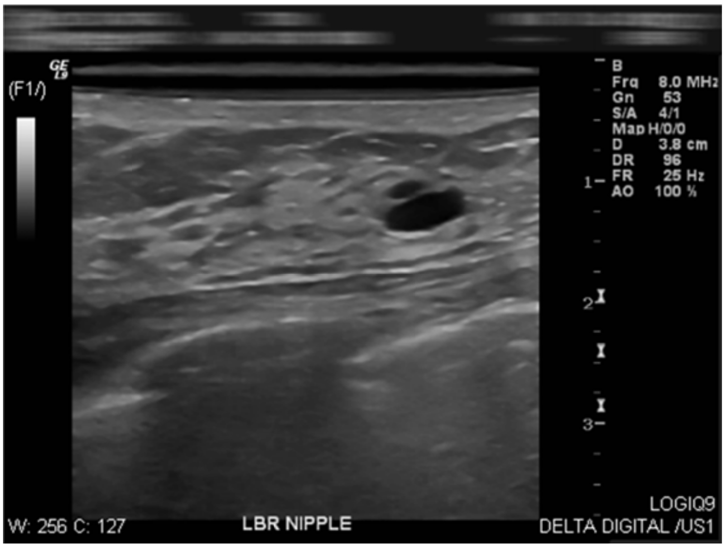


Figure 2.3: A typical US image

However, many cancers cannot be detected via US examination. These cases include breast cancers that present calcifications, i.e. small areas of calcium in the breast, at their early stages, as they are not visible to US. Furthermore, studies have indicated that the use of US can increase the number of false-positives, i.e. erroneously identifying a potential area of concern as malignant, and consequently lead to more unnecessary and painful procedures, such as biopsies. For these reasons, US cannot replace DM (Berg et al., 2008).

2.5 Digital Mammography characterisation of breast masses

DM findings that are commonly associated with benign cases include smooth walled masses within the breast tissue, that feature a lucent center and the presence of scattered (not clustered) microcalcifications. In contrast, finding suggesting malignancy include masses with irregular borders and spiculated density. Additionally, the presence of clustered microcalcifications, typically 5 or more calcifications in 1 cm area, is also suggestive of malignancy and requires further evaluation.

2.6 Ultrasound characterisation of breast masses

Sonographic characteristics commonly found in benign breast lesions have been defined by several previous studies (Stavros et al., 1995; Stavros et al., 2004; Mainiero et al., 2005; Gokhale, 2009)

In summary, benign breast lesions appear with hyperechoic or isoechoic texture, smooth, well circumscribed, and gently lobulated shape with three or fewer lobulations. In addition, benign breast lesions present an elliptical shape, appearing wider that tall and feature a thin echogenic capsule.

On the other hand, hypoechoic breast lesions with ill-defined borders are characteristics (indications) of malignancy. In particular, malignant breast lesions commonly feature hypoechoic texture, spiculated margins with thick echogenic shape, and taller than wide shape. Microlobulations and calcifications are also indications of malignancy.

In Tables 2.1 and 2.2, representative lesions of benign and malignant cases used in the present study, employing US and DM of the same patient, are presented.

Table 2.1: Representative lesions of benign cases employing DM and US

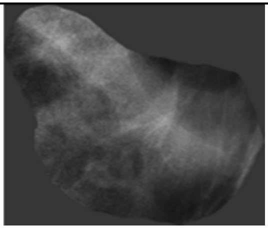
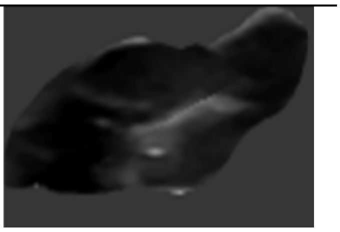
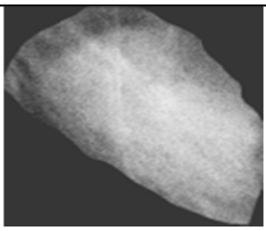
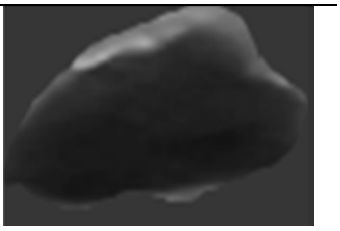
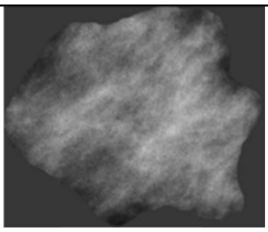
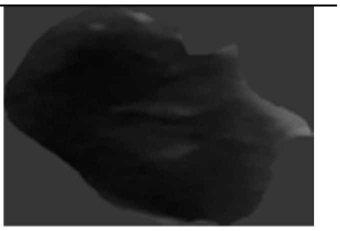
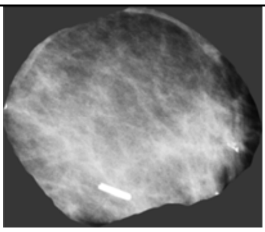
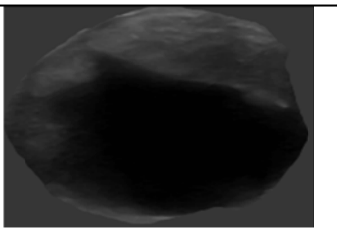
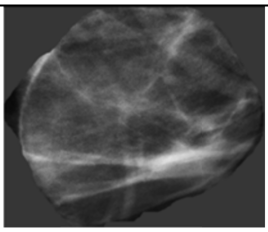
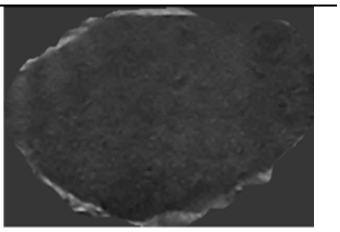
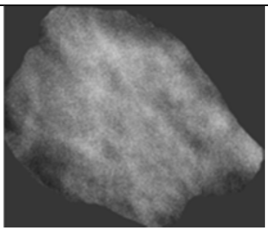
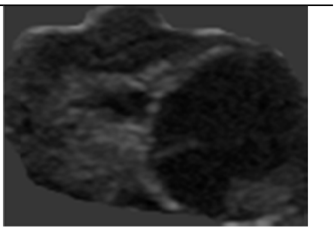
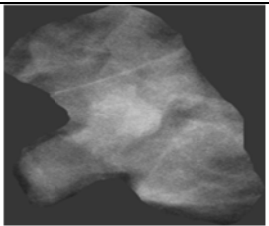
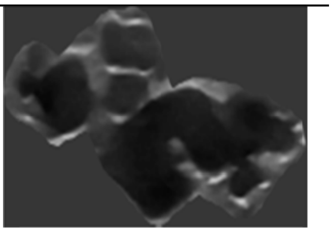
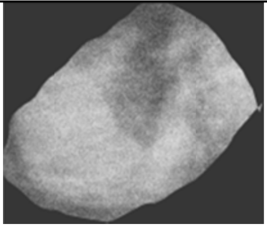
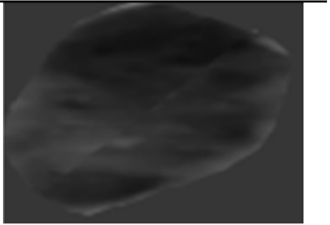
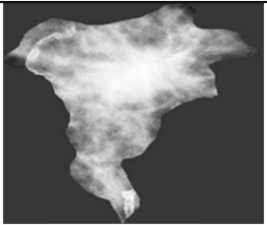
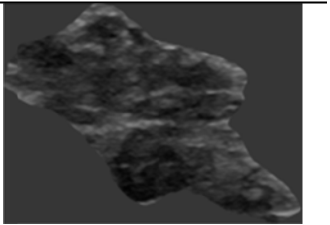
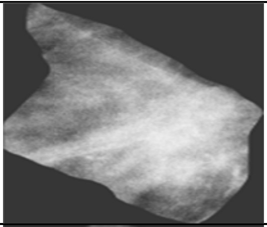
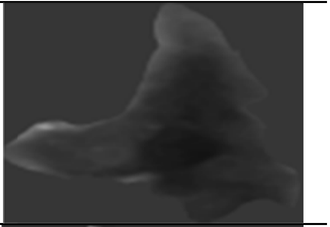
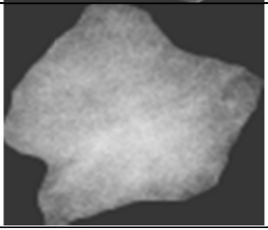
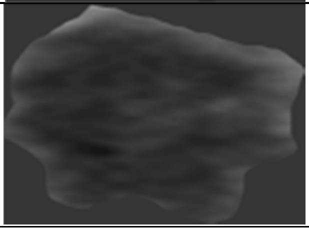
Benign Breast Lesions	Digital Mammography	Ultrasound
Adenolipoma		
Cyst		
Fibroadenoma		
Lipid Oil Cyst		
Lipoma		
Papilloma		

Table 2.2: Representative lesions of malignant cases employing DM and US

Malignant Breast Lesions	Digital Mammography	Ultrasound
Ductal Carcinoma In Situ		
Infiltrated Lymph Node		
Infiltrating Ductal Carcinoma		
Infiltrating Lobular Carcinoma		
Mucoid Carcinoma		

In the current study, the focus was on the texture of breast lesions on both DM and US images. The proposed CAD system combines textural information from both modalities to achieve higher precision.

2.7 Summary

Breast cancer constitutes the most frequent malignant tumour among women. There are several types of BC. Digital Mammography (DM), and Ultrasound breast examination (US), have been employed in the early diagnosis of BC. DM's fundamental limitation is that non-calcified breast cancers are often obscured in dense breast tissues. Thus, special interest has been placed on US, since it is of low cost and it has been shown to be of value as an adjunct method to DM examination due to the fact that it has the potential to depict early, node-negative breast cancers, not seen on DM, and its performance is improving in dense parenchyma.

Chapter 3

Introduction to Statistical Pattern Recognition

3.1 Overview

Recently, CAD has been embedded in the daily clinical routine as several studies suggest that utilisation of such systems seems to increase the detection rates of breast cancer. Typically, CAD systems are based on Pattern Recognition (PR) methods and employ a classifier to characterise breast lesions. This chapter provides an introduction into the basic concepts and methods of statistical pattern recognition. Key ideas and the basic algorithms of all stages of PR, including feature extraction and reduction techniques, methods for classification and feature selection, along with evaluation approaches, are discussed.

3.2 Basic principles of pattern recognition

As stated in the previous chapter, although, screening mammographic examination is currently the most effective tool for early detection of BC, the potential of diagnostic errors still remains substantially high in routine conditions. Failure to predict tumour's behaviour due to misdiagnosis, might lead to inadequate therapy affecting patient survival and to increased management costs. More specifically, in false-positive cases, noncancerous lesions can be misinterpreted as a cancer, and patients are needlessly subjected to biopsy, while in false-negative cases, cancers are missed leading to delays in administering medical care, with adverse effects to patient survival.

In order to enhance sensitivity of mammography, complimentary modalities such as US and MRI are often recommended to achieve additional information. Recently, CAD systems have been developed to assist radiologists in interpreting medical images and in differentiating between benign and malignant tissues.

Decision support systems, relying on statistical pattern recognition, have been shown as a promising solution in reducing diagnostic errors and in improving diagnostic concordances in breast lesions' characterisation. Thus they are of particular interest in this thesis.

Pattern Recognition (PR) refers to the scientific discipline which aims to assign objects to different categories, or classes (Theodoridis and Koutroumbas, 2003; Kuncheva, 2004). Objects are often referenced by the term patterns and, depending on the application, they can be images or signals or any type of measurements that need to be classified. Patterns are described by characteristics called features. In particular, a set of features is used to form a feature vector. Every single pattern can be uniquely identified by its feature vector. The task of a classifier is to draw a decision line and partition the feature space into regions that belong to each category, or class. In order to perform this division, the classifier is based on a criterion, known as the prediction rule, and to a set of patterns (training patterns) whose true class is known. Once completed, this process, denoted by the term classifier training, enables the classifier to take as input a new pattern and categorise it under one of the available classes. However, the classifier's ruling is not necessarily correct. There are cases where patterns are erroneously classified, or misclassified. Thus, there are methods for assessing the performance of a designed classifier.

As illustrated in Figure 3.1, the backbone of such a system comprises the following stages:

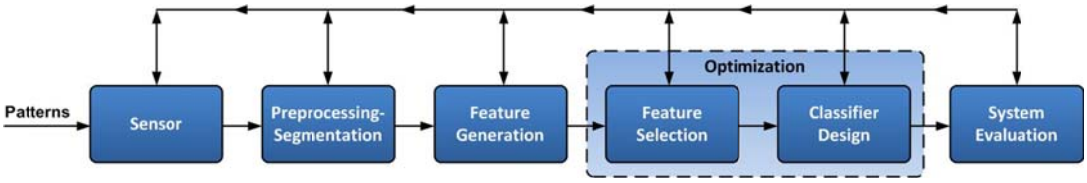


Figure 3.1: The basic stages involved in the design of a classification system. Adapted from Theodoridis and Koutroumbas (2003)

Pre-processing-Segmentation: This stage concerns the application of methods for image conditioning (e.g. image enhancement) and image segmentation, for extracting a region of interest (ROI). ROIs are formed by separating the information-rich suspicious area (lesion) from the information-poor (surrounding tissue) image regions.

Feature extraction: The extracted ROIs are then described in the form of quantitative numerical values, the so called 'features', which characterise individual patterns and describe object attributes.

Feature reduction-selection: Features dimensionality is reduced to balance computational complexity. Features that provide highest discrimination accuracy are considered most informative and are, therefore, selected to be used in the design of the PR system.

Classification: This stage concerns the construction of a prediction rule, able to optimally classify data into specific classes, based on most informative features selected in the previous stage.

System Evaluation: Once the prediction rule is constructed, its accuracy/performance is assessed in order to get an estimate of the probability of correct classifications in both known and, most importantly, unknown objects. Each of the aforementioned stages is illustrated in detail in the following paragraphs.

3.3 Segmentation

Generally referred to as the image partitioning into non-overlapping regions segmentation is an unavoidable step for every medical image analysis problem. It is considered a non-trivial problem and it is a subject of on-going research activity, especially in medical imaging applications such as magnetic resonance imaging, ultrasound, X-ray computed tomography, nuclear medicine, and microscopy imaging. Many segmentation methods are readily available, especially in research software tools such as Matlab, however, in our case segmentation was performed manually, by the physician, using a software tool that was specially built for the purposes of this research. The software tool is described in chapter 5. It provided us with a secure level of confidence that lesion outlines would be delineated with accuracy by the expert.

3.4 Feature extraction

In this stage the goal is to generate features, which quantify image properties from delineated image ROIs. It has been long shown that texture of medical images can be used to encode useful diagnostic information, suitable for distinguishing tissues into clinical meaningful classes (e.g. benign from malignant).

Image texture may be described as the spatial arrangement of pixels, which gives an image a distinct pattern, such as coarse, smooth, granular. Similar descriptions are also used by radiologists for diagnosis. A number of features have been proposed in previous studies (Haralick et al., 1973; Galloway, 1975; Gose et al., 1996; Gonzalez and Woods, 2002; Theodoridis and Koutroumbas, 2003) that quantify the image texture of segmented ROIs. Other features have also been proposed in literature, such as those regarding the morphology or shape of the lesions (Mavroforakis et al., 2005).

The basic theory of textural measures, employed for the purposes of the present thesis, is given below.

3.4.1 Histogram features

The histogram of an image shows the distribution of all image pixels according to their intensity value. Its shape provides information about the pixel content of the image. Based on first order statistics, that do not consider neighbour pixel relationships, informative features extracted from image histogram are (Theodoridis and Koutroumbas, 2003):

Mean value (*MV*)

$$MV = \frac{\sum_i \sum_j I(i,j)}{N} \tag{3.1}$$

where $I(i,j)$ is the pixel intensity in position (i,j) and N the total number of pixels. It gives the average intensity of the ROI.

Standard deviation (*SD*)

$$SD = \sqrt{\frac{\sum_i \sum_j \left[I(i,j) - MV\right]^2}{N}} \tag{3.2}$$

The standard deviation describes the variation of the image grey levels from the mean value. It is a measure of variation in pixel intensity.

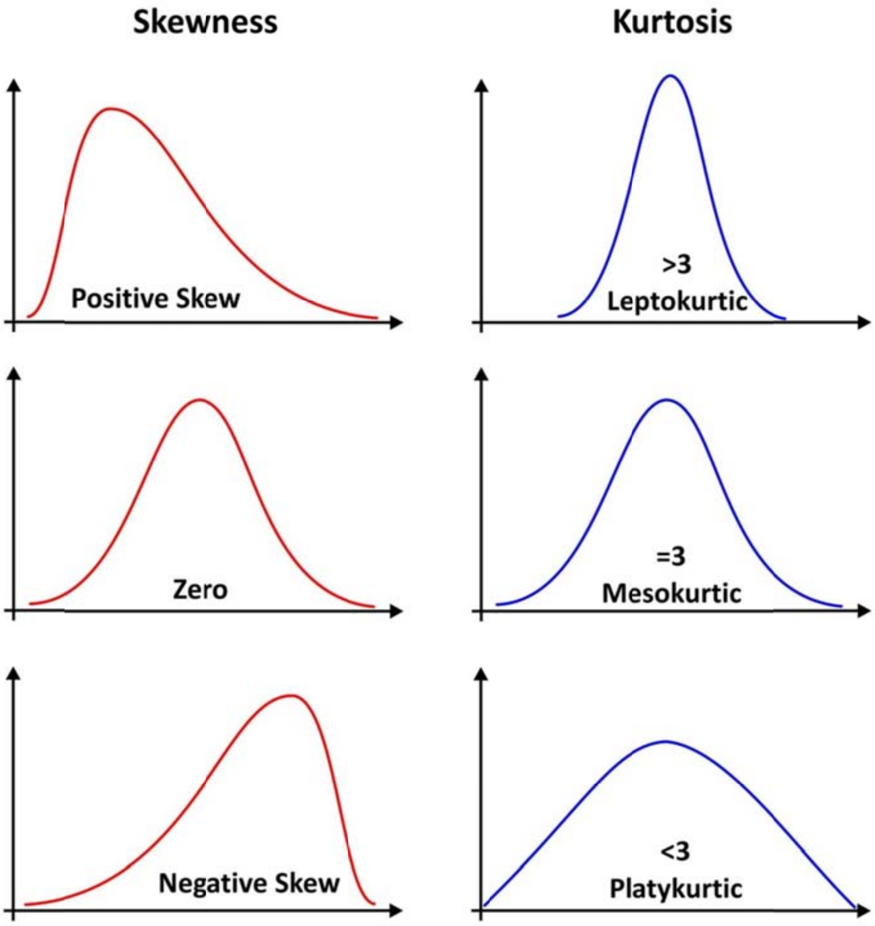


Figure 3.2: Examples illustrating how Skewness and Kurtosis are indicative of the histogram's shape

Skewness (*SKEW*)

$$SKEW = \frac{1}{N}\frac{\sum_i \sum_j [I(i,j) - MV]^3}{SD^3} \tag{3.3}$$

As illustrated in Figure 3.2, skewness describes the distribution asymmetry around the mean value. Positive skewness indicates predominance of darker (than mean intensity) pixel values.

Kurtosis (KURT)

$$KURT = \frac{1}{N}\frac{\sum_i \sum_j [I(i,j) - MV]^4}{SD^4} \tag{3.4}$$

Kurtosis is indicative of the histogram shape and describes the distribution sharpness compared to the normal distribution (see Figure 3.2). Wider shapes signify broader pixel intensity distributions.

3.4.2 Features calculated from the Co-occurrence Matrix

The grey level co-occurrence matrix (GLCM), has been proposed by Haralick (Haralick et al., 1973) for textural feature extraction. The GLCM is a two dimensional matrix $P(i,j)$ that presents the frequency of appearance of two neighbour pixels, having grey levels i, j respectively and interpixel distance d at an angle $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. Thus a total of 4 GLCMs, one for every angle, is calculated. This is illustrated in Figure 3.3. It must be noted that, for the purposes of this research, the interpixel distance d was set to 1 pixel.

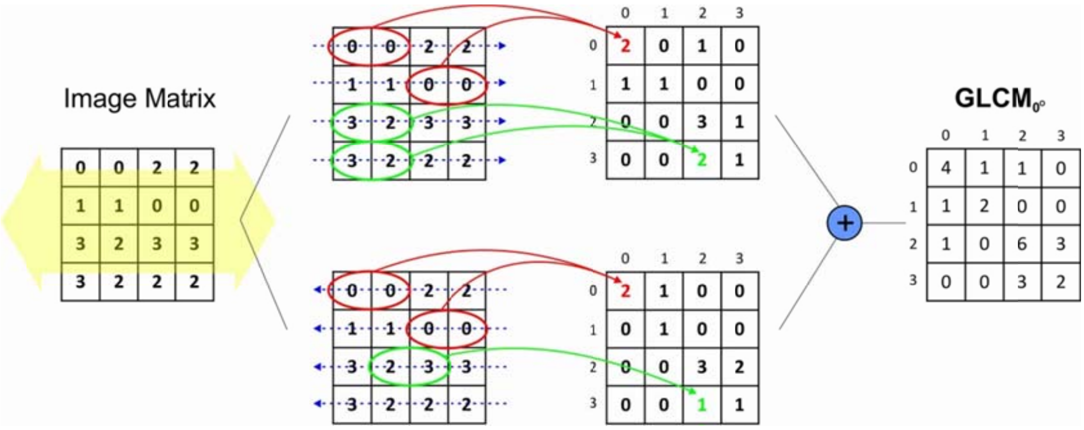


Figure 3.3: Calculation of GLCM at angle 0°

After their calculation, the frequency values of every GLCM are normalised. Thus, for every (i,j) element, the normalised frequency $p(i,j)$ is calculated by dividing $P(i,j)$ by the total number of neighbouring cell pairs R in the respective angle (Haralick et al., 1973). This is described in Equation 3.5.

$$p(i,j) = \frac{P(i,j)}{R} \tag{3.5}$$

For example, considering the GLCM of the horizontal direction ($\theta = 0^{\circ}$) there are $2(Columns - 1)$ neighboring cell pairs for each row of the matrix, providing a total of $R_{0^{\circ}} = 2Rows(Columns - 1)$ horizontal neighbouring pairs.

Angular Second Moment (ASM)

$$ASM = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j)^2 \tag{3.6}$$

where N_g is the number of grey levels in the image, and $p(i,j)$ is the (i,j)th element of the normalised spatial dependence matrix. ASM measures the homogeneity of the original image.

Contrast (CONTR)

$$CONTR = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \left(p(i,j)\right) \right\}, |i-j| = n \tag{3.7}$$

which measures the amount of local intensity variations in the image

Inverse Difference Moment (IDM)

$$IDM = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{p(i,j)}{1+(i-j)^2} \tag{3.8}$$

which is a measure of image homogeneity; larger values for smaller grey-tone differences.

Entropy (ENTR)

$$ENTR = -\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j) \log\left(p(i,j)\right) \tag{3.9}$$

Entropy measures the levels of disorder and randomness in the original image. Its value is maximised for a flat GLCM (Bocchi et al., 1997).

Correlation (COR)

$$COR = \frac{\sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}(ij)\,p(i,j) - m_x m_y}{\sigma_x \sigma_y} \tag{3.10}$$

where m_x, m_y, σ_x and σ_y the respective mean values and standard deviations of p_x and p_y, which are described as follows:

$$p_x(j) = \sum_{j=1}^{N_{rows}} p(i,j) \tag{3.11}$$

$$p_y(j) = \sum_{j=1}^{N_{columms}} p(i,j) \tag{3.12}$$

Correlation quantifies the level of linear dependencies among the grey tones of the image.

Autocorrelation (ACOR)

$$ACOR = \sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} ij\,p(i,j) \tag{3.13}$$

ACOR measures the energy spread around the diagonal of the co-occurrence matrix, which in effect gives an assessment of the correlation amongst intensities of pixel pairs. It attains high values when intensities in pixel pairs are very close (Pratt, 2007).

Sum of Squares (SOQ)

$$SOQ = \sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1} (i - MV)^2\, p(i,j) \tag{3.14}$$

SOQ is a measure of variance of grey levels and, thus, quantifies textural dissimilarity.

Variance (VAR)

$$VAR = \sum_{i=0}^{N_g-1}\sum_{j=0}^{N_g-1}(i-m)^2 p(i,j) \tag{3.15}$$

where m is the mean value of the GLCM.

Sum Average (*SAV*)

$$SAV = \sum_{i=2}^{2N_g} i p_{x+y}(i) \tag{3.16}$$

SAV assesses grey-level inhomogeneity. It attains higher values when there are many differences at high grey levels, suggesting the existence of structures at higher grey tones.

with $$p_{x+y}(k) = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j), i+j=k, k=2,3,\ldots,2N_g$$

Sum Entropy (*SENTR*)

$$SENTR = -\sum_{i=2}^{2N_g} p_{x+y}(i)\log\left(p_{x+y}(i)\right) \tag{3.17}$$

SENTR is a measure of randomness or lack of order in the distribution of structures throughout the grey-tone values. High values signify equally distributed structures within the grey-tone values of the image.

Sum Variance (*SVAR*)

$$SVAR = \sum_{i=2}^{2N_g}(i-SENTR)^2 p_{x+y}(i) \tag{3.18}$$

SVAR expresses the variance in the distribution of structures throughout the grey-tone values.

Difference Variance (DVAR)

$$DVAR = variance\ of p_{x-y} \tag{3.19}$$

where $$p_{x-y}(k) = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j), |i-j|=k, k=0,1,2,\ldots,N_g-1$$

DVAR is a measure of variation in image contrast. It attains low values for equally distributed contrast transitions or image grey-tone differences.

Difference Entropy (DENTR)
$$DENTR = -\sum_{i=0}^{N_g-1} p_{x-y}(i)\log\left(p_{x-y}(i)\right) \tag{3.20}$$

DENTR is a measure of randomness or lack of structure in the image contrast. Attains high values for equally distributed image grey-tone differences or for images with low variation in image contrast.

3.4.3 Features Extracted from the Run-Length Matrix

A set of consecutive, collinear picture points having the same grey level value is a grey level run. The number of picture points in the run is defined as its length. For a given picture, we can compute a grey level run length matrix for runs having any given direction.

Let $Q_{RL}(i,j)$ denote the (i,j)-element of the run-length matrix (Galloway, 1975), where i is the grey-level and j is the maximum number of consecutive pixels with grey-level equal to i at an angle θ. In the present work, the run-length matrices for angles $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ were calculated. From the run-length matrix the following features are defined:

Short-Run Emphasis (SRE)
$$SRE = \frac{\sum_{i}^{N_g}\sum_{j}^{N_r}\frac{Q_{RL}(i,j)}{j^2}}{\sum_{i}^{N_g}\sum_{j}^{N_r}Q_{RL}(i,j)} \tag{3.21}$$

where N_g the number of grey levels in the image, N_r the number of different run lengths that occur. Short-Run Emphasis measures the distribution of short runs of pixels with the same grey level, hence, fine textures result to higher SRE values, while coarse textures present lower SRE values (Galloway, 1975).

Long-Run Emphasis (LRE)

$$LRE = \frac{\sum_{i}^{N_g}\left(\sum_{j}^{N_r} Q_{RL}(i,j)\cdot j^2\right)}{\sum_{i}^{N_g}\sum_{j}^{N_r} Q_{RL}(i,j)} \tag{3.22}$$

In contrast to SRE, Long-Run Emphasis quantifies the presence of long runs and larger structures in the image. Therefore fine textures result to lower LRE values, while coarse textures present higher LRE values (Galloway, 1975).

Grey-Level Non-Uniformity (GLNU)

$$GLNU = \frac{\sum_{i}^{N_g}\left(\sum_{j}^{N_r} Q_{RL}(i,j)\right)^2}{\sum_{i}^{N_g}\sum_{j}^{N_r} Q_{RL}(i,j)} \tag{3.23}$$

GLNU measures the similarity of grey level values. Lower values suggest an even distribution of run lengths throughout the grey levels (Galloway, 1975).

Run-Length Non-Uniformity (RLNU)

$$RLNU = \frac{\sum_{j}^{N_r}\left(\sum_{i}^{N_g} Q_{RL}(i,j)\right)^2}{\sum_{i}^{N_g}\sum_{j}^{N_r} Q_{RL}(i,j)} \tag{3.24}$$

Accordingly, RLNU measures the similarity of length of runs, and thus lower values suggest an even distribution of runs throughout the run-length groups (Galloway, 1975).

Run-Percentage (RP)

$$RP = \sum_{i}^{N_g}\sum_{j}^{N_r} Q_{RL}(i,j)/P \tag{3.25}$$

where P the maximum number of run-lengths in the image. In particular, provided that all runs had a length of one pixel, P matches the total number of pixels in the image. Run Percentage quantifies the distribution of runs, as higher RP values indicate the prevalence of short runs in the texture of the original image. Consequently, RP attains its lowest value for pictures with linear structure (Galloway, 1975).

3.5 Data Normalisation

Following the feature extraction, generated feature values usually vary in different dynamic ranges. Hence, features with higher values have a stronger impact on the design of the classifier than features with lower values. In order to counteract this, the intermediate step of data normalisation is often adopted prior to the classifier design. The main function of data normalisation is to adjust all features values so that they fall within a predefined range (Theodoridis et al., 2010).

A quite common linear technique dictates that all features in each dataset are normalised to zero mean and unit standard deviation, according to Equation 3.26.

$$x'(i) = \frac{x(i) - \mu}{\sigma} \tag{3.26}$$

where *x′(i)* is the normalised version of feature *x(i)*, μ and σ are the mean and standard deviation of feature vector *x*, both calculated over all patterns of all classes. Data normalisation introduces an additional computational burden that will be discussed and addressed during the design of the GPU-based system.

3.6 Feature reduction – selection

As stated afore, the main purpose of this stage is to search through a large pool of available features and select a subset of features that exhibits high discriminatory potential. The importance of this stage in the design of the classification system cannot be overstressed, as a poor selection of features has a negative impact on the performance of the classifier. Besides minimising the computational complexity involved, reducing the number of features by selecting a smaller, yet more informative subset, contributes to avoiding overfitting to the specific training dataset and, thus, results in improved generalisation performance, i.e. the classifier's ability to perform well on unseen data. There are two main approaches to feature selection.

3.6.1 Scalar Feature Selection

The first step examines every available feature individually and determines whether it is an informative one. This is typically performed by employing statistical tests, such as the t-test, the Wilcoxon rank sum test (Hollander and Wolfe, 1999), or the Fisher ratio, to examine whether there is a statistically significant difference in the feature's value among the classes. Features can be ranked according to their statistical difference, while in case there is no statistically significant difference the feature under examination may be discarded. Other class separability criteria include the Receiver Operating Characteristic (ROC) curve, Bhattacharyya distance, and measures based on Scatter matrices (Theodoridis and Koutroumbas, 2003).

Although computationally simple, the main drawback of these methods is that they do not take into consideration possible correlations among the features. According to (Theodoridis and Koutroumbas, 2003), a better approach to scalar feature selection is to incorporate the cross-correlations between the features. First, by employing one of the aforementioned criteria, features are ranked in descending order. Let f_1 denote the index of the top ranked feature. Next, to select the index f_2 of the second most important feature, Equation 3.27 is used.

$$f_2 = \max_j \left\{ w_1 C_j - w_2 \left| \rho_{f_1,j} \right| \right\}, \qquad j \neq f_1 \tag{3.27}$$

where C_j is the value of the criterion for the jth feature, and $\rho_{f_1,j}$ the cross correlations between f_1 and any other feature $j \neq f_1$, and w_1 and w_2 two user defined weights. The cross-correlation coefficient between any two features is given by Equation 3.28

$$\rho_{ij} = \frac{\sum_{n=1}^{N} x_{ni} x_{nj}}{\sqrt{\sum_{n=1}^{N} x_{ni}^2 \sum_{n=1}^{N} x_{nj}^2}} \tag{3.28}$$

where N is the number of patterns, x_{ni} the ith feature of the nth pattern and x_{nj} the jth feature of the nth pattern

Similarly, the rest of the features are selected by considering the average correlation with all the previously selected features, according to Equation 3.29

$$f_k = \max_j \left\{ w_1 C_j - \frac{w_2}{k-1} \sum_{r=1}^{k-1} \left| \rho_{f_r, j} \right| \right\}, \qquad j \neq f_r, \qquad r = 1, 2, \ldots k-1 \tag{3.29}$$

for $k = 3, 4, 5 \ldots M$ (the number of features)

The main advantage of scalar feature selection, i.e. evaluating the discriminatory potential of each individual feature, is that it is computationally non-demanding. Nevertheless, this comes at a cost, as scalar feature selection methods may not be suitable for more complex classification problems where features with high mutual correlation may be used (Theodoridis and Koutroumbas, 2003).

3.6.2 Feature Vector Selection

Feature vector selection methods search through candidate feature combinations, evaluating their classification capability and aiming to identify the “best” feature combinations, based either on an optimality rule (Filter methods) or on the performance of a classifier (Wrapper methods).

Filter selection methods evaluate each feature combination by its information content, typically employing one of the separability criteria mentioned afore, such as interclass distance, correlation measures, etc. These methods are fast in execution and produce results that exhibit more generality to a larger family of classifiers (Theodoridis and Koutroumbas, 2003).

On the other hand, in wrapper selection methods the feature selection criterion is bound together with the construction of the prediction rule (the classifier). Hence, wrapper methods embed a pattern classification scheme in order to evaluate each feature combination by its predictive accuracy (recognition rate on test data) and select the combination that results in the minimum classification error probability. Wrapper feature selection techniques generally achieve better classification accuracy than filter methods, since they are fine-tuned to the specific interactions between the classifier and the dataset. However, common drawbacks of these methods include higher risk of overfitting than filter techniques, and high computational cost since the wrapper must train the classifier once or multiple times for each feature subset, depending on the evaluation method adopted (Saeys et al., 2007).

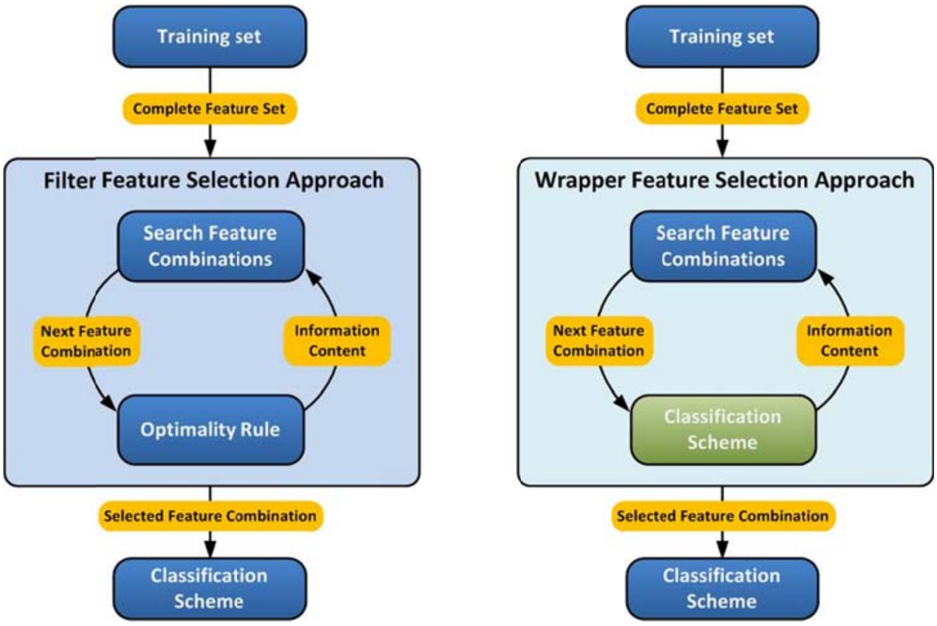


Figure 3.4: Filter and Wrapper feature selection approaches

Both of the aforementioned approaches can be used with optimal and suboptimal searching techniques.

3.6.2.1 Optimal Feature Searching

The optimal case of exhaustive search (Theodoridis and Koutroumbas, 2003), entails the formation and evaluation of all possible vector combinations of k features out of the n originally available. Hence the number of possible combinations, that should be evaluated, is given by

$$\binom{n}{k} = \frac{n!}{(n-k)!\,k!} \tag{3.30}$$

It must be pointed out that optimal feature selection via the exhaustive search poses a challenge in terms of computational complexity, even for moderate values of n and k. For example, exhaustive evaluation of 10 out of 20 features involves examining 184,756 feature subsets, while exhaustive evaluation of 10 out of 100 involves more than 17 trillion feature subsets (Devyver and Kittler, 1982). Additionally, in many classification problems, the value k (the number of elements in the selected feature vector) is not known a priori, and therefore, feature combinations for different values of k have to be examined. This further increases the number of feature combinations to be evaluated. Thus, the complexity of the exhaustive search can be described by $O(z)$ where,

$$z = \sum_{k=1}^{K} \binom{n}{k} = \sum_{k=1}^{K} \frac{n!}{(n-k)!\,k!} \tag{3.31}$$

As a result, an exhaustive evaluation of all combinations up to 10 out of 20 available features involves examining 616,665 feature subsets.

Because optimal feature selection via the exhaustive search method is characterised by high computational complexity, a number of efficient, computationally simple, yet suboptimal searching techniques have been suggested, particularly in cases where the number of available features is large.

3.6.2.2 Suboptimal Feature Searching

These techniques include, among others, the sequential forward and backward selection (SFS and SBS respectively), and the sequential forward and backward floating selection methods (SFFS and SBFS respectively) (Theodoridis et al., 2010).

Sequential Forward Selection (SFS) is the simplest greedy search algorithm. Starting from an empty set, the SFS algorithm determines an “optimal” set of features through a number of steps. Sequentially, in each step, SFS adds the feature f^+ that, when combined with the features Y_S, selected in previous steps, results in the highest value of the chosen separability criterion $C(Y_S+f^+)$. It must be noted that the latter can be any of the criteria mentioned afore or the predictive accuracy of a pattern classifier, depending on whether the filter or the wrapper approach is adopted. SFS performs best when the optimal subset has a small number of features.

Similarly to SFS, Sequential Backward Selection (SBS) determines an “optimal” set of features through a number of steps, but works in the opposite direction. The particular search begins with the full set of available features, and in every step it removes a single feature which when discarded leads to an improvement in the chosen criterion. SBS works best when the optimal feature subset has a large number of features (Theodoridis and Koutroumbas, 2003). Pseudocode for SFS and SBS is shown in Figure 3.5 and Figure 3.6 respectively.

```
1. Start with the empty feature set $Y_o = \{\emptyset\}$
                                    $s = 0$
2. Select the next best feature $f^+ = \mathrm{argmax}_{f \notin Y_s}[C(Y_s + f)]$
3. Update the set $Y_{s+1} = Y_s + f^+$
                  $s = s + 1$
4. Goto 2
```

Figure 3.5: Pseudocode for SFS algorithm

```
1. Start with the full feature set $Y_o = X$
                                  $s = 0$
2. Select the worst feature $f^- = \text{argmax}_{f \in Y_s}[C(Y_s - f)]$
3. Update the set $Y_{s+1} = Y_s - f^-$
                  $s = s + 1$
4. Goto 2
```

Figure 3.6: Pseudocode for SBS algorithm

The main disadvantage of the aforementioned sequential approaches is that they suffer from the so-called nesting effect. Thus, as a trade-off for their reduced computational complexity, SFS and SBS often tend to become trapped in local minima, while evaluating the separability criterion for given feature combinations. This emanates from their inability to re-evaluate features that were previously added (SFS) or discarded (SBS).

In an attempt to counteract the nesting effect (Pudil et al., 1994) has suggested the floating search method. The name results from the fact that dimensionality of the feature subset, during the search, can be thought to be “floating” up and down. Again, there are two schemes that implement this technique. Sequential Forward Floating Selection (SFFS) scheme starts with an empty set of features. After each forward step in which a feature has been added in the subset, SFFS performs backward steps, removing features provided that the criterion increases. In contrast, Sequential Backward Floating Selection (SBFS) begins with the full set of available features, and following each backward step, SFBS performs forward steps as long as the criterion increases. Pseudocode for SFBS and SBFS is shown in Figure 3.7 and Figure 3.8 respectively.

Although still suboptimal techniques, both floating search methods incorporate backtracking capabilities and, therefore, provide improved results compared to SBS and SFS, at the cost of higher computational complexity (Theodoridis and Koutroumbas, 2003).

```
1. Start with the empty feature set
        $Y_o = \{\emptyset\}$
        $s = 0$
2. Select the next best feature and update the set
        $f^+ = \mathrm{argmax}_{f \notin Y_s}[C(Y_s + f)]$
        $Y_{s+1} = Y_s + f^+$
        $s = s + 1$
3. Select the worst feature
        $f^- = \mathrm{argmax}_{f \in Y_s}[C(Y_s - f)]$
4. If $C(Y_s - f^-) > C(Y_s)$ then
        $Y_{s+1} = Y_s - f$
        $s = s + 1$
        Goto 3
   Else
        Goto 2
```

Figure 3.7: Pseudocode for SFFS algorithm

```
1. Start with the full feature set
        $Y_o = X$
        $s = 0$
2. Select the worst feature and update the set
        $f^- = \mathrm{argmax}_{f \in Y_s}[C(Y_s - f)]$
        $Y_{s+1} = Y_s - f^-$
        $s = s + 1$
3. Select the best feature
        $f^+ = \mathrm{argmax}_{f \notin Y_s}[C(Y_s + f)]$
4. If $C(Y_s + f^+) > C(Y_s)$ then
        $Y_{s+1} = Y_s + f$
        $s = s + 1$
        Goto 3
   Else
        Goto 2
```

Figure 3.8: Pseudocode for SBFS algorithm

3.7 Classification

As mentioned in section 3.2, the ultimate goal of classification is to assign unknown patterns (objects described by features) to a class (category or group), based on a criterion, referenced by the term “prediction rule”. Thus, a classifier is the mathematical representation of a prediction rule. A classifier takes as input a pattern, described by the most informative features, extracted in previous stages, and outputs a label or value indicating which class the pattern belongs to (Duda et al., 2012). The basic theory of benchmark classifiers, used for the purposes of this thesis, is given below.

3.7.1 Bayesian

The Bayesian classifier (Theodoridis and Koutroumbas, 2003), considered as the basic approach to statistical pattern recognition, is designed in order to minimise the total classification error, assuming normal probability distribution of the class-samples. The discriminant function i.e. mathematical representation of the prediction rule of the Quadratic Bayesian classifier for class C is given by :

$$D_C(x) = \log P_C - \frac{1}{2}\log\left|S_C\right| - \frac{1}{2}\left[(x-\mu_C)^T S_C^{-1}(x-\mu_C)\right] \tag{3.32}$$

where x is the unknown pattern vector, P_c is the prior probability of class C, μ_C is mean value of class C and S_C is the covariance matrix of class C defined as :

$$S_C = (x-\mu_C)(x-\mu_C)^T \tag{3.33}$$

The unknown pattern x is classified to the class C with the larger discriminant function value because it expresses the probability of pattern x to belong to class C.

3.7.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) classifier finds a linear transformation that, when applied on the features of the patterns, it yields a new set of transformed values that provide a more accurate discrimination. Assuming that each class has multivariate normal distribution and all classes have the same covariance matrix, the discriminant function of the LDA classifier for class C is:

$$D_C(x) = \log P_C + \mu_C S_w^{-1} x^T - \frac{1}{2} \mu_C^T S_w^{-1} \mu_C \tag{3.34}$$

where the within-class covariance matrix S_w is the weighted average of the separately estimated class-conditional covariance matrices,

$$S_w = \sum_{C}^{K} P_C S_C \tag{3.35}$$

and

$$P_C = \frac{N_C}{N_{all}} \tag{3.36}$$

where P_C is the prior probability of class *C*, *K* is the number of classes, N_C the patterns of class *C*, N_{all} the total number of patterns, μ_C is mean value of class *C*, S_C is the covariance matrix of class *C*, and *x* is the unknown pattern vector, which is classified to the class with the highest discriminant function value (Kuncheva, 2004; Duda et al., 2012).

3.7.3 k-Nearest Neighbours

The Nearest Neighbour classifier is a non-parametric classifier, where the unknown pattern is classified to the class that contains its closest neighbour. Similarly, the k-Nearest Neighbour (k-NN) classifier classifies the unknown pattern to class *j* if the majority of its *k* neighbours belong to class *j* (Kuncheva, 2004).

3.7.4 Probabilistic Neural Network

The implemented Probabilistic Neural Network (PNN) (Specht, 1990) algorithm, is a four-layer (input, pattern, summation and output layers), feed-forward and one-pass structure and encapsulates the Bayes' decision rule together with the use of Parzen estimators of data's probability distribution function. The discriminant function of the PNN classifier for class *C* is:

$$D_C(x) = \frac{1}{(2\pi)^{d/2}\sigma^d}\frac{1}{N_C}\left(\sum_{i=1}^{N_C} - \frac{(x-x_{ci})^T(x-x_{ci})}{2\sigma^2}\right) \tag{3.37}$$

where σ is the spread of the Gaussian activation function, N_C is the number of feature vectors of class C, d is the dimensionality of pattern vectors, x_{ci} is the *i-th* feature vector of class C, and x is the unknown feature vector. The latter is classified to the class with the highest discriminant function value.

3.7.5 Support Vector Machines

By mapping input vectors into a higher dimension feature space and defining the hyperplane that has the maximum distance from the closest training data, the SVM (Burges, 1998) can be utilised for binary classification problems with discriminant as follows:

$$D_C(x) = sign\left(\sum_{i=1}^{N_c} a_i L_C K(x, x_i) + b\right) \tag{3.38}$$

where x_i is the *i-th* feature vector belonging to class C, N_C is the number of feature vectors of class C, L_C is the label of class C, α_i, b are weight coefficients and K the transformation or kernel function (Burges, 1998). Kernel functions utilised were the radial basis function (RBF) kernel with value of $\gamma = 1/(2\sigma^2)$ and polynomial kernels with degree 1-3.

$$K_{RBF}(x, x_i) = \exp\left(\frac{-\|x - x_i\|^2}{2\sigma^2}\right)$$
$$K_{POLYNOMIAL}(x, x_i) = ((x^T x_i) + 1)^r \tag{3.39}$$

3.8 Evaluation strategies

The last stage of the design process of a pattern recognition system involves evaluating its performance. Selection of the adopted evaluation strategy is considered of paramount importance in pattern recognition, as the former provides an estimation of the classification error probability, using the available, hence finite set of data. Also, it should be underlined that the evaluation strategy is incorporated in the design of the pattern recognition system and, hence, it is not cut off from the previous stages of the design procedure (Theodoridis and Koutroumbas, 2003). Thus, besides allowing the extraction of conclusions regarding the generalisation of the designed classification system, evaluation methods can also enable selection of the "optimal" parameter(s) for a given classification problem. For instance, these parameters could typically include the number k of neighbours in a kNN classifier, or the value of the smoothing parameter σ in the PNN classifier. In addition, as mentioned afore, in the feature selection stage, wrapper methods utilise the misclassification probability to measure the performance of the classifier and accordingly choose the best features.

Accurate evaluation of a classification scheme would be a rather straightforward task provided that access to an unlimited number of samples was feasible. However, in the majority of real applications, this is not the case, as the number of available training and testing samples is finite.

Aiming to maximise the number of data used to build the classifier (training set) and simultaneously maximise the number of data used to assess its performance (testing set), one rather naïve approach is the re-substitution method (Kuncheva, 2004), according to which, the entire dataset is used for both training and testing. The re-substitution approach suffers from two main inefficiencies. Firstly, due to the fact that the same data are used for both training and testing, the re-substitution method provides optimistically biased estimates of the classifier performance (Foley, 1972). In fact, it is very common to achieve 100% correct classification on the training dataset. Secondly, because the classifier might overtrain to the available dataset i.e. the classifier perfectly learns and adapts on the available data, it fails to generalise in new, unseen data.

Another method for estimating a classifier's performance is the holdout method. In this approach the available dataset is split into two distinct datasets, as illustrated in Figure 3.9. Accordingly, the training dataset is utilised for training, while the test dataset is used to estimate the performance of the trained classifier. The main shortcoming of the holdout method is that it shrinks the size of both datasets, and consequently cannot be easily adopted in case the available data is limited (Kuncheva, 2004).

Cross-validation methods attempt to overcome the aforementioned limitations of re-substitution and holdout techniques at the cost of increased computational complexity. Hence, in the K-Fold Cross-Validation approach (see Figure 3.9), the complete dataset of size N is randomly partitioned into K subsets, each having a size N/K. Next, one subset is excluded and used to test the classifier. Training of the latter is accomplished utilising the union of the remaining $K-1$ subsets. This is repeated K times, each time selecting a different subset for testing, and using the remaining $K-1$ for training. The error probability is then averaged over the K steps. In the extreme case that $K=N$, i.e. the testing set comprises one sample, the Leave-One-Out (LOO) method is formed as presented in Figure 3.9 (Kuncheva, 2004).

The advantage of Cross-Validation method is that the complete dataset is eventually used for both training and testing, while ensuring independence of the training and testing subsets (Theodoridis and Koutroumbas, 2003). Nevertheless, its major drawback is that it involves high computational complexity due to the fact that classifier training has to be repeated K times. This continues to pose a challenge even with modern computing technology (Kuncheva, 2004).

In case the classifier design (feature selection, fine tuning of classifier parameters) and generalised performance estimates to new and unseen by the system data are to be computed simultaneously, it is not an uncommon practice to divide the available data into three disjoint datasets, one used for training, one for validation and finally one for testing. In addition, as it has previously (Ambroise and McLachlan, 2002) demonstrated cross-validation testing samples should be kept external to the design process. Accordingly, the validation dataset is used for feature selection, and fine tuning of the classifier parameters, while the cross-validation testing dataset remains unseen during the training process and is used only to assess the generalised performance of a fully-trained classifier (Kuncheva, 2004). It must be underlined that after assessing the performance of the classification scheme on the testing set, one should not proceed to any further tuning of the classifier, otherwise, this may lead to overtraining of the classifier.

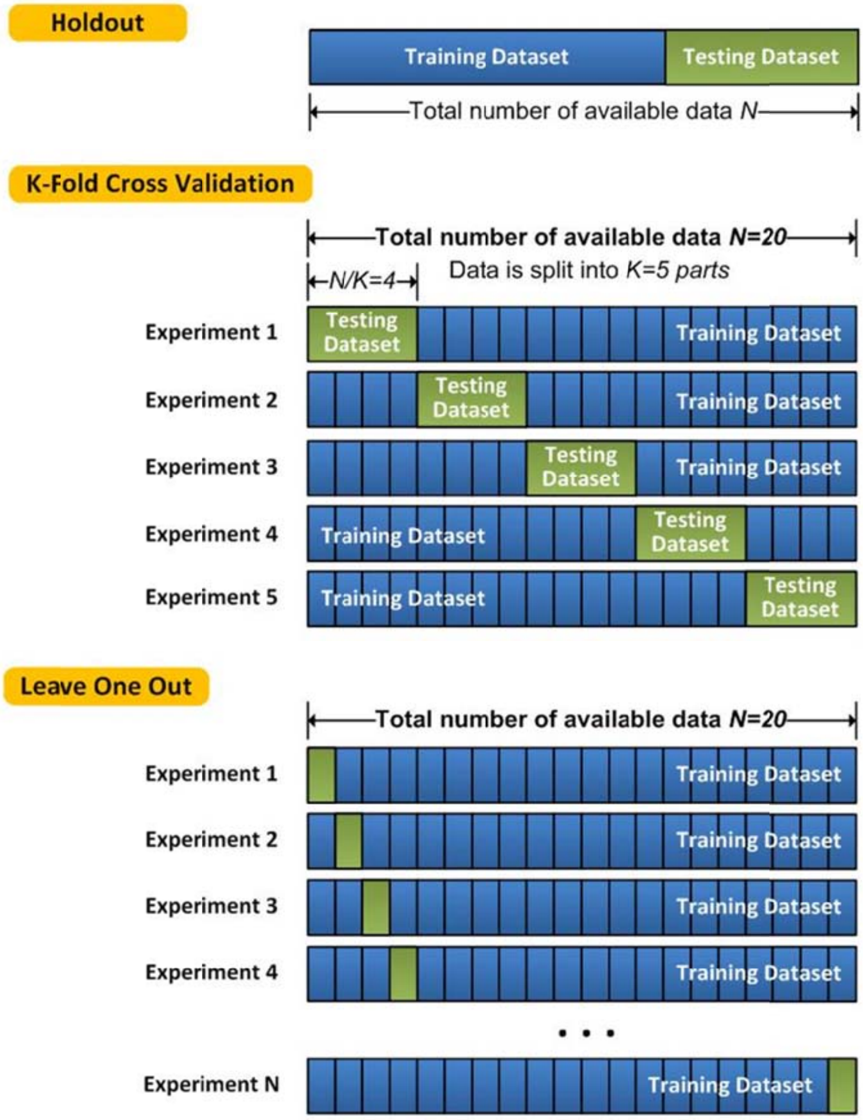


Figure 3.9: Different evaluation techniques

However, when three way splitting results in insufficiently small data-partitions, then, as it has been shown (Laine et al., 1996) the external k-fold cross-validation provides one of the most unbiased estimations (Ambroise and McLachlan, 2002). Hence, available data are randomly divided into two disjoint subsets: a training subset comprising 2/3 of available samples, and a test subset consisting of the remaining 1/3 of available samples. Training data are further split into training and validation datasets. Employing these two datasets, both feature selection and training of the classifier are performed. In particular, feature selection can be accomplished by adopting a wrapper method that evaluates the classifier's predictive accuracy on the validation dataset. Following optimal classifier design, its performance is evaluated on the remaining testing data. This process is repeated 10 times for 10 different random splits of all available data. The average accuracies and ranges are finally computed. In this way the external cross-validation estimate of the classifiers' performance is determined, which is considered as indicative of the classifiers' performance into new, unseen, data.

Nevertheless, as it can be easily inferred, selecting an external k-fold cross-validation evaluation technique for a PR system, imposes a significant computational burden. Moreover, the involved computational complexity is rendered prohibitive even for modern CPUs, in case the aforementioned technique is combined with an optimal feature selection method that employs exhaustive search in a large feature space (e.g. the use of multiple imaging modalities, as in our case) and the predictive accuracy of a classifier following a wrapper approach. A practical way out of this "computational maze" may be the adoption of parallel processing methods and the employment of powerful, programmable GPUs, originally developed to cover the growing need for more realistic and entertaining videogames.

3.9 Summary

In this chapter, the basic concepts and methods of statistical pattern recognition were introduced. Key ideas and the stages involved in the design of a classification system were presented. In particular, algorithms and methods discussed in this chapter include a) calculation of textural features from the image histogram, the co-occurrence and run-length matrices, b) feature reduction and both optimal and suboptimal feature selection, c) basic classifiers, and d) approaches for pattern recognition system performance evaluation. In addition, all the aforementioned methods were assessed in terms of their computational complexity. As a conclusion, optimal training of a classification scheme presents a computational challenge even for modern CPUs, giving rise for the adoption of efficient, yet suboptimal methodologies that fail to identify the full discriminatory potential of the available dataset. The necessity of applying optimal design methods for increasing the precision of pattern recognition systems, thus, becomes apparent. To deal with the computational challenge imposed, one may have to resort to parallel processing through the use of GPUs. This is necessary especially in the present study, due to the requirement of increased feature space dimensionality, introduced by the multimodality nature of the problem at hand.

Chapter 4

GPUs as Parallel Processing Accelerators

4.1 Overview

In this chapter the use of GPUs as parallel processing hardware accelerators is introduced. It begins with the history and evolution of multi-core microprocessors and GPUs over the last 2 decades. An account of the main architectural differences between CPUs and GPUs is given, along with the main reasons behind the ever growing popularity of GPU-accelerated computing and its limitations. This chapter also includes a brief description of the Compute Unified Device Architecture (CUDA) programming model, thus, providing a theoretical foundation for the subsequent chapter.

4.2 The dawn of the Multi-Core Era

Since their introduction in early 1980s, and for more than two decades, personal computers have been continuously improving in their performance, powered by microprocessors that featured a single CPU. Until the first years of the new millennium, major microprocessor vendors, such as Intel and AMD, managed to relentlessly deliver CPUs able to perform more and more arithmetic operations per second.

Predicted by Moore’s law (Moore, 2006), this continuous increase in performance was the product of technical innovations that enabled the semiconductor industry to pack more transistors in an affordable CPU, operating at an ever-increasing clock speed. This was particularly convenient for both users and the software industry, due to the fact that their algorithms and applications were guaranteed to run faster, as every new generation of CPUs was introduced.

However, in 2003, this “roller coaster” ride was slowed down, as the semiconductor industry hit, what is known as, the “power wall”, a term used to summarise the inherent technical limitations, related to energy consumption and heat emission levels of a CPU. In particular, in their effort to reduce the size and increase the speed of transistors i.e. their clock frequency, microprocessor designers realised that reduction in the operating voltage could not be sustained at the same pace as in the past. Consequently, the transistors produced prohibitively high amounts of waste heat per square millimetre of silicon. Unable to resolve this heat dissipation issue in a cost effective way, chip manufacturers were forced to re-evaluate their approach and accept the fact that although they could still produce smaller transistors and pack more of them in a single chip, any increase in their operating clock speed, was not an option. Hence, chipmakers came up with the only viable solution at the time. They reduced the clock frequency of the transistors and added multiple microprocessor cores in a single CPU. Even though each processor core would have lower computational abilities, their aggregate computational capacity would be higher, leading to increased overall performance. Therefore, designers expected that the gain, resulting from the extra parallelism, would compensate for, or even overbalance, the loss from the lower clock frequency, while at the same time, the CPU would consume less power and, thus, generate less waste heat per square millimetre of silicon (Moore, 2010). The first dual-core CPUs designed for desktop PCs were made available in 2005, as the Athlon 64 X2 (AMD, 2005), and the Pentium D (Intel, 2005) were released by AMD and Intel respectively. To distinguish these novel CPUs from traditional single-core architectures, the term multicore microprocessors is used (Moore, 2010; Patterson, 2010).

Since the advent of the first generation of multicore CPUs, “the core has become the new transistor” (Patterson, 2010) as all microprocessor vendors focused on incorporating more processor cores in each CPU chip, aiming to achieve higher performance through parallelism. Today, multicore CPUs can be found in wide range of everyday devices, from desktop PCs and notebooks, to smartphones and tablets.

This advancement had a significant impact on the software industry mainly because most software applications at the time were following the conventional sequential programming model, whereas instructions are executed in a sequential order (one after the other) by a single processor thread (Sutter and Larus, 2005). Because the traditional sequential model could not take advantage of the increasing number of processor cores found on these new CPUs, the software industry had to find ways for developers to introduce concurrency in their applications, without lowering the quality of the produced software, or stretching the development time (Patterson, 2010). This increasing need for concurrency gave rise to parallel programming.

4.3 Parallel Programming

Although parallel programming was not, by any means, new, it had been historically confined to the high-performance computing community for decades, and only recently it has evolved into a mainstream paradigm in everyday information processing, mainly due to the proliferation of multicore CPUs. Parallel programming employs concurrency to build programs that execute faster and, thus, achieve high-performance computing. In other words, parallel programming enables many instructions of the same program to be executed in parallel. The idea is to decompose, or split, a specific problem, solved sequentially, into smaller, independent parts that can be solved in parallel, exploiting the ubiquity of multi-core processors. The term computational grain refers to these pieces of sequential algorithms that can be executed in parallel (Sen, 2012).

Defined as the ratio of execution time required by a sequential program, to the time required by its parallel version, speed-up is a common measure of the achieved performance gain. It could be easily inferred that speed-up increases with the number of available processor cores. However this is not entirely accurate. According to Amdahl's law (Kumar et al., 1994), the theoretical maximum speed-up of a program executed by multiple processors is limited by its parallel fraction, i.e. the part of the program than can be executed in parallel (Park et al., 2011). In particular, this law states that if P is the fraction of a program that can be made parallel and 1-P is the fraction that cannot be parallelised, i.e., that must remain sequential, then the maximum theoretical speed-up that can be achieved using n processors is expressed by Equation 4.1:

$$Speedup = \frac{1}{(1-P)+\frac{P}{n}} \tag{4.1}$$

One obvious repercussion is that just by adding more processors does not always result in a significant speed-up as, in practice, performance falls rapidly once there is even a small component of (1-P). Hence, effort must be put to increase the parallel fraction and shrink the sequential. This was one of the primary concerns while designing the proposed GPU-based PR system.

Increasing the aforementioned parallel fraction can be accomplished by introducing Task and Data parallelism. Task parallelism refers to independent tasks that are executed in parallel, while data parallelism is achieved by a task or a set of tasks operating on different data (Sen, 2012). These two categories of parallelism will be discussed in a subsequent paragraph, as not both of them are compatible for the GPU execution model.

Despite the fact that quite a few parallel programming languages, Application Programming Interfaces (APIs) and models have been proposed in the past, the OpenMP and the Message Passing Interface (MPI) managed to stand out and be widely used (Mattson et al., 2004).

OpenMP (OpenMP, 2005) is an extension of C++ programming language and is based on the shared memory model. As presented in Figure 4.1, in the shared memory model all processor cores share a single global memory. The latter provides a means of communication among the processor cores and significantly reduces the effort required to port a sequential program to OpenMP, as it enables incremental conversion of its blocks of code so that they can be executed in parallel. Although flexible enough to support a wide range of applications, OpenMP's main drawback is that, due to hardware limitations related to thread management and cache coherence, it fails to efficiently support more than a couple of hundred processor cores (Chapman et al., 2008; Kirk and Wen-mei, 2012).

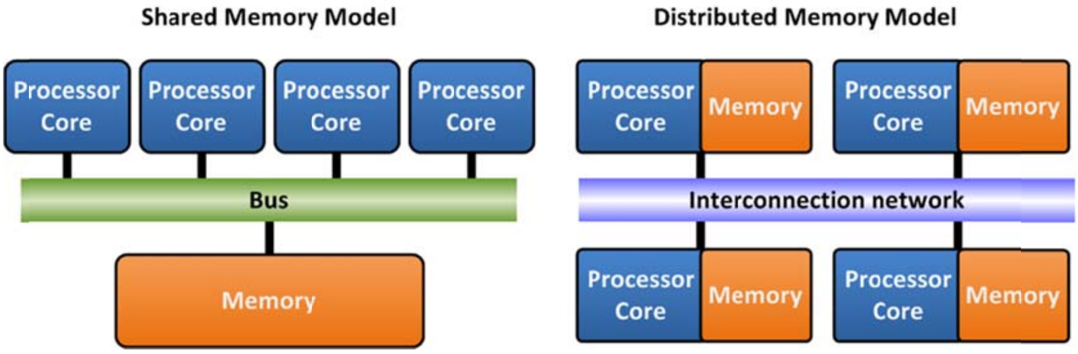


Figure 4.1: The two main memory models used in parallel programming.

On the other hand, MPI (MPI, 2009) is based on the distributed memory model, according to which every computing node has its own memory and communication among all nodes is performed via message passing through an interconnection network, e.g. Ethernet (see Figure 4.1). MPI has been particularly popular in the high-performance scientific community, as it mainly targets expensive high-end computer clusters, and its scalability has reached cluster systems with thousands of computing nodes. However, the lack of shared memory increases the effort required to efficiently port sequential applications to MPI (Kirk and Wen-mei, 2012).

4.4 The evolution of GPU as programmable parallel processor

Today, a modern GPU constitutes a typical example of massively parallel computing platform. Although, they were originally designed as dedicated processors, optimised to handle solely computer graphics computations, over the last 30 years, GPUs have been transformed into widely deployed commodity desktop parallel computers, as described in (Crow, 2004; Luebke and Humphreys, 2007).

Driven by the growing market demand for high quality, real time graphics in computer games and applications, GPUs have evolved from devices that simply implemented the traditional fixed-function 3D graphics pipeline towards flexible, programmable, general-purpose computational engines. GPUs present a significantly higher theoretical performance and memory bandwidth than CPU, as depicted in Figure 4.2.

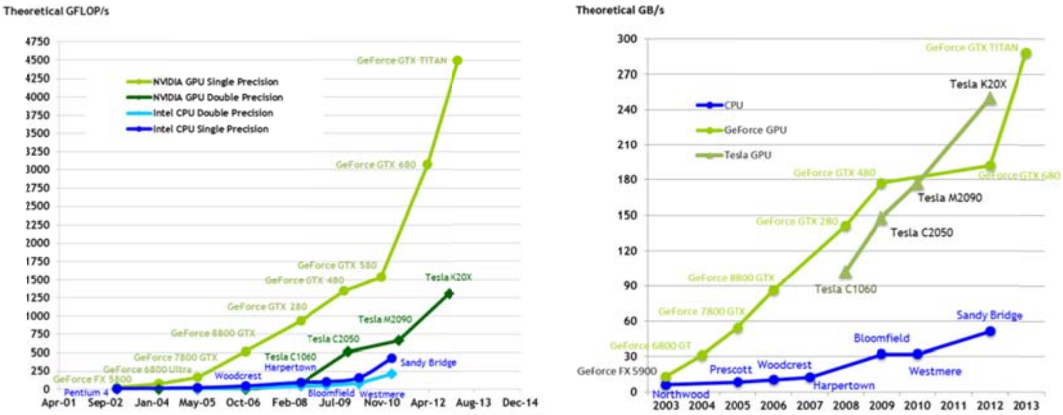


Figure 4.2: Comparison of Floating-Point Operations per Second (FLOPS) and memory bandwidth for the CPU and GPU (Nvidia, 2014)

A quick comparison between a CPU and a GPU can reveal their fundamental architectural differences. As illustrated in Figure 4.3, unlike a typical CPU, which features only a few processor cores, e.g. 4 in case of Intel's Core i7 series, even a low end GPU comprises thousands of massively parallel processing units. In other words, GPUs devote more transistors to data processing, while CPUs count on mechanisms for data caching and flow control to achieve low latency. As a result, while CPUs are optimised for low latency, GPUs are optimised for high throughput. In addition, a GPU features a dedicated, high-speed memory.

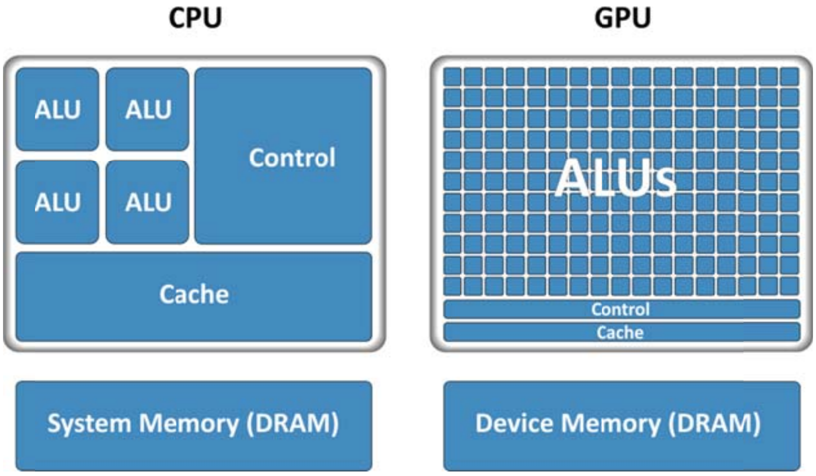


Figure 4.3: Typical CPU and GPU architectures

This radical difference in the GPU’s architectures was shaped by the fact that real-time computer graphics demand a highly parallel workload. Thus, GPUs combine extremely high arithmetic throughput with high memory bandwidth as they were designed exclusively to carry out numerous floating point calculations that determine which screen pixels are to be lighted and how, 24 or even 60 times within a single second. Over the last decades, this demand for more realistic graphics forged GPUs to inherently support high data parallelism (Luebke and Humphreys, 2007).

Early on, as GPUs were getting more powerful and more programmable, researchers began to seek ways to exploit this raw computational power for applications other than graphics. Although certain algorithms and applications cannot benefit by the GPU’s architecture, mainly because they are inherently sequential, many compute-intensive science and engineering problems can leverage the GPU’s computational “horsepower” and achieve tremendous speed-ups. Consequently, in 2002 the term “General Purpose Computation on Graphics Processing Units” (GPGPU) was introduced (Di Blas and Kaldewey, 2009).

Nevertheless, in those early years GPGPU was not an easy task as it required researchers to map scientific problems onto native graphics operations that could be represented by triangles and polygons, using graphics programming languages such as OpenGL, Direct3D or Cg. In this rather obscure way, researchers “tricked” the GPU to perform general purpose computations.

It wasn’t until the introduction of specialised GPU programming frameworks that the use of GPUs for general problems began to gain momentum. In 2003, Stanford University’s BrookGPU (Buck et al., 2003) and University of Waterloo’s Sh (McCool et al., 2002) were the first programming models to expose the GPU as a general-purpose processor in a high-level language.

In 2006, Nvidia introduced the Compute Unified Device Architecture (CUDA), the first complete solution for GPGPU. CUDA included both hardware specifications, which were implemented across all Nvidia’s GPU models, and software programming tools, that made the development of GPGPU applications an easier task (Nvidia, 2014). Soon after its debut, CUDA technology was widely adopted by researchers who employed this new GPU architecture to accelerate challenging, scientific, traditionally CPU domain problems in various fields such as physics, mathematics, chemistry, and biology (Che et al., 2008; Schenk et al., 2008).

In 2009, GPGPU had received so much attention that a group of major vendors, including Nvidia, AMD Graphics, Intel, IBM and Apple, put their forces together and created OpenCL, a new, cross-platform, parallel programming language (Khronos, 2008).

In the present work, the CUDA framework was selected over other GPU programming platforms for the implementation of the proposed PR system, mainly because, compared to OpenCL, CUDA was more mature and constituted a more complete solution. Additionally, it has been argued that due to its portability, OpenCL achieves lower performance compared to CUDA (Fang et al., 2011; Weber et al., 2011; Kakimoto et al., 2012; Pallipuram et al., 2012).

4.5 GPU Architecture Overview

Since its introduction, Nvidia has released three distinct CUDA-capable GPU architectures: i) Tesla-class GPUs, debuted in 2006, in the GeForce 8800 GTX (G80), ii) Fermi-class GPUs were introduced in 2010, in the GeForce GTX 480 (GF100), and iii) Kepler-class GPUs, debuted in 2012, in the GeForce GTX 680 (GK104). Although, every new hardware class introduced new features, they all share common architectural characteristics. The following paragraphs contain a brief description of the key architectural concepts considered essential in understanding GPU programming.

Typically, a GPU comes in the form of an individual card that can be plugged onto one of the high speed IO bus slots of a computer workstation’s motherboard, such as the PCI-express port. Details of the connection between the CPU and the GPU are defined by the chipset used. The term “device” is used to describe the GPU, while the “host” refers to the rest of the workstation, including the system’s CPU.

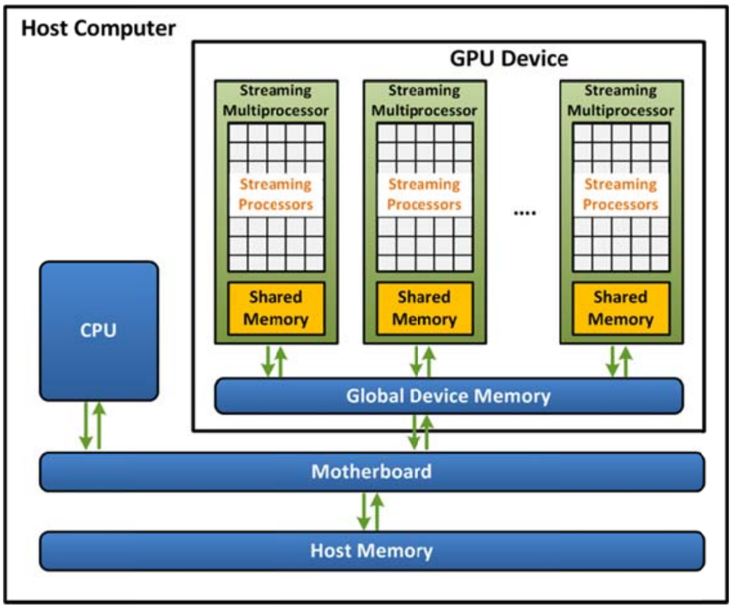


Figure 4.4: Hardware structure of a workstation featuring a GPU

As presented in Figure 4.4, the GPU arrives with its own device memory. The latter varies in size among different GPU models but up to 4 gigabytes can be commonly found in current configurations. CUDA enables programmers to perform data transfers between the device and the host memories through Direct Memory Access (DMA). As explained afore, GPU’s architecture does not depend on complex memory caches. Instead, the device connects its dedicated memory with its processor cores via a high bandwidth channel that reaches 512 bits in certain GPU models, enabling multiple memory accesses in one clock cycle.

Under the hood, the workhorse of the GPU is the Streaming Multiprocessor (SM). GPUs inherent parallelism stems from the fact that each of these SMs executes in parallel with the others. The number of SMs in a CUDA GPU may vary from 2 to a few dozens. For instance, the very first CUDA-capable GPU, the GeForce 8800 GTX, featured a total of 16 SMs while the GeForce GTX 280, featured a total of 30 SMs.

In the heart of every SM, lies a number of processing units called Stream Processors (SPs), or CUDA cores. On Tesla-class GPUs, each SM featured one group of 8 SPs, while every Fermi-class SM featured 2 groups of 16 SPs. Consequently, high end Tesla and Fermi GPUs had a total of 240 SPs and 512 SPs respectively. It should be mentioned that, currently, high end Kepler-class GPUs, such as the GK110b, feature 2880 SPs in a single GPU chip. Figures 4.5 and 4.6 present the block diagrams of Tesla and Fermi GPUs respectively.

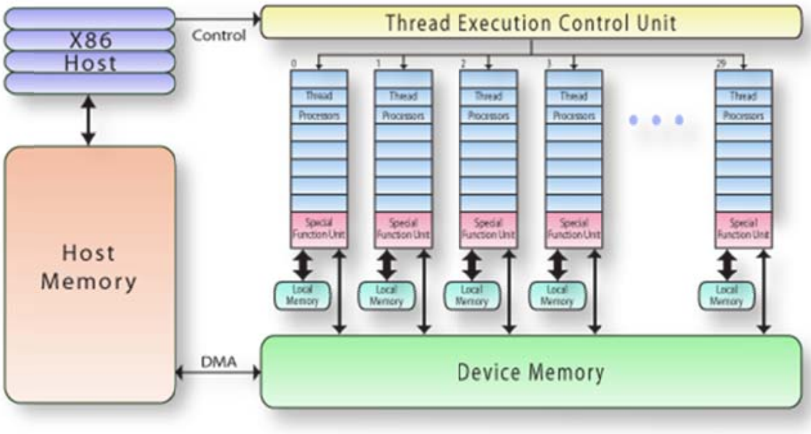


Figure 4.5: Block diagram of Tesla-class GPUs (Wolfe, 2010)

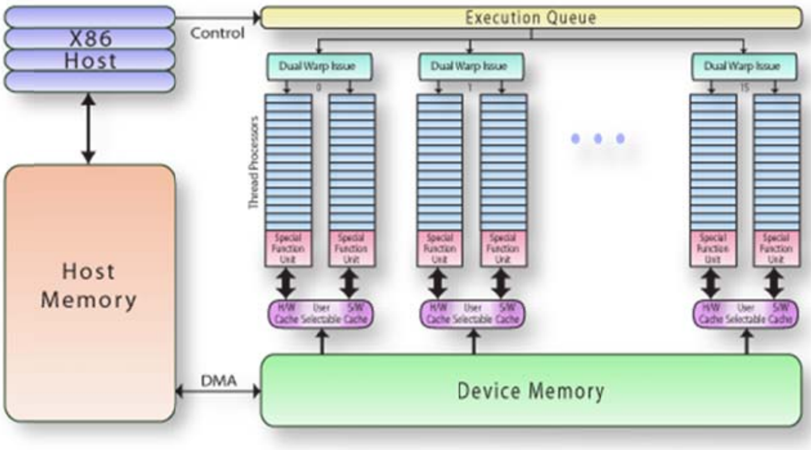


Figure 4.6: Block diagram of Fermi-class GPUs (Wolfe, 2010)

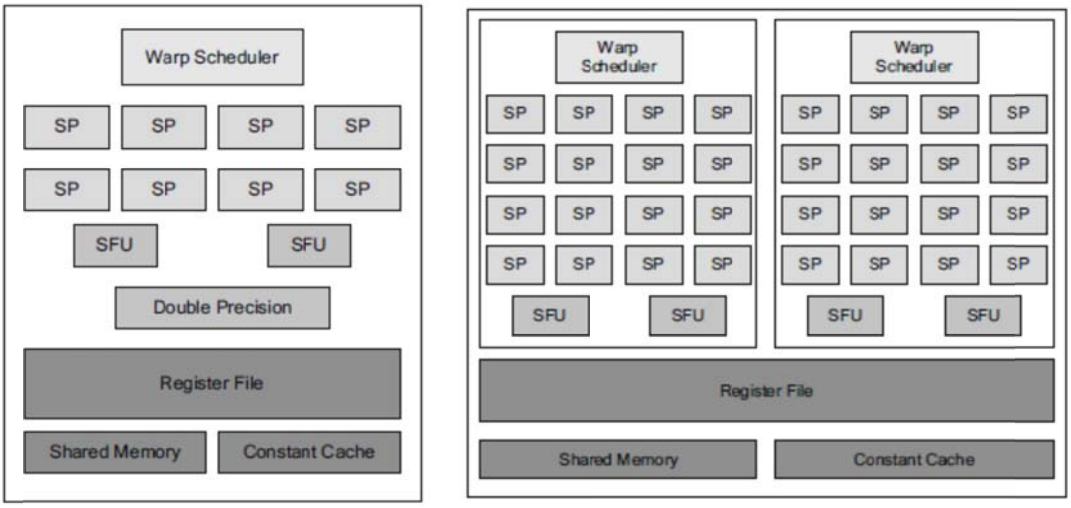


Figure 4.7: Streaming Multiprocessor of Tesla (left) and Fermi-class (right) GPUs (Wilt, 2013)

Every SP has a single-precision floating point and an integer functional units, while shared Special Function Units (SFUs) in each multiprocessor handle transcendental (exponential, logarithmic, and trigonometric functions) and double-precision operations. Especially for handling double-precision instructions, Fermi SM combines both groups of SPs to look like a group of 16 double-precision units. Figure 4.7 presents the SM for both architectures.

Every single SP has the capacity to execute a thread, which is a piece of sequential code. However, all SPs of the same group execute the same instruction at the same time. This way of execution is called Single Instruction, Multiple Thread (SIMT) and leads to high performance provided that there is no significant divergence in the instruction executed by each SP, due to a conditional branch (if-else) or other control flow statements. In case threads within a warp diverge, the warp serialises execution of every branch path. It should be noted that this model explains the reason why GPUs are suitable for data parallelism. In contrast, it is quite difficult to achieve high performance through task parallelism on GPUs, because the existence of threads following different instruction paths throughout the algorithm results in severe decrease in performance.

Another CUDA key-concept is the warp. A warp is a group of threads (currently 32) that are executed in SIMT fashion. In particular, a Tesla GPU employs the 8 SPs of an SM to execute one single-precision instruction of every thread across an entire warp in four clock cycles. Accordingly, a Fermi GPU uses its 16 SPs to execute one single-precision instruction for each of two warps in two clock cycles.

Warps are very important as they are used by GPUs to hide memory latency issues. For instance, when the threads of one warp issue a time consuming instruction to fetch data from memory, the responsible SM finds another warp and switches execution to those threads. In this way, the SPs are kept constantly occupied, and provided that there is enough parallelism, the GPU tolerates memory latency by using a high degree of multithreading. Every Tesla-class SM supports up to 32 active warps while each Fermi-class SM supports up to 48 active warps. This results in an extremely high number of active threads (Farber, 2011; Kirk and Wen-mei, 2012; Wilt, 2013).

4.6 The CUDA Execution Model

A kernel is a special function executed in a number of copies, or threads, by the GPU device. On Tesla-class hardware, each kernel completes execution before the next kernel begins. Figure 4.8 provides a small example of how the GPU executes a small program that multiplies a set of data by the number 2, in parallel.

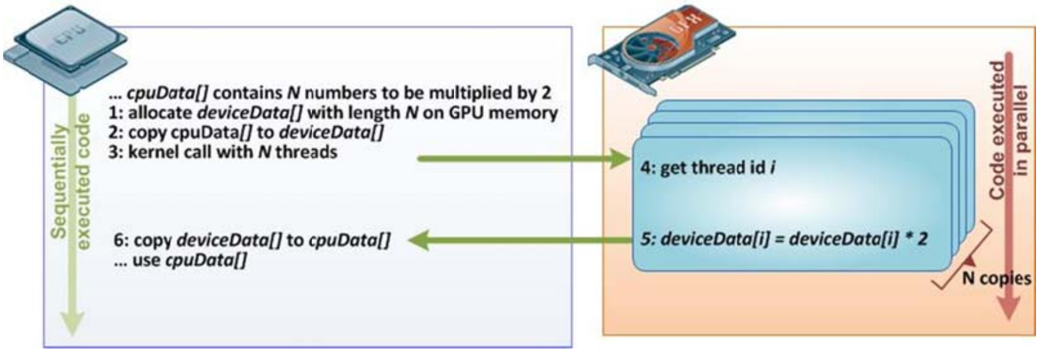

CPU
GPU
... cpuData[] contains N numbers to be multiplied by 2
1: allocate deviceData[] with length N on GPU memory
2: copy cpuData[] to deviceData[]
3: kernel call with N threads
4: get thread id i
5: deviceData[i] = deviceData[i] * 2
6: copy deviceData[] to cpuData[]
... use cpuData[]
Sequentially executed code
Code executed in parallel
N copies

Figure 4.8: Example of a GPU-based program that multiplies a set of data by the number 2

Concurrent execution of different kernels is supported by both Fermi and Kepler GPUs. According to the execution model defined by CUDA, the host program can launch a sequence of kernels on the device. CUDA organises every kernel as a hierarchy of threads.

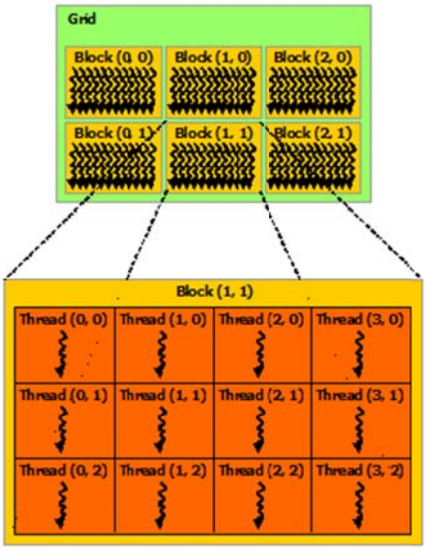


Figure 4.9: Thread hierarchy in the CUDA programming model (Nvidia, 2014)

Before distributed to SMs for execution, these kernel threads are grouped into a number of blocks, and blocks are, in turn, grouped into a grid. Every thread can be identified by the combination of its unique local index within the block, and the unique block index within the grid (see Figure 4.9). Tesla limits the maximum number of threads per block to 512 while Fermi puts the limit to 1024 threads.

Threads within a single block are assigned on the same SM and utilise its available registers. Synchronisation and data sharing among the threads of the same block is enabled via a small software-managed data cache shared among the SPs, called shared memory. The size of this low-latency memory is 16KB and 64KB on Tesla and Fermi respectively.

Additionally, blocks are not handled in an orderly fashion as their execution is scheduled dynamically. As every SM executes warps of threads in parallel and asynchronously, threads belonging to different blocks can exchange data only through the device's global memory. It must be noted that, although this is the only device memory accessible from the host, its bandwidth is relatively low compared to other types of memories.

Another characteristic of the CUDA execution model is that, until the advent of Kepler-class hardware, dynamic parallelism was not supported. As a result, a thread cannot launch a kernel and spawn more threads on Tesla and Fermi architectures.

4.7 Limitations of GPUs -Considerations

Modern GPUs are used as massively parallel processors to solve challenging scientific problems. However, in order to fully exploit their tremendous computational power, GPUs have to be appropriately utilised. Hence, understanding the GPU's programming model and its inherent limitations is considered vital while parallelising algorithms and applications in order to achieve high performance.

Similarly to any parallel programming problem, the first step involves the identification of specific parts of the code responsible for prolonging the execution time. Keeping in mind Amdahl's law, the aim is to increase the parallel part of the algorithm, while decomposing the problem in a way that exposes enough parallelism not only to populate all the SMs of the GPU but also to enable them to keep their SPs busy and maximise throughput by multiple active threads. Furthermore, because dynamic parallelism was not supported until the advent of Kepler-class GPUs, the computational grain has to be identified and implemented in kernel functions.

The SIMT execution model of a GPU is suitable for data parallelism, as any thread divergence, within the same warp, limits the overall efficiency. Therefore, GPU kernels should be carefully designed in such a way that thread divergence is eliminated. To the programmer, this entails that use of conditional and flow control statements should be avoided where possible; something that is hardly met in real problems. Furthermore, the block size, i.e. the number of threads inside each block, should be chosen as a multiple of the warp size, to maximise efficiency.

Another issue that could affect performance is the use of global device memory. Due to the fact that data must be copied from the host to the device memory before processing and results must be retrieved from the device memory afterwards, overall speed-up is limited by the amount of data transfers, as they involve high latency. Thus, one approach is to copy most, or if possible all, data to GPU and keep them there. Additionally, kernels should be designed in a way that visits to global device memory are minimised. If this cannot be applied, device memory accesses should be optimised for contiguous data while enough threads need be introduced in order to hide latency.

Other issues that should be considered include, among others, the use of single instead of double precision operations in favour of performance, the size of kernels that affects register usage, and the use of intrinsic functions instead of regular functions whenever accuracy is not an issue (Farber, 2011; Kirk and Wen-mei, 2012; Wilt, 2013).

In short, optimising GPU code, in order to maximise its performance, poses a significant challenge when developing GPU-based applications, as it requires tremendous efforts and it is usually problem-specific.

4.8 Summary

In the present chapter the fundamentals of GPU's architecture were presented. In particular, GPU comprises thousands of massively parallel processing units. While CPUs are optimised for low latency, GPUs are optimised for high computational throughput. The introduction of specialised GPU programming frameworks rendered the use of GPUs for general problems accessible to researchers. CUDA, includes both hardware specifications, which were implemented across all Nvidia's GPU models, and software programming tools, that made the development of GPGPU applications an easier task. Under the hood, the workhorse of the GPU is the Streaming Multiprocessor (SM) while in the heart of every SM, lies a number of processing units called Stream Processors (SP), or CUDA cores. Every single SP has the capacity to execute a thread, i.e. a piece of sequential code, in a Single Instruction, Multiple Thread (SIMT) fashion, which leads to high performance provided that there is no significant divergence in the instruction executed among the SPs of the same SM.

Chapter 5

System Design and Implementation

5.1 Overview

This chapter describes in detail and in a systematic way the steps followed throughout the whole CAD (PR- system) design procedure such as protocols followed and software designed for multimodality data collection, methods of feature generation and reduction, classifiers designed and tested for execution on GPU and in parallel using the CUDA framework, and evaluation methods employed for assessing system precision and acceleration of system design.

5.2 Data Collection and Clinical Material

Women scheduled for yearly screening breast examination or with clinical symptoms of breast pathology were subjected to digital mammography (DM). Those having further findings were also submitted to ultrasound (US) examination. Both DM and US examinations were conducted by the same experienced radiologist at the Department of Radiology of the Delta Digital diagnostic centre, Athens, Greece. Subjects assessed lower than III in the BI-RADS (Breast Imaging Reporting and Data System, according to American College of Radiology) categorisation were excluded from the study. Those having BI-RADS score of III or higher were submitted to surgical biopsy for histological verification. Hence, tumour specimens, excised from each patient by a physician, were histologically verified.

Finally, the selected dataset comprised both DM and US images from 62 female patients; thirty two (32) were diagnosed as malignant (Ductal Carcinomas In Situ, Ductal Infiltrating Carcinomas, Infiltrated Lymph Nodes, Mucoid Carcinomas) and thirty (30) as benign tumours (Adenolipomas, Cysts, Fibroadenomas, Lipomas, Papillomas). The mean age of the patients with malignant lesions was 50.6 years old (standard deviation: 8.4; range: 34-73 years) and of the patients with benign lesions was 47.3 years old (standard deviation: 9.6; range: 35-72 years). Table 5.1 and Figure 5.1 provide a summary of the dataset used for the purposes of this research.

All US examinations were performed on a General Electric Logiq 9 US system featuring a linear, 3D/4D-capable probe operating at 8-16 MHz, while the DM system utilised for this study was a General Electric Senographe Essential. Both diagnostic imaging devices were regularly checked by a professional medical physicist in order to safeguard them against degradation of their image quality. Thus, the contrast resolution, the spatial resolution, the contrast-to-noise ratio and other image quality measures were assessed on a regular basis and the devices were calibrated accordingly.

Table 5.1: A summary of the verified breast cancer cases comprising the collected dataset.

Breast Lesions Dataset			
Benign Cases	Adenolipoma	2	**Total**
	Cyst	6	
	Fibroadenoma	16	30
	Lipid Oil Cyst	2	
	Lipoma	3	
	Papilloma	1	
Malignant Cases	Ductal Carcinoma In Situ	2	**Total**
	Infiltrated Lymph Node	1	
	Infiltrating Ductal Carcinoma	27	32
	Infiltrating Lobular Carcinoma	1	
	Mucoid Carcinoma	1	

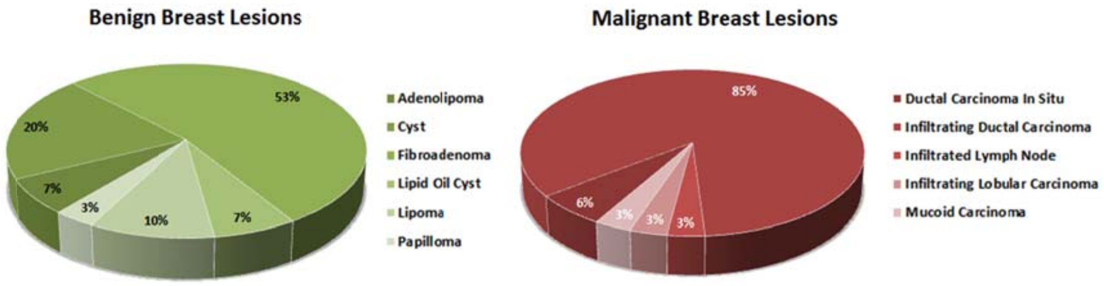


Figure 5.1: Percentages of benign and malignant breast cancer cases in the collected dataset.

5.3 Ethical Issues

At this point, it should be underlined that collection and processing of every piece of patients' sensitive data was performed in accordance with well-established ethical guidelines and regulations.

In particular, all data collected for the purposes of this research was handled in compliance with the EU Data Protection Directive, or also known as Directive 95/46/EC (Directive, 1995). Adopted by all EU countries, EU Data Protection Directive aims to protect the privacy of all personal data collected for, or about citizens, especially as it relates to processing, using, or exchanging such data.

Thus, multimodality breast images and accompanying demographic, clinical and histological data were collected following a written permission from the individual, obtained by the physician in charge of the patient. For this purpose the subject was provided with an informative document, an appropriate form (see Appendix B) to give her consent and with sufficient time to reach a decision.

Regardless of the patient’s decision, whether to participate in the study or not, she got the best available treatment and care, as under no circumstances did the data collection process represent a risk for the patient or did it interfere with the standard treatment and diagnostic work-up.

All personal data were handled anonymously and the identity of the subjects involved remained known only to the physician responsible in the diagnostic centre. Therefore, data pertaining to patients were anonymised and a running number was issued for each person. Finally, any and all anonymised patient information was kept in a secure database which was accessible only to the authorised researchers involved.

5.4 Breast Lesions’ Delineation

The next step involved the delineation and extraction of Regions of Interest (ROIs). Hence, from both the US (512x384) and DM (2394x3062) images, lesions were outlined by the radiologist, who delineated the boundaries of each tumour employing a specially designed software. ROIs, thus, outlined by the physician (see Figure 5.2), were stored and were subsequently used in the design of the PR-system.

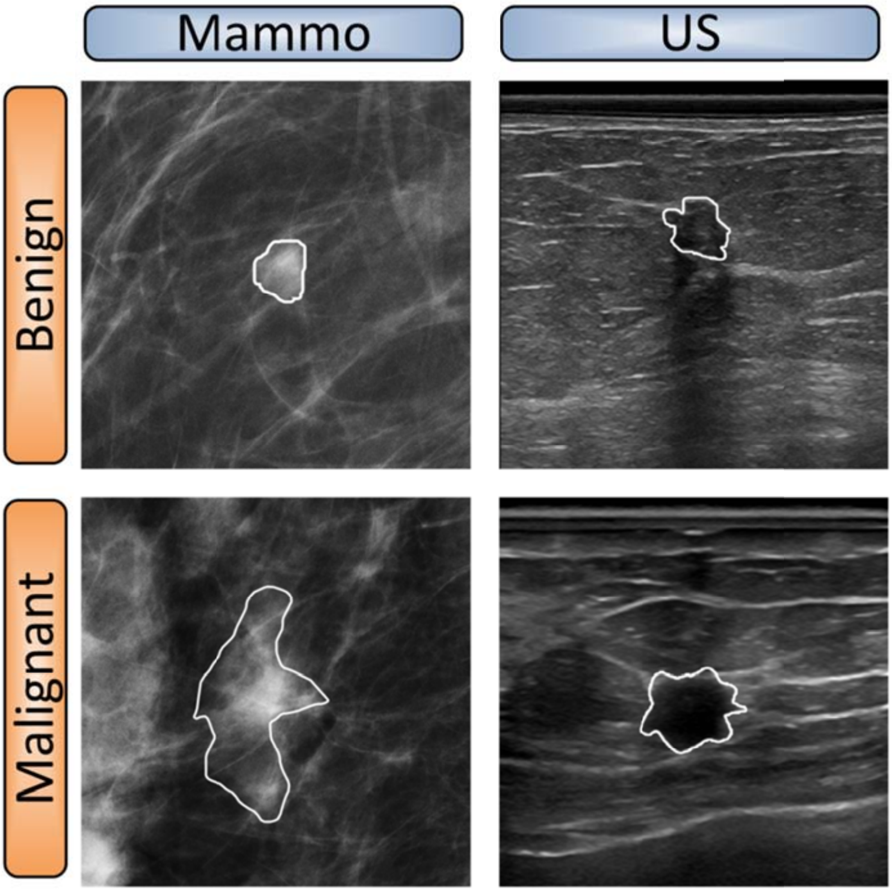


Figure 5.2: From each case Digital Mammography and Ultrasound ROIs were delineated. First row presents the ROIs from a benign case while the second row presents a malignant case

ImageNotes was a custom made desktop application developed in Java programming language version 1.5, solely for the purposes of this research, which enabled the user to view and manipulate DICOM images. JAVA was chosen mainly because it combines the principles of Object Oriented Programming (OOP), enabling the development of modular programs and reusable code, with the advantage of being platform-independent, i.e. to be able to move easily from one computer platform to another. Software development was assisted by the Netbeans Integrated Development Environment – IDE. Additionally, PixelMed's, open source, DICOM Toolkit was utilised for reading, validating and viewing of DICOM images, while Oracle's Java Advanced Imaging (JAI) and the associated Image I/O Tools libraries were employed for decoding JPEG images, mainly due to the fact that the Java SDK cannot always handle correctly JPEG Lossless compression. Patient information along with delineated ROIs were stored in an XML file (see Appendix A). A screenshot of the aforementioned application is presented in Figure 5.3.

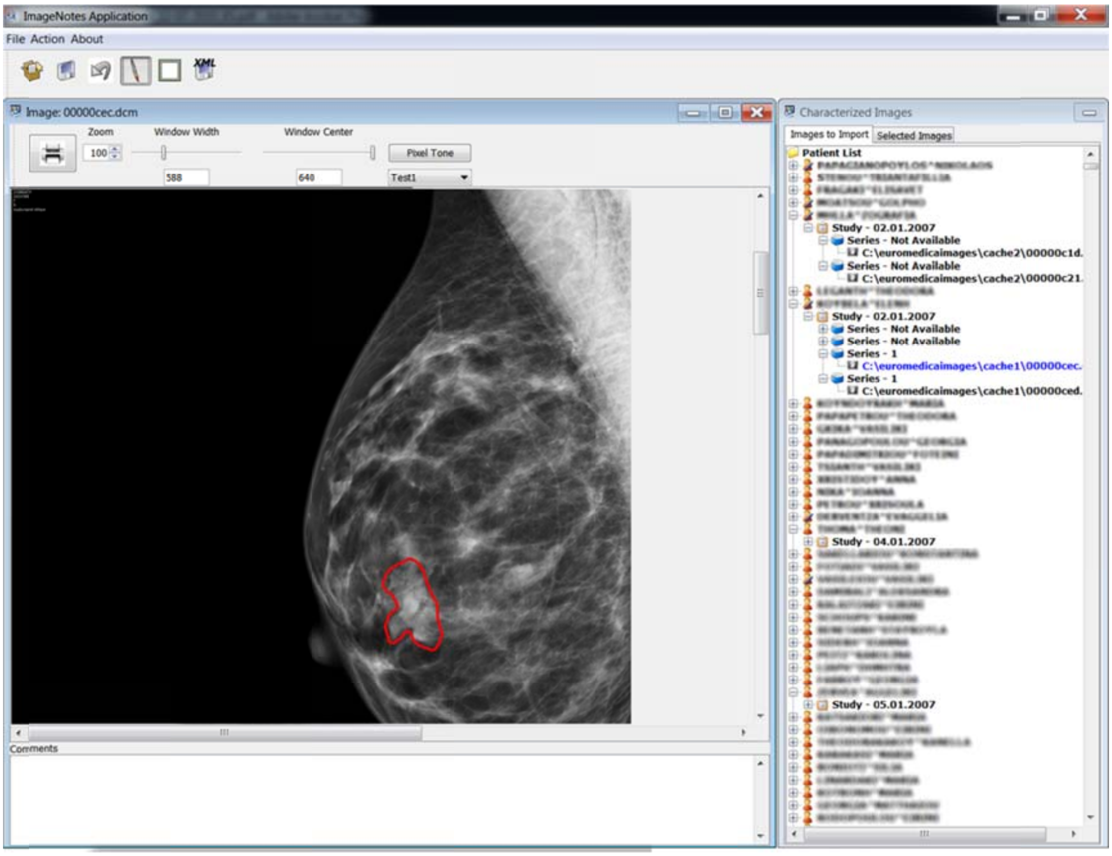


Figure 5.3: Screenshot of the custom made software developed for the purposes of this research. The software enabled the physician to browse for a case and delineate the region of interest

5.5 Feature Extraction

From each image-ROI, a large number of textural features was calculated. These features quantified textural properties of each tumour, as described in chapter 3, and comprised: four (4) features from the ROI's histogram, twenty six (26) features from the ROI's co-occurrence matrices (average and range values), and 10 features from the ROI's run-length matrices (average and range values). Thus, each patient was represented by an eighty (80) features vector, which consisted of 40-features extracted from the DM image and 40-features computed from the US image. For further analysis, data were split into three different configurations: (i) the US dataset, containing one class with 32 malignant patterns and one class with 30 benign patterns, each pattern containing 40 features, (ii) the DM dataset, containing one class with 32 malignant patterns and one class with 30 benign patterns, each pattern containing 40 features, and (iii) the US-DM dataset, containing one class with 32 malignant patterns and one class with 30 benign patterns, each pattern containing 80 features. A full list of textural features employed is presented in Table 5.2.

Table 5.2: List of textural features employed in the current study along with their abbreviations for DM and US

Histogram	Grey Level Co-occurrence Matrices*	Grey Level Run Length Matrices*
Mean Value (MV)	Angular Second Moment (ASM)	Grey Level Non Uniformity (GLNU)
Standard Deviation (SD)	Contrast (CONTR)	Run Length Non Uniformity (RLNU)
Skewness (SKEW)	Correlation (COR)	Long Run Emphasis (LRE)
Kurtosis (KURT)	Autocorrelation (ACOR)	Short Run Emphasis (SRE)
	Sum of Squares (SOQ)	Run Percentage (RP)
	Inverse Difference Moment (IDM)	
	Entropy (ENTR)	
	Sum of Entropy (SENTR)	
	Sum Average (SAV)	
	Sum of Variance (SVAR)	
	Difference Variance (DVAR)	
	Difference Entropy (DENTR)	
	Variance (VAR)	

*For these features both the average and the range values over four directions (0°, 45°, 90°, 135°) were included

Figure 5.4 illustrates the flow diagram of the procedure followed for creating the three datasets. Each dataset was used separately in the design of the PR-system so as to investigate the benefit of employing multimodality imaging data.

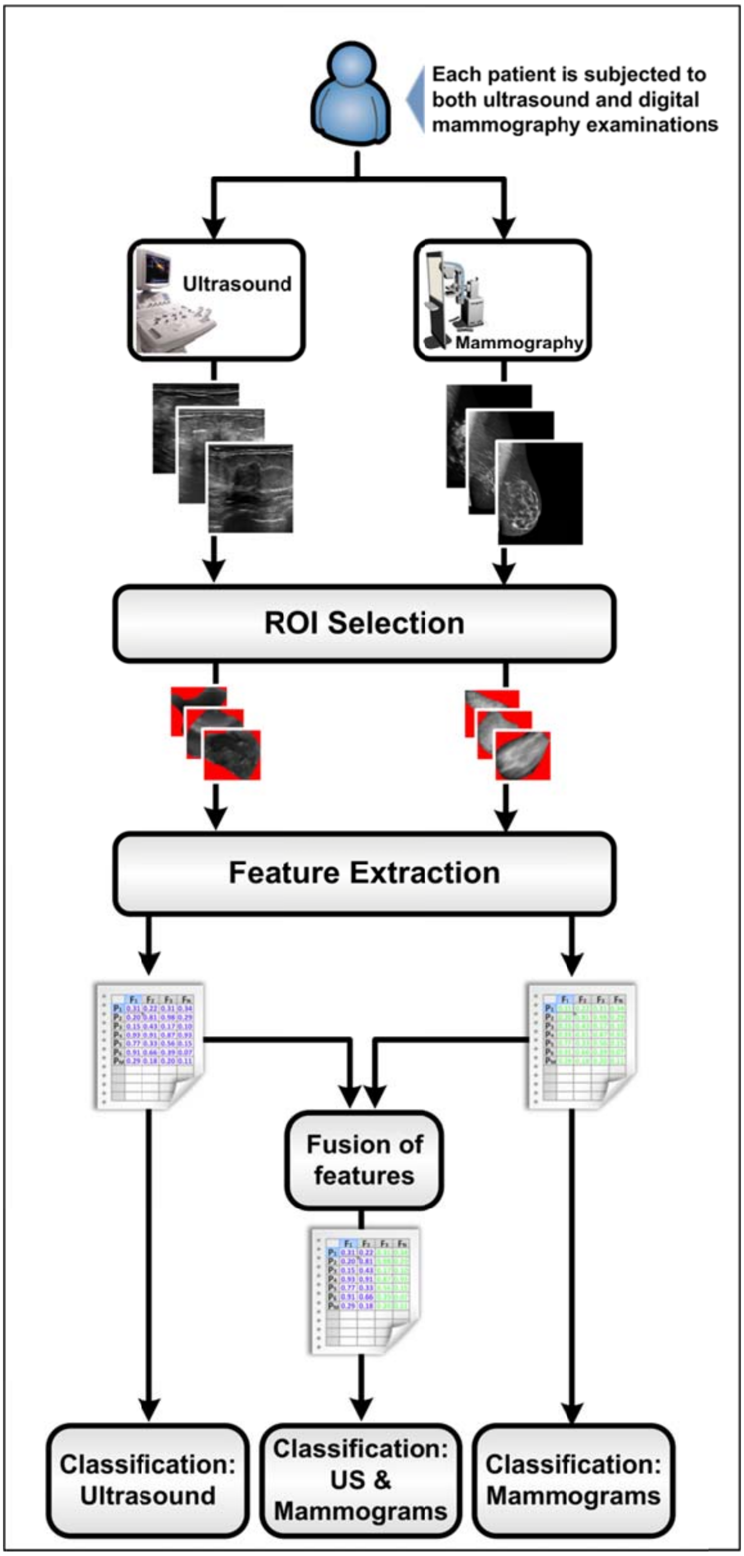

Each patient is subjected to both ultrasound and digital mammography examinations
Ultrasound
Mammography
ROI Selection
Feature Extraction
Fusion of features
Classification: Ultrasound
Classification: US & Mammograms
Classification: Mammograms

Figure 5.4: The flow diagram of the procedure followed for creating the three datasets

All features in each dataset were normalised to zero mean and unit standard deviation, according to Equation 3.26. As stated in chapter 3, feature normalisation was mandatory in order to avoid bias, since features with higher values have a stronger impact on the design of the classifier (Theodoridis and Koutroumbas, 2003).

Implementation of all textural features was performed in accordance with the explicit equations described in chapter 3.

Features were also tested for their discriminatory capacity between the two classes, by employing the non-parametric Wilcoxon statistical test (Hollander and Wolfe, 1999). Features that were found to sustain statistically significant differences between the two classes at the 0.005 statistical level ($p<0.005$) were used to draw conclusions (see chapter 6) as to existing differences in texture between the two types of lesions (i.e. benign-malignant).

5.6 Classifier Selection

The choice of the classifier, which was employed in the proposed PR-system, was influenced by the complexity of the design process. The design process involved the determination of a feature-combination with highest classification accuracy. To optimally determine this feature combination, an exhaustive search among all possible features is required. However, searching amongst all features (e.g. 80 features for the combined US-DM multimodal dataset) to find the best features-combination is, in terms of processing time, a highly demanding task. The classifier would, thus, have to be fast in execution and of high discriminatory ability. A number of candidate classifiers were considered, such as the k-NN, the Bayesian, the PNN, the LDA and the SVM. The discriminatory power of the different classifiers was tested employing the following suboptimal procedure: Features were first ranked according to the composite test given in Equation 3.29, where the Wilcoxon non-parametric test is combined with the correlation coefficient, so that highest ranked features would be of high discriminatory value and be least correlated with the rest of the features, which is a pre-requisite in the design of PR-systems (Theodoridis and Koutroumbas, 2003). Features were thus ranked in descending order and the first 10 features were tested exhaustively in the design of the classifiers. This design could be easily materialised in an ordinary desktop pc. As illustrated in Table 5.3, the Probabilistic Neural Network (PNN) proved more efficient in terms of accuracy, complexity, and processing time, for the dataset (US and DM combined) of the present study. Thus, the PNN was employed in the GPU implementation of the PR-system.

Table 5.3: The list of candidate classifiers with their respective accuracy

Classifier	Overall Accuracy LOO(%)	Number of features employed
k-NN	79	5
PNN	82	5
LDA	75	4
Bayesian	76	3
SVM (RBF)	78	4

5.7 PR System Design

PR-system design consisted of a) forming the classifier by means of Equation 3.37 employing various combinations of textural features and b) evaluating the performance of the classifier, in discriminating between benign and malignant lesions, using the available datasets and evaluation methods described in chapter 3. The aim was to determine the optimum PR-system design, i.e. obtain highest classification accuracy with the least number of textural features. Three feature selection methods were used, two sub-optimal and one optimal (Theodoridis and Koutroumbas, 2003; Gunal et al., 2009). Moreover, two evaluation methods were considered, the LOO (Theodoridis and Koutroumbas, 2003; Gunal et al., 2009) and the ECV (Ambroise and McLachlan, 2002).

5.8 Feature Selection Methods

In the first suboptimal feature selection method, as was previously mentioned, a criterion was formed for each feature, by combining the Wilcoxon statistical test (i.e.

the feature's between-classes separability criterion) and the feature's correlation with rest of the features (Theodoridis and Koutroumbas, 2003), since the least correlated features are expected to form combinations of high discriminatory power. That criterion ranked features in descending order and the first ten highly ranked features were retained for further processing. Next, the exhaustive search was employed to form all possible feature-combinations amongst the selected 10 top-ranked features. Each combination was employed to design the PNN and the classification accuracy of the design was evaluated by means of the leave-one-out method. Finally, the whole cycle of "exhaustively forming feature combinations - designing the PNN - evaluating the performance by the LOO method" eventually revealed the particular best design.

The second suboptimal features selection method utilised the sequential backward selection (SBS) technique (see chapter 3), using as class separability criterion the performance of the PNN classifier itself. The accuracy of the PNN design was tested at each feature combination formed by the SBS by means of the LOO evaluation method. Briefly, the SBS starts by selecting all n features and testing the classification accuracy by the LOO method. Then it forms all possible feature combinations of n-1 features retaining that combination with the best classification accuracy. The procedure is repeated using the latter best combination of n-1 features down to one feature. Then the combination that has provided the highest classification accuracy with the least number of features is chosen as the best design.

The optimal feature selection method consisted of forming all possible feature combinations of up to k features out of the n originally available features (see chapter 3). The number of possible combinations is enormous (see Equation 3.31), considering the number of 80 features involved in the design. To illustrate the complexity of the problem, if all possible, up to six (6) features, combinations were to be formed out of, say, forty (40) originally existing features in the US dataset alone, about 4.5 million combinations would have resulted. For each feature combination, if we were to test the accuracy of the PNN classification by the LOO method, the PNN would have to be re-designed as many times as the number of available patterns in both classes, i.e. 62 designs, to get the performance of the PNN design for each feature-combination, resulting in more than 280 million re-designs. This could only be accomplished by exploiting the parallelism of the GPU card.

5.9 PR-System Performance Evaluation

As explained in detail in chapter 3, the leave-one-out evaluation method operates by leaving one pattern out, training the classifier with the rest of the patterns, and classifying the left-out pattern. This cycle is repeated for all patterns. The class, to which the left-out pattern is assigned by the classifier, is then compared with its true class. The results of correctly and/or wrongly classified left-out patterns are presented in the form of a truth-table (Theodoridis and Koutroumbas, 2003). The latter indicates the number of correctly/incorrectly classified left-out patterns in both classes and, hence, the sensitivity (prediction rates for malignant lesions) and specificity (prediction rates for benign lesions) of the classification of the PR-system.

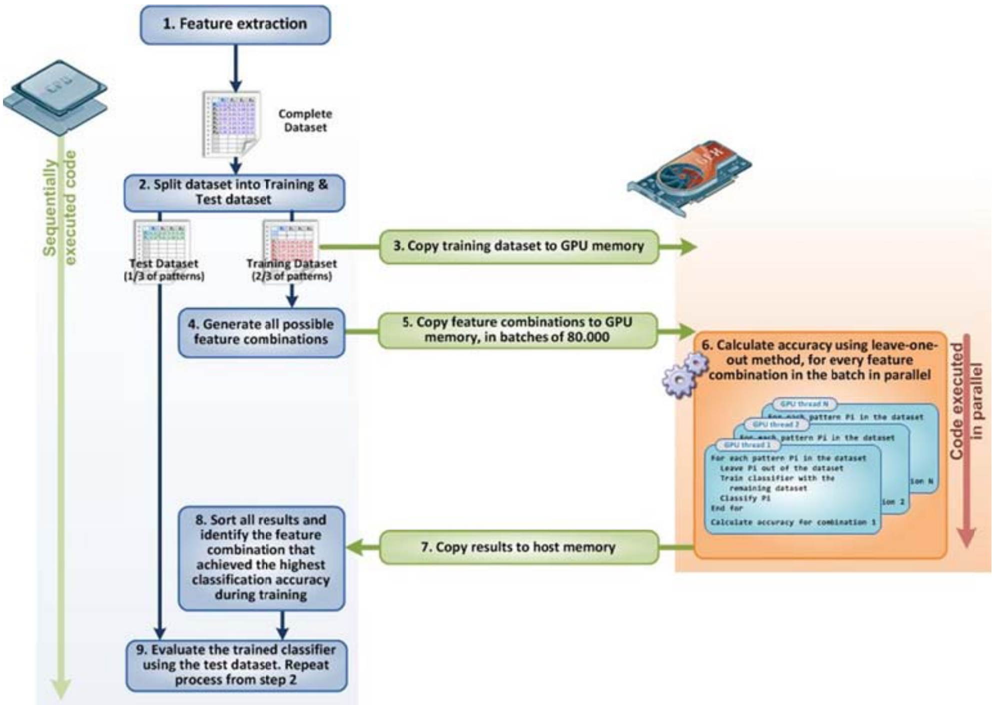
As it has been shown in a previous study (Ambroise and McLachlan, 2002), the LOO method introduces a bias in the calculation of the PR-system's performance, since the same dataset is used to design and evaluate the PR-system. The external-cross-validation method (ECV) may be adopted instead, in order to get a realistic assessment of the PR-system's performance to "unseen" data. Accordingly, the dataset is randomly split into two non-overlapping subsets: the training subset (70%), which is used to design the classifier by means of the exhaustive features search and LOO methods, and the test subset (30%), which comprises the input to the system to be classified. This process was repeated ten (10) times and the average ECV accuracy is calculated, as an assessment of the PR-system's classification accuracy to new, "unseen" by the system, data.

5.10 Parallel Processing GPU Platform

To deal with the enormous computational demands of the PR-system design, when employing the exhaustive search and the ECV methods, processing had to be transferred on to the multi-processors of the GPU card. The classifier design process had to be split into small tasks which were loaded on single GPU-threads separately, that run concurrently on the different cores of the GPU-multiprocessors.

Such single task, running on a separate GPU thread in parallel, was to design the classifier by one combination of features and then to assess the performance of the so-designed classifier by the LOO evaluation method.

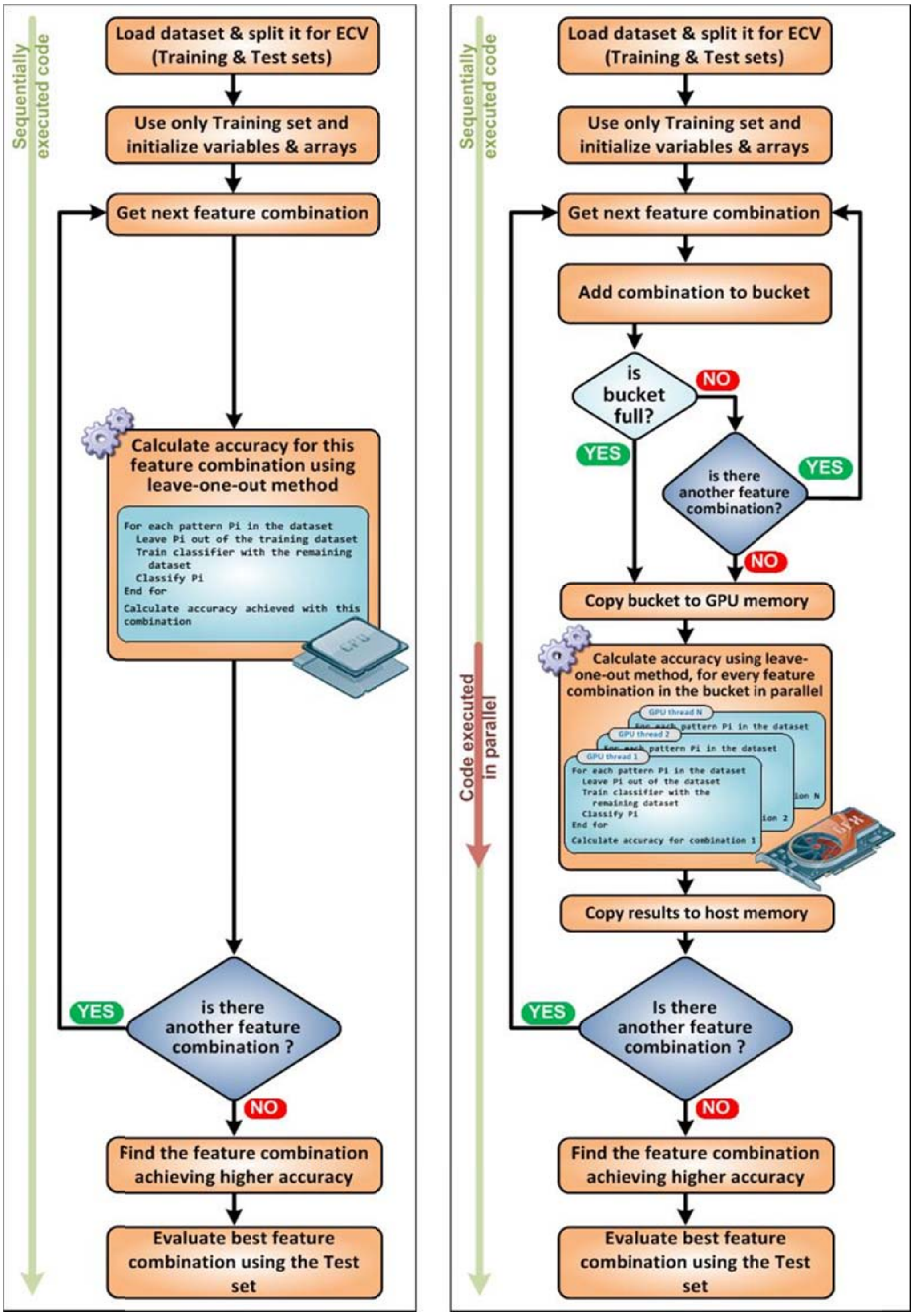
The overall design process (see Figure 5.5) included a sequential part, which is executed by a single CPU thread, and a parallel part executed exclusively by the GPU. In detail, the tasks responsible for loading of the training subset, the arrays and variable initialisation, and memory copies/transfers from host PC to GPU memory and vice versa, are all executed in sequence by the CPU (see Figure 5.6)


1. Feature extraction
Complete Dataset
Sequentially executed code
2. Split dataset into Training & Test dataset
Test Dataset (1/3 of patterns)
Training Dataset (2/3 of patterns)
3. Copy training dataset to GPU memory
4. Generate all possible feature combinations
5. Copy feature combinations to GPU memory, in batches of 80.000
6. Calculate accuracy using leave-one-out method, for every feature combination in the batch in parallel
GPU thread N
GPU thread 2
GPU thread 1
Code executed in parallel
7. Copy results to host memory
8. Sort all results and identify the feature combination that achieved the highest classification accuracy during training
9. Evaluate the trained classifier using the test dataset. Repeat process from step 2

```
For each pattern Pi in the dataset
  Leave Pi out of the dataset
  Train classifier with the
    remaining dataset
  Classify Pi
End for

Calculate accuracy for combination 1
```

Figure 5.5: Block diagram describing the training of the proposed PR system, steps 1, 2, 4, 8 and 9 are sequentially executed on the CPU; steps 3, 5 and 7 refer to memory transactions between CPU and GPU; and step 6 is executed in parallel on the GPU, employing numerous threads.


Sequentially executed code
Load dataset & split it for ECV (Training & Test sets)
Use only Training set and initialize variables & arrays
Get next feature combination
Calculate accuracy for this feature combination using leave-one-out method
CPU
YES
is there another feature combination ?
NO
Find the feature combination achieving higher accuracy
Evaluate best feature combination using the Test set
Sequentially executed code
Code executed in parallel
Load dataset & split it for ECV (Training & Test sets)
Use only Training set and initialize variables & arrays
Get next feature combination
Add combination to bucket
is bucket full?
NO
YES
is there another feature combination?
YES
NO
Copy bucket to GPU memory
Calculate accuracy using leave-one-out method, for every feature combination in the bucket in parallel
GPU thread N
GPU thread 2
GPU thread 1
pattern Pi in the dataset
ion N
ion 2
GPU
Copy results to host memory
YES
Is there another feature combination ?
NO
Find the feature combination achieving higher accuracy
Evaluate best feature combination using the Test set

```
For each pattern Pi in the dataset
  Leave Pi out of the training dataset
  Train classifier with the remaining
    dataset
  Classify Pi
End for
Calculate accuracy achieved with this
combination
```

```
For each pattern Pi in the dataset
  Leave Pi out of the dataset
  Train classifier with the
    remaining dataset
  Classify Pi
End for
Calculate accuracy for combination 1
```

Figure 5.6: The flow diagram of the PR-system design task. Left part illustrates the sequentially executed task on CPU. Right part illustrates the same task as it has been modified to run in parallel on GPU

Regarding the kernel code running in all GPU SPs in parallel, each GPU thread is fed with a single feature combination as input and is assigned with the task to train the PNN classifier with this unique feature combination and evaluate its classification accuracy by means of the LOO method as presented in Figure 5.7. A pseudocode version of the proposed implementation is given in Figure 5.8.

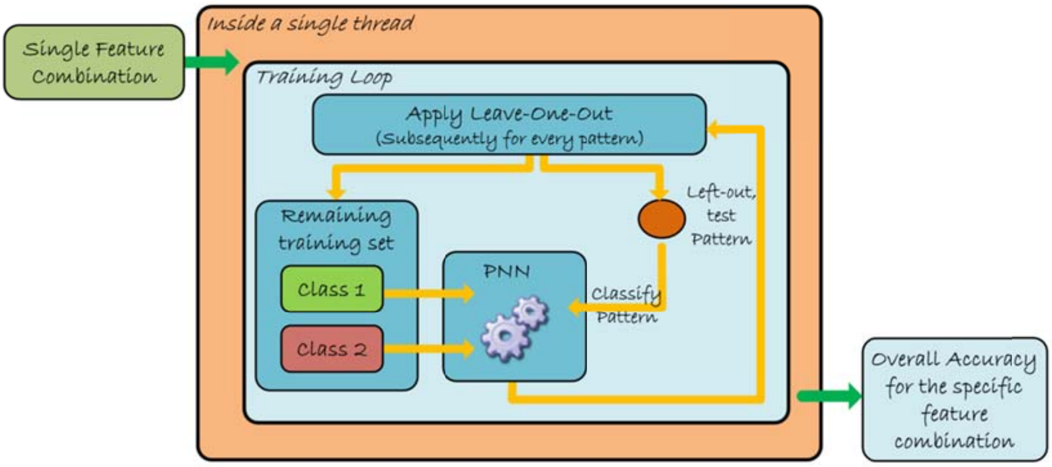

Single Feature Combination
Inside a single thread
Training Loop
Apply Leave-One-Out (Subsequently for every pattern)
Remaining training set
Class 1
Class 2
PNN
Left-out test Pattern
Classify Pattern
Overall Accuracy for the specific feature combination

Figure 5.7: The task of each thread, running concurrently, was to train the PNN classifier with a unique feature combination and evaluate its classification accuracy by means of the leave-one-pattern-out technique. In every case the output of each thread was the overall accuracy achieved for the specific feature combination.

```
Read feature combination $C_t$ $[F_1, F_2, F_3 \ldots F_n]$ from Device global memory
FOR each pattern $P_{j,i}$ in the training dataset
    SET $P_{j,i}$ as the test pattern $X$
    Leave $P_{j,i}$ out of the training dataset
    Train PNN with the remaining dataset and using feature combination $C_t$
    Classify test pattern $X$ and UPDATE truth table in local memory
ENDFOR
Calculate overall accuracy $A_t$ for feature combination $C_t$
Store $A_t$ in Device global memory
```

Figure 5.8: Pseudocode illustrating the PNN training in each CUDA thread *t* for a single feature combination.

Figure 5.5 shows a schematic diagram of the parallel computation of the tasks, involving the GPU. For the parallel processing set-up to function on the GPU processors, a large bunch of feature combinations were first organised in a batch and were transferred from the host CPU to the global memory of the GPU. There, the feature combinations were loaded onto an equal number of GPU-threads, where the task of designing the classifier and evaluating its precision by the LOO method was executed concurrently.

All possible feature combinations up to length k were generated employing an algorithm inspired by Donald Knuth (Koepf, 2009). Decomposing the task of PR design in this way, led to maximising the portion of the problem that could be processed in parallel; that parallel portion constituted 99.99% of the program and thus rendered plausible the achievement of high acceleration in performance. Features' normalisation in each concurrently running thread had been a priori calculated on the CPU to avoid excess global memory access.

As illustrated in Figure 5.6, feature combinations were transferred from the host PC memory onto the GPU's global memory in batches of 80,000 feature-combinations, right before each GPU kernel launch. For example, the first batch transfer contained combinations of one, two, three, etc. features until the batch-size of 80,000 combinations was filled. Regarding thread configuration, Block and Grid sizes were experimentally set to [128 x 1] and [625 x 1], respectively, while the batch size was selected so as to be a multiple of the block size (128 threads). The selected block size allowed 4 warps per SM. Upon completion of all threads in the launch, results were transferred back to the host PC memory and the kernel was launched for the next batch of 80,000 feature combinations. The cycle of batch transfer of feature combinations and parallel processing was repeated until all tasks of the cycle were completed (see Figure 5.9). Especially for the Fermi-class GPU, the number of feature-combinations in each batch was set to 160,000, which are organised in 1250 blocks of 128 threads each.

```
Read training dataset
Copy training dataset to Device Memory
Set S the maximum size of batch
REPEAT
WHILE there is another features combination = TRUE AND batch size <S
        Generate next feature combination
        Add combination to batch
        Increment batch size
ENDWHILE
Copy batch to Device Memory
Launch PNN Kernel on Device using S threads*
Wait until all threads in batch have finished
Copy results to Host Memory
Empty batch
UNTIL there is another features combination = FALSE
Identify the feature combination that achieved the highest classification accuracy during training
```

Figure 5.9: Pseudocode illustrating Host implementation not including the ECV method

For the ECV evaluation method, a number of similar training tasks was executed, where the whole ECV-cycle of classifier training-classification had to be repeated ten times, with guidance from the host-PC's CPU. Specifically for the ECV, the tasks responsible for loading of the complete dataset, division of dataset in training and test subsets, and memory copies/transfers from host PC to GPU memory and vice versa, were executed in sequence by the CPU.

GPU cards used were NVIDIA’s GeForce 8800GT featuring 112 CUDA cores and GeForce 580GTX featuring 512 CUDA cores. Their specifications are presented in detail in the Table 5.4. Figure 5.10 displays the kernel used to design the proposed GPU-based PR system

Table 5.4: Specifications of the GPUs employed in this research

Model	GeForce 8800 GT	GeForce GTX 580
Number of multiprocessors	14	16
Cores per multiprocessor	8	32
Number of CUDA cores	112	512
Total global memory	512 MB GDDR3	1536 MB GDDR5
Shared memory per block	16 KB	48 KB
Memory interface width	256 bits	384 bits
Memory clock/Data Rate	920 MHz/1840 MHz	2100 MHz/4200 MHz
Warp size	32	32
Core clock/Shader clock	700 MHz/1715 MHz	832 MHz/1664 MHz
CUDA compute capability	1.1	2.0

```
__global__ void
GPUSingleCombination_LOO_PNN(
// Pointer to the dataset stored in global memory
float* features,
// Pointer to the mean values lookup table stored in global memory
float* meanFeatures,
// Pointer to the standard deviations lookup table stored in global memory
float* stdvFeatures,
// Size of dataset - Number of patterns
int D_SIZE,
// Defines the point in dataset where class 2 begins
int CDIVIDER,
// Max number of features
int P_SIZE ,
// Pointer to all possible feature combinations stored in global memory
int* combination ,
// Pointer to the size of each feature combination
int* combinationSize,
// Pointer to an array used to store the overall accuracy
// for each possible feature combination
float* overallAccuracy )
{
  //Unique thread identifier
  int index = (blockIdx.x * blockDim.x) + threadIdx.x;
  //Various variables
  int i, k, c1;
  unsigned int  combK;
  float unknownP[20];
  int class1Count, class2Count;
  float p = 2.0f;
  float sigma = 0.24f;
  float pi = 3.14159f;
  float distance, distanceInP;
  float denominator = 2.0f * powf(sigma,2);
  float denominatorP = 0.0f;
  float tempSum, tempExpSum, expvar;
  int classSelected=0;
  float g1 = 0.0;
  float g2 = 0.0;
  int truthTable[2][2];
  truthTable[0][0] =0;
  truthTable[0][1] =0;
  truthTable[1][0] =0;
  truthTable[1][1] =0;

  //Get the size of the specific combination
  unsigned int COMBINATION_SIZE = combinationSize[index];

  // For all patterns
  // Apply Leave one out method
  for(i=0; i<D_SIZE; i++){

    classSelected=1;
     if(i<CDIVIDER)
        classSelected=0;
     //If 0 then patterns belongs to Class 1 else it belongs to Class 2
     // Copy unknown pattern from global memory and normalise it
     for(k=0; k<P_SIZE; k++){
        unknownP[k] = ( features[(i * P_SIZE) + k]
                - meanFeatures[(i * P_SIZE) + k] )  / (float) stdvFeatures[(i * P_SIZE) + k];
     }
     // Calculate discriminant PNN function for Class 1
     class1Count = 0;
     tempExpSum = 0.0;
     for(c1=0; c1<CDIVIDER; c1++) {

        if(c1!=i){ // NOT the left-out pattern
           class1Count++;
           // Calculate square distance
           tempSum = 0.0;
           for(k=0; k<COMBINATION_SIZE; k++){
              combK = combination[ (index * P_SIZE) + k ];
             distance =  unknownP[ combK ] - ( ( features[ (c1 * P_SIZE) + combK ] -
                          meanFeatures[ (i * P_SIZE) + combK  ] ) / (float)
                          stdvFeatures[  (i * P_SIZE) + combK  ]  );
             distanceInP = powf( distance , 2);
             tempSum = tempSum + distanceInP;
           }
           expvar = expf( - tempSum/denominator);
           tempExpSum = tempExpSum + expvar;

        }//if(c1!=i)

     }

     denominatorP = powf( 2.0f * pi , (COMBINATION_SIZE/2) ) * powf( sigma ,
COMBINATION_SIZE);
     g1 = tempExpSum / ( denominatorP * class1Count);
     // Calculate discriminant PNN function for Class 2
     class2Count = 0;
     tempExpSum = 0.0;
     for(c1=CDIVIDER; c1<D_SIZE; c1++) {
        if(c1!=i){ // NOT the left-out pattern
           class2Count++;
           // Calculate square distance
           tempSum = 0.0;
           for(k=0; k<COMBINATION_SIZE; k++){
              combK = combination[ (index * P_SIZE) + k ];
                          distance =  unknownP[ combK ] - ( ( features[ (c1 * P_SIZE) + combK
] -
                          meanFeatures[ (i * P_SIZE) + combK  ] ) / (float)
                          stdvFeatures[  (i * P_SIZE) + combK  ]  );
              distanceInP = powf( distance , 2);
              tempSum = tempSum + distanceInP;
           }
           expvar = expf( - tempSum/denominator);
           tempExpSum = tempExpSum + expvar;

        }//if(c1!=i)
     }
     denominatorP   =   powf(   2.0f   *   pi   ,   (COMBINATION_SIZE/2)   )   *   powf(   sigma   ,
COMBINATION_SIZE);
     g2 = tempExpSum / ( denominatorP * class2Count);
     //Fill up the truth table
     if(g1>=g2){
        truthTable[classSelected][0]++;
     }
     else{
        truthTable[classSelected][1]++;
     }
  }
  //Calculate OverallAccuracy for this feature combination
  float OverallAccuracy = 0.0f;
  OverallAccuracy = (truthTable[0][0] + truthTable[1][1]) * 100
        / (truthTable[0][0] + truthTable[0][1] + truthTable[1][0] + truthTable[1][1] );
  //Update the value in global memory
  overallAccuracy[index] = OverallAccuracy;
  //Wait for all threads to reach this point
  __syncthreads();
}
```

Figure 5.10: The kernel used to design the proposed GPU-based PR system

5.11 Summary

This chapter described in detail and in a systematic way the steps followed throughout the whole PR-system design. Data collected included images of 62 patients with histologically verified (benign or malignant) breast lesions, who underwent both ultrasound (US) and digital mammography (DM) examinations. Lesions were outlined on the images by an experienced radiologist employing a custom-made software tool, and 40 features were calculated from the texture of the outlined lesions, for each one of the two modalities used. Thus each patient was represented by an 80 features vector. A PR-system was designed to provide highest possible precision by programming in parallel the multiprocessors of the NVIDIA's GPU-cards, GeForce 8800GT or 580GTX, and using the CUDA programming framework and C++. The PR-system was built around the Probabilistic Neural Network classifier and its performance was evaluated by the LOO method, for estimating the system's highest accuracy, and by the ECV method, for assessing the PR-system's unbiased accuracy to new, "unseen" by the system, data.

Chapter 6

System Evaluation

6.1 Overview

In this chapter, results are presented regarding the achieved classification accuracy and GPU acceleration on system design. Additionally, textural features involved in the best design feature combination are analysed as to the information they convey and their significance in medical diagnosis of breast cancer.

6.2 Evaluation of System Accuracy

Table 6.1 shows the classification accuracies achieved by designing the PR-system with the sub-optimal and optimal feature selection methods that were investigated, using features extracted from US images only. The first two columns of Table 6.1 indicate the precisions of the PR-system in classifying the benign and malignant lesions respectively, the third column shows the overall system accuracies, and the fourth column shows the best feature combinations for the design of the PNN, regarding the three different design methods (two suboptimal and one optimal). As it may be observed, highest classification accuracies were achieved by the optimal feature selection method (bottom row: 86.7%, 84.4%, 85.5%), using the PNN classifier, the exhaustive search, for testing all possible features combinations, and the LOO evaluation method on the processors of the GPU. Examining the benign lesions column, precisions were 76.7%, 83.3% and 86.7%, using the 1st sub-optimal, 2nd sub-optimal, and optimal features selection methods, corresponding to 23, 25,

Table 6.1: Classification accuracy results for discriminating between benign and malignant lesions from ultrasound images, using the PNN classifier, two sub-optimal and one optimal features-selection methods, and the LOO evaluation method

	Benign (%)	Malignant (%)	Overall Accuracy (%)	Features
1st suboptimal: Wilcoxon-rank-features & LOO	76.7 (23/30)	81.3 (26/32)	79.0 (49/62)	DENTR^r, RP^a
2nd suboptimal: SBS & LOO	83.3 (25/30)	71.9 (23/32)	77.4 (48/62)	ACOR^a, CONTR^r, COR^r, DENTR^r, LRE^a, SD, SKEW, SOQ^a
Optimal: Exhaustive search & LOO on GPU	86.7 (26/30)	84.4 (27/32)	85.5 (50/62)	DVAR^a, RLNU^r, RP^a

ACOR: Autocorrelation, CONTR: Contrast, COR: Correlation, DENTR: Difference Entropy, DVAR: Difference Variance, LRE: Long Run Emphasis, RLNU: Run Length Non Uniformity, RP: Run Percentage, SD: Standard Deviation, SKEW: Skewness, SOQ: Sum of Squares, a: average, r: range and 26 out of 30 correctly classified benign lesions, respectively. Similar findings may be observed in the malignant lesions column: 81.3%, 71.9%, and 84.4% accuracies were achieved in correctly characterising malignant lesions, assigning 26, 23 and 27 out of 32 lesions to the correct classes, respectively. Overall accuracies (third column of Table 6.1) were 79.0% (49/62), 77.4% (48/62), and 85.5% (50/62) respectively. In more detail, as it may be observed in Table 6.4 and regarding the optimal features selection method, four (4) benign lesions were mistaken by the PR-system for malignant and five (5) malignant lesions for benign, leading to nine (9) lesions classified to the wrong classes.

Regarding lesions found in DM images, similar findings may be observed in Table 6.2. Highest classification accuracies were achieved by the optimal features selection method (bottom row); 83.3%, 81.3%, and 82.3% in correctly classifying benign, malignant, and discriminating between the two classes of lesions, respectively. As it may be observed in more detail in Table 6.4, eleven (11) lesions were classified to the wrong classes, five (5) benign were mistaken for malignant and six (6) malignant were characterised as benign.

Table 6.2: Classification accuracy results for discriminating between benign and malignant lesions from DM images, using the PNN classifier, two sub-optimal and one optimal features-selection methods, and the LOO evaluation method

	Benign (%)	**Malignant (%)**	**Overall Accuracy (%)**	**Features**
1st suboptimal: Wilcoxon-rank-features & LOO	73.3 (22/30)	75.0 (24/32)	74.2 (46/62)	GLNU^r, SOQ^a, VAR^r
2nd suboptimal: SBS & LOO	73.3 (22/30)	78.1 (25/32)	75.8 (47/62)	ACOR^a, GLNU^r, VAR^r
Optimal: Exhaustive search & LOO on GPU	83.3 (25/30)	81.3 (26/32)	82.3 (51/62)	DVAR^a, GLNU^r, SAV^a, VAR^a, VAR^r

ACOR: Autocorrelation, DVAR: Difference Variance, GLNU: Grey Level Non Uniformity, SAV: Sum Average, SOQ Sum of Squares, VAR Variance,

a: average, r: range

Table 6.3: Classification accuracy results for discriminating between benign and malignant lesions using combined features from ultrasound (US) and digital mammography (DM) images, and employing the PNN classifier, two sub-optimal and one optimal features-selection methods, and the LOO evaluation method

	Benign (%)	**Malignant (%)**	**Overall Accuracy (%)**	**Features**
1st suboptimal: Wilcoxon-rank-features & LOO	80.0 (24/30)	87.5 (28/32)	83.9 (52/62)	SENTR^r_DM, KURT_US, DENTR^r_US, RP^a_US
2nd suboptimal: SBS & LOO	73.3 (22/30)	90.6 (29/32)	82.3 (51/62)	MV_DM, SKEW_DM, ASM^r_DM, SVAR^a_DM, SKEW_US, DENTR^r_US, RP^a_US
Optimal: Exhaustive search & LOO on GPU	90.0 (27/30)	96.9 (31/32)	93.5 (58/62)	SD_DM, SOQ^r_DM, SAV^a_DM, SRE^a_DM, DENTR^r_US, SRE^a_US

ASM: Angular Second Moment, DENTR: Difference Entropy, KURT: Kurtosis, MV: Mean Value, RP: Run Percentage, SAV: Sum Average, SD: Standard Deviation, SENTR: Sum of Entropy, SKEW: Skewness, SOQ: Sum of Squares, SRE: Short Run Emphasis, SVAR: Sum of Variance, a: average, r: range, _DM: features from DM images and _US: feature from US images

Combining features from the two modalities, US and DM, classifications improved, with the optimal features selection method achieving highest accuracies, 90%, 96.9%, and 93.5% (bottom row in Table 6.3) in correctly classifying benign, malignant, and discriminating between the two classes of lesions, respectively. As it may be observed in more detail in Table 6.4, four (4) lesions were misclassified, three (3) benign were classified as malignant and one (1) malignant was characterised as benign. In particular, those misclassified cases comprised three (3) Fibroadenomas and one (1) Infiltrating Ductal Carcinoma. These high accuracy results were achieved using four (4) features from DM (standard deviation, sum of squares, sum average, short run emphasis) and two (2) features from US (difference entropy and short run emphasis). Figure 6.1 combines data from Tables 6.1, 6.2 and 6.3 and presents the overall accuracies achieved by the two sub-optimal and one optimal features-selection methods, and the LOO evaluation method using the three datasets.

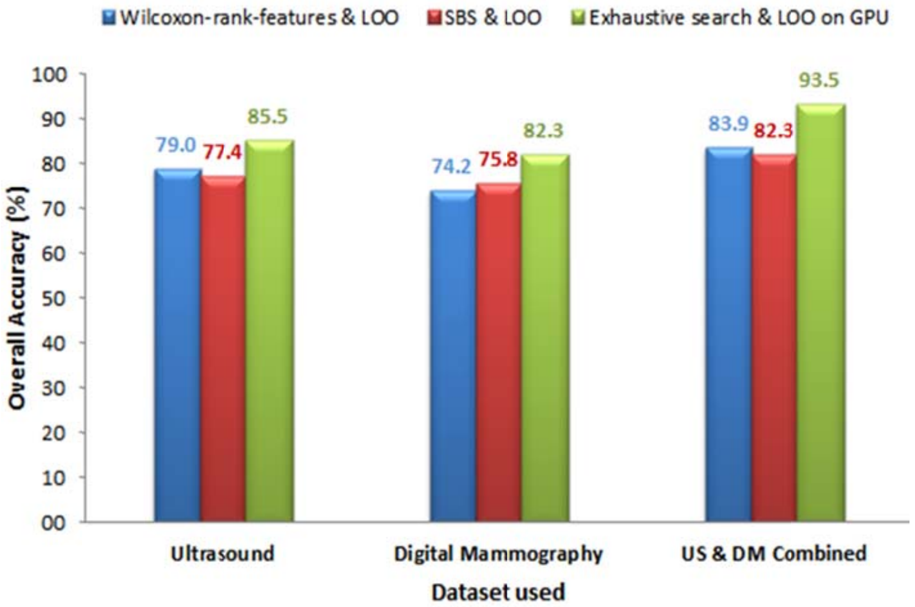


Figure 6.1: Overall accuracies achieved by the two sub-optimal and one optimal features-selection methods, and the LOO evaluation method using the three datasets (US, DM and US & DM combined)

Table 6.4: Truth table of GPU results using PNN, exhaustive search, and LOO for features from US images, features from DM images, and combined features from US and DM images

		Verified Benign	**Verified Malignant**	**Partial Accuracies (%)**	**Overall Accuracy (%)**
Features from US images	Benign	26	4	86.7	85.5
	Malignant	5	27	84.4	
Features from DM images	Benign	25	5	83.3	82.3
	Malignant	6	26	81.3	
Combined features from US and DM images	Benign	27	3	90.0	93.5
	Malignant	1	31	96.9	

Partial and overall accuracies achieved via the optimal feature selection method, applied on GPU, for all three datasets are graphically illustrated in Figure 6.2.

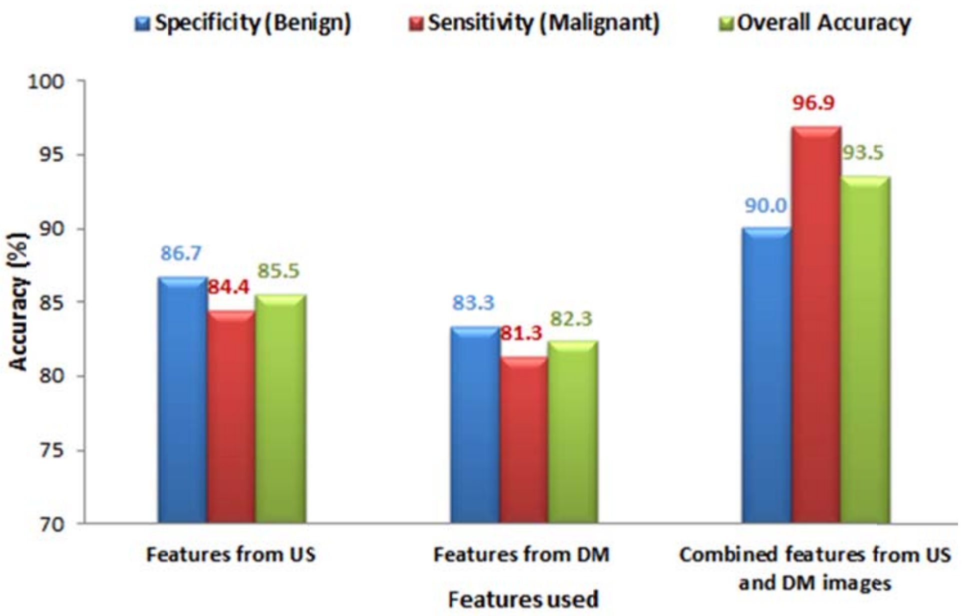


Figure 6.2: Partial and overall accuracies achieved via the optimal feature selection method and LOO, applied on GPU, using features from US images, features from DM images, and combined features from US and DM images

Figure 6.3 (a), (b), and (c) demonstrate transformed scatter diagrams (by quadratic least-squares rule), of the best feature vectors in the optimal design method, using US, DM, and both US and DM textural features, respectively. Those scatter diagrams have transformed the best feature combinations spaces into two-dimensional spaces (dimensionality is equal to the number of classes) by the quadratic least squares (QLS) mapping process (Ahmed and Rao, 1975). Triangles and stars correspond to benign and malignant lesions, respectively.

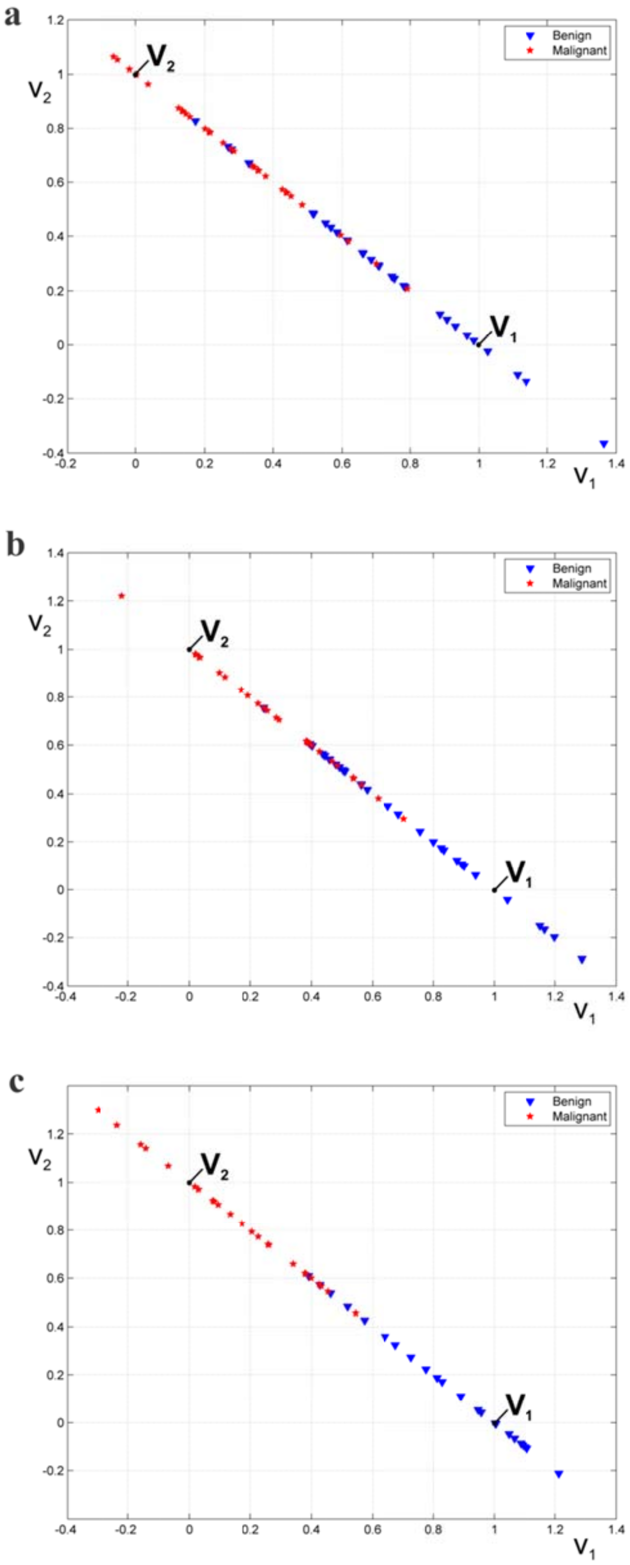


Figure 6.3: Quadratic LS-mappings of benign and malignant data, employed in the optimal PNN design using (a) US, (b) DM, and (c) combined US and DM features

Figure 6.4 shows boxplot diagrams of the features used in the optimal classifier design. The median of the feature values are represented by the horizontal lines within the boxes. Of the six features from DM dataset (standard deviation, sum of squares (range), sum average (mean), short run emphasis (mean), and from ultrasound: difference entropy (range) and short run emphasis (mean)), only one, the difference entropy (range) sustained statistically significant difference ($p<0.005$) between the two classes of lesions.

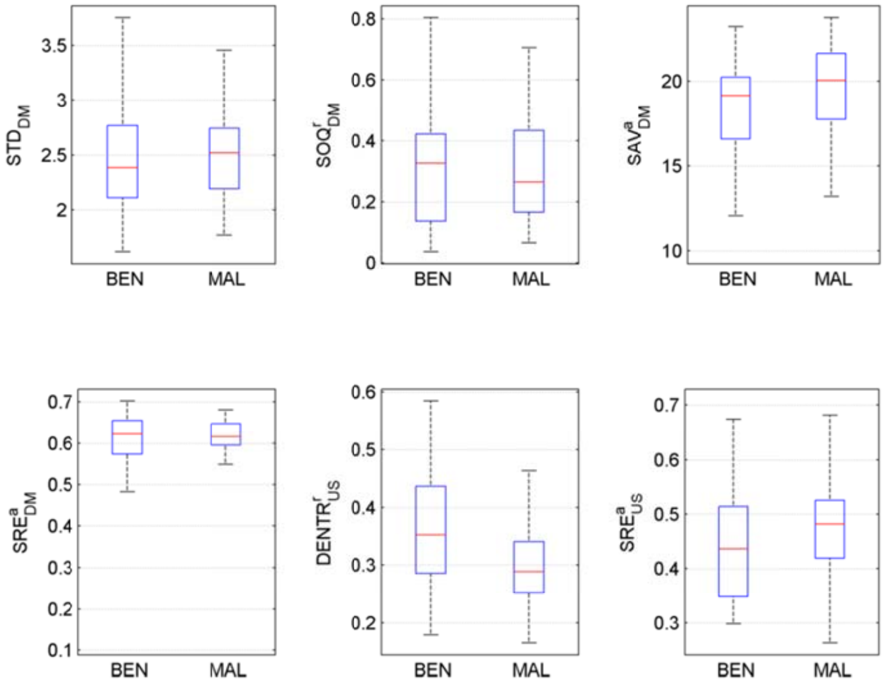


Figure 6.4: Box-plots of the six textural features that formed the best feature combination in the optimal PNN design. The features' values are not normalised. (STD: Standard Deviation, SOQ: Sum of Squares, SAV: Sum Average, SRE: Short Run Emphasis, DENTR: Difference Entropy, where a: average, r: range, _DM: features from DM images and _US: feature from US images.

Table 6.5 shows the classification results obtained when the PR-system was tested by the 10-fold external cross-validation method, using combined US and DM textural features. ECV provides an assessment of the performance of the optimally designed system to new, “unseen” by the system, data. At each one of the ten trials, the accuracies achieved for the benign, malignant, and the overall accuracies are presented. The last row of Table 6.5 shows the means and standard deviations of the accuracies achieved for the benign (77 ± 8.2) and malignant (85 ± 10.8 %) lesions and the overall accuracy (81 ± 6.1%).

Table 6.5: Classification accuracies in discriminating benign from malignant lesions, achieved by employing the ECV method on GPU, and by designing the PNN classifier by combined features from ultrasound and digital mammography features

Trial	Benign (%)	Malignant (%)	Overall Accuracy (%)
1	70	100	80
2	90	90	75
3	90	90	95
4	70	80	80
5	70	100	80
6	70	80	75
7	80	80	95
8	80	70	80
9	80	70	75
10	70	90	75
Mean ± Std	77.0 ± 8.2	85.0 ± 10.8	81.0 ± 6.1

To verify that the optimal features selection method was better than the two sub-optimal methods, the ECV was also employed for the two suboptimal methods. This was done in order to discard any doubts that a suboptimal method could have been equally employed instead of the optimal method in the design of the PR-system. Tables 6.6 and 6.7, show the results by the ECV performance evaluation method, employing the Wilcoxon rank-features and the SBS methods, respectively. As it may be observed, the mean accuracies (last rows) achieved by the PR-system, were inferior in the case of the sub-optimal methods (72.5% and 62.5% for the rank-features and SBS respectively) than the accuracies reached by the PR-system employing the optimal features selection method (81%).

Table 6.6: ECV Wilcoxon-rank-features Classification accuracies in discriminating benign from malignant lesions, achieved by employing the ECV method on GPU, and by designing the PNN classifier by combined features from US and DM features

Trial	Benign (%)	Malignant (%)	Overall Accuracy (%) (Number of Features)
1	33.3	100.0	66.7 (3)
2	33.3	83.3	58.3 (4)
3	66.7	83.3	75.0 (4)
4	66.7	100.0	83.3 (4)
5	66.7	66.7	66.7 (5)
6	66.7	83.3	75.0 (5)
7	83.3	83.3	83.3 (3)
8	66.7	83.3	75.0 (4)
9	66.7	83.3	75.0 (3)
10	33.3	100.0	66.7 (4)
Mean ± Std	58.3 ± 18.0	86.7 ± 10.5	72.5 ± 7.9

Table 6.7: ECV SBS Classification accuracies in discriminating benign from malignant lesions, achieved by employing the ECV method on GPU, and by designing the PNN classifier by combined features from US and mammography features

Trials	Benign (%)	Malignant (%)	Overall Accuracy (%) (Number of Features)
1	100.0	83.3	91.7 (7)
2	50.0	83.3	66.7 (8)
3	66.7	50.0	58.3 (8)
4	33.3	66.7	50.0 (8)
5	66.7	50.0	58.3 (3)
6	50.0	100.0	75.0 (9)
7	66.7	66.7	66.7 (9)
8	66.7	50.0	58.3 (6)
9	33.3	66.7	50.0 (7)
10	50.0	50.0	50.0 (10)
Mean ± Std	58.3 ± 19.6	66.7 ± 17.6	62.5 ± 13.2

6.3 Evaluation of GPU acceleration on System Design

In order to evaluate the performance of the developed GPU-based PR-system, a series of experiments on two different NVidia GPU cards (GeForce 8800GT and 580GTX) was performed and the required training time was measured. The latter was then compared with the respective training time of the same classifier running on a typical CPU (Intel Core 2 Quad) and programmed in C programming language. Hence, both systems were trained with a dataset, consisting of 62 patterns that progressively increased their feature dimensions, beginning with 20 features and reaching 80 by a step of 10. In every case, all feature combinations up to size of 6 were exhaustively examined. Table 6.8 presents this comparative performance assessment illustrating in detail the design time required by both the GPU and CPU-based PR-system varying the number of features exhaustively examined. The GPU-based PR-system achieved about 25 times faster design, on the 8800GT and about 150 times faster on the 580GTX. Considering the problem at hand, for eighty features

Table 6.8: Comparative performance assessment varying the total number of features in the dataset (all time measurements have been averaged over 10 runs)

Total Number of Features [Number of features exhaustively combined]	Number of unique feature combinations	PNN Training Time for 62 patterns (s)								
		CPU	GPU #1: GeForce 8800GT			CPU/GPU#1 Training Time	GPU #2: GeForce 580GTX			CPU/GPU#2 Training Time
			Processing	Transfer	Total		Processing	Transfer	Total	
20[6]	60459	150.80	5.98	0.02	6.01	25.11	0.98	0.01	0.99	151.62
30[6]	768211	1978.94	78.22	0.20	78.42	25.24	11.91	0.13	12.04	164.30
40[6]	4598478	11990.42	473.68	1.39	475.07	25.24	80.14	0.60	80.73	148.52
50[6]	18260635	48129.67	1973.77	6.64	1980.41	24.30	325.05	2.48	327.53	146.95
60[6]	56049057	>>24h	6438.80	27.61	6466.41	N/A	1007.56	8.54	1016.10	N/A
70[6]	144193119	>>24h	16806.96	79.43	16886.39	N/A	2656.10	25.16	2681.26	N/A
80[6]	326207196	>>24h	38907.82	199.58	39107.40	N/A	6210.93	65.81	6276.74	N/A

available and sixty two lesions (benign and malignant), it took 39,107 secs (or about 10 hours) on the GeForce 8800GT, 6276 secs (or less than 2 hours) on the 580GTX, against much more than 24 hours on the CPU. Figure 6.5 shows comparative processing time requirements for the two GPUs and the CPU for accomplishing same tasks.

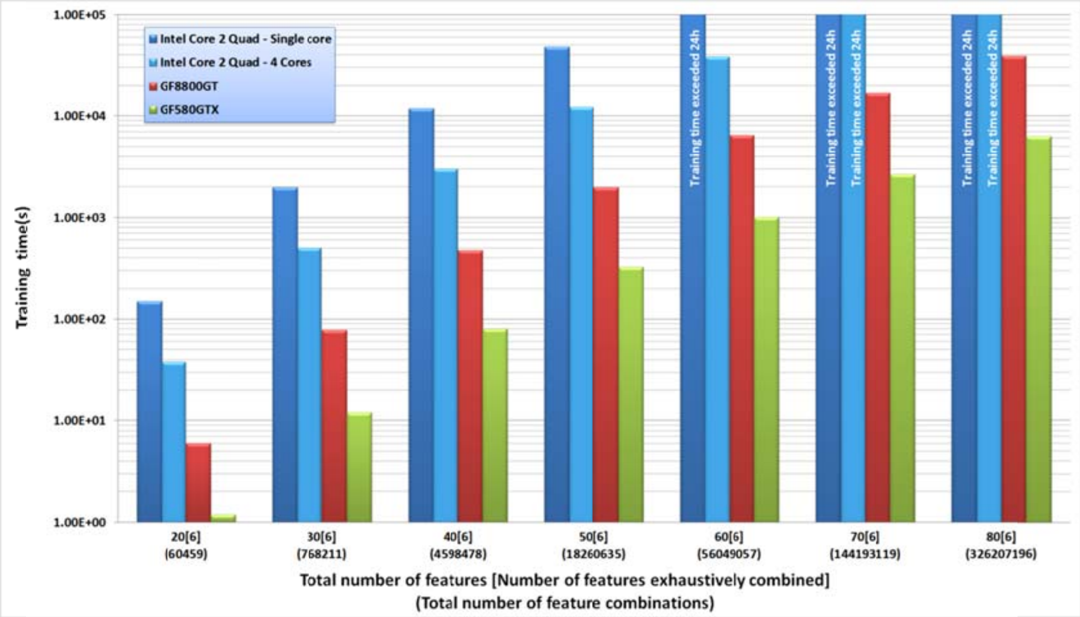


Figure 6.5: Comparative evaluation of processing times for 2 different NVidia GPU cards (GeForce 8800GT and 580GTX) and standard CPU system (Intel Core 2 Quad). Times refer to the processors (GPUs or CPU) executing the same task, by forming all possible combinations of 6 features within a pool of 20, 30,..,or 80 features

Additionally, in order to exploit the full processing potential of the CPU too, the OpenMP framework was utilised and the time required for the same processing task employing all four CPU cores was evaluated. The CPU's multi-core time performance, for all the aforementioned training tasks, was boosted resulting in a 4-fold reduction in training time over the original code running on a single CPU core. Accordingly, in the task of exhaustively searching all possible feature combinations of up to 6 within a pool of 60 features (resulting in over 56 million unique feature combinations), PR system design required 152,415 secs (or about 42.3 hours) running on a single CPU core against 39,080 secs (or about 10.9 hours) employing all 4 CPU cores. This is still significantly lower compared to the 1,016 secs (or about 17 minutes) achieved by the GPU implementation. In the more computationally intensive tasks of exhaustively searching all possible feature combinations within a pool of 70 or even 80 features, even the 4 CPU cores, running in parallel, did not manage to complete PR design within 24 hours. Figure 6.6 presents a comparative evaluation of the performance achieved in a similarly demanding scenario, where the number of patients has grown (153 cases) and the training task of searching exhaustively for combinations of up to 20 feature combinations from a pool of 20 features is plausible, without overfitting issues. For this particular experiment, a dataset used in a previous study, containing a larger number of patterns, was employed for benchmarking purposes (Sidiropoulos et al., 2012).

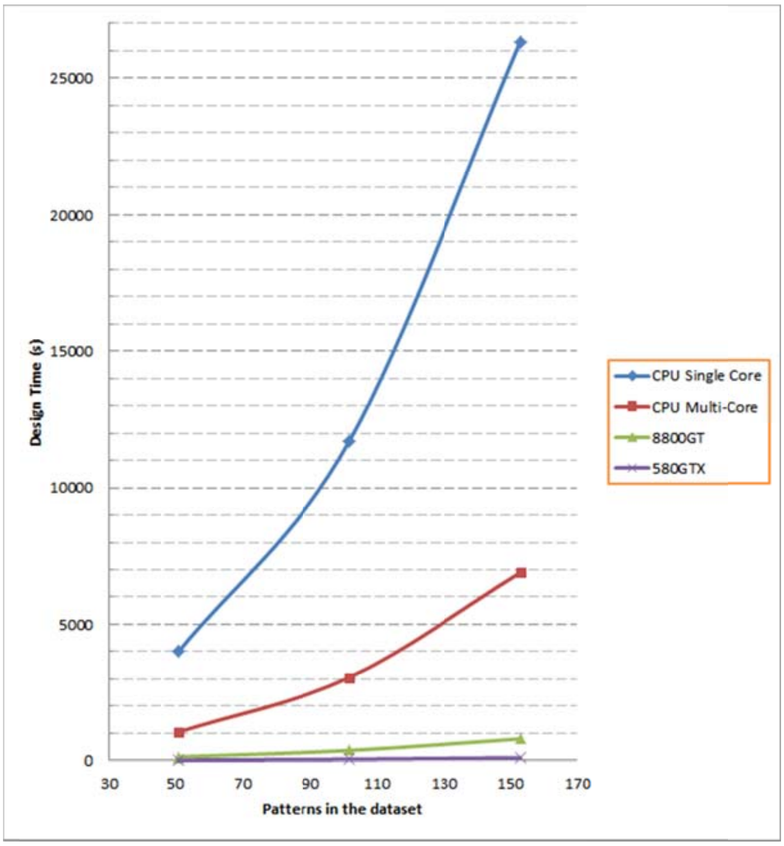


Figure 6.6: Comparative assessment of CPU (single and multi-core) and GPUs (8800GT and 580GTX) against the number of patterns in the dataset, for the task of classifier training with up to 20 feature combinations from a pool of 20 features

Figure 6.6 presents training times achieved by the host CPU (single and multi-core) and the GPUs (8800GT and 580GTX) against the number of patterns included in the dataset. As it is illustrated, the aforementioned task requires 26,280 secs (or about 7.3 hours) running on a single core of the CPU and 6,850 secs (or about 1.9 hours) utilising all 4 cores of the CPU. However, the same task takes 998 secs (or about 17 minutes) on the GeForce 8800GT, and 99 secs on the 580GTX.

In addition, the GPU-based PR-system was also tested on more advanced and powerful GPU models featuring NVidia's newest Kepler architecture. In particular, the same computationally demanding task required 59 secs running on the Tesla K20c and only 35 secs running on the GeForce GTX 780 Titan.

6.4 Discussion on the Results

The designer of a PR-system is often faced with a high dimensionality problem, as was the case of the present study, and the designer may either resort to feature reduction methods, such as the rank-features, that would probably compromise the PR-system's precision, or make use of powerful computation hardware, such as supercomputers or server clusters. Those, however, are expensive solutions and beyond the reach of most designers. The other alternative is the powerful processors of the graphics cards, commonly used in desktop computers but specially designed to deal with demanding 3-D graphics required in the games industry, which, in conjunction with parallel programming methods, can provide significant savings in processing times. Previous studies have employed similar measures to tackle processing time demanding image processing procedures, such as implementations of neural networks (Oh and Jung, 2004), Support Vector Machines (Kecman, 2001), K-Nearest Neighbour (Beliakov and Li, 2012), tomographic reconstruction algorithms (Xu and Mueller, 2007), image registration methods (Lapeer et al., 2010; Shams et al., 2010), and dose simulation (Berg et al., 2008).

Regarding the workload requirements of the present study, those were set by (a) the eighty textural features, forty from each modality in the combined US-DM dataset, (b) the number of lesions contained in both classes, (c) the features reduction-selection method, (d) the type of classifier, and (e) the evaluation method to test the accuracy of the design and the system’s precision to new, “unseen” by the PR-system, data. It is easy to assess the workload required to design the PR-system employing an optimal features selection method, such as the exhaustive search method. First, a rule of thumb, as to the maximum numbers of features that may be combined in the design of the classifier and at the same time safeguard against over-fitting, is to allow for combinations to be formed of up to one third of the size of the smallest class (30 patterns in the benign class), which in our case was 10 features-combinations (Foley, 1972). Thus, searching within 80 features to form all possible feature-combinations of up to ten features would require an enormous effort. To quantify the immensity of the problem, Figure 6.5 illustrates a comparative evaluation of the required processing times for various datasets. As it may be observed, it required more than 24 hours for the CPU to sequentially process up to 6 features combinations; in fact, processing had to be interrupted as it would have taken approximately 261 hours, or more than 10 days to be completed. The number of combinations required was 326,207,196, which may be estimated by Equation 3.31, where n is the number of features (n=80) and k is the size of feature combinations (k=1, 2,…, 6). It has to be stressed that for each features-combination, the performance of the system had to be evaluated by the leave-one-out method, which dictated that the classifier had to be redesigned as many times as was the number of data in both classes. Such processing time requirements, however, rendered the system design on a single CPU unrealisable (Karahaliou et al., 2008).

Additionally, employing a parallel PR design strategy on the CPU, by utilising all the available cores, could provide a boost in performance, which in the current case is proportional to the number of cores engaged (4-fold increase in speed). However, the handful of cores inside a typical CPU falls short against the massive parallelism provided by the hundreds of CUDA cores residing in the GPU (512 cores in 580GTX).

Therefore, employing parallel processing programming techniques on the powerful processors of the GPUs at hand, allowed for optimal system design techniques to be employed (i.e. exhaustive search and LOO) and, thus, to obtain system designs with improved classification accuracy. Explicitly, various classifiers were tested on the available data such as the Probabilistic Neural Network (PNN), the k-nearest neighbour (k-NN), the linear discriminant analysis (LDA), the non-linear Bayesian, and the more sophisticated support vector machines (SVM) with Gaussian kernel. The PNN displayed best behaviour, in terms of ease of training, classification performance and precision and it was, thus, adopted in the final design of the system.

The GPU technology adopted made also possible the fine tuning of the PNN's parameters, such as the smoothing parameter σ in the discriminant function g_j of the PNN, given by Equation 3.37, which was varied between 0.10 and 0.90, each time checking the achieved overall system accuracy, and finally fixing its value to 0.24, as it provided the highest overall system accuracy.

Another finding, which is evident from Tables 6.1-6.3 and Tables 6.5-6.7, was that classification accuracies, achieved by the optimal system design methods on the GPUs, outperformed the suboptimal design methods and that when textural features from both modalities were combined, system precision improved significantly.

Tables 6.3 and 6.4 show the highest classification accuracies achieved (93.5%) when features from both digital mammography and ultrasound were combined in the design of the classifier. Also illustrated in the box-plots of Figure 6.4, those textural features were the standard deviation, sum of squares, sum average, short run emphasis, and difference entropy that quantify specific image textural properties. The standard deviation (STD) is a measure of pixel intensity variation, signifying that on digital mammography, benign lesions were found of smoother texture than malignant ones. The range of sum of squares (SOQ^r) is a measure of anisotropy in grey-levels variance; the SOQ^r was higher in the benign lesions, implying that SOQ varied more among benign lesions. The mean of sum average (SAV^a) is an indirect measure of inhomogeneity, in terms of the existence of structures with variation in grey-levels; it attains higher values in coarser textures, thus on digital mammography malignant lesions displayed coarser texture than benign lesions. The short run emphasis (SRE^a) indicates the existence of small structures within the lesion. SRE was found lower in the malignant than in the benign lesions. The range of difference entropy ($DENTR^r$) is a measure of anisotropy in grey-levels randomness; the $DENTR^r$ was found higher in the normal lesions, indicating that DENTR differed more within benign lesions on ultrasound. Since the $DENTR^r$ was the only textural feature that displayed significant statistical differences ($p<0.005$) between the two classes (benign-malignant lesions), that feature can only be considered as sound evidence of the differences in texture between the two classes of lesions. Another important finding is that high precision pattern recognition systems may be designed by combining textural features that do not necessarily sustain statistical significant differences among the classes. This may become evident by inspection of Figure 6.3, which is a display of the QLS-mapping diagram of the members of the two classes, formed by employing the particular six-features combination in the combined DM-US dataset. As it can be observed, there is good class separation with insignificant class-overlap. These results were only plausible by the employment of the parallel processing capabilities of the GPU processors, since more textural features could be searched by examining, in realistic processing times, all possible (up to six) features combinations exhaustively. Additionally, due to speed of execution, classifier design parameters could be easily modified and tested on our data, thus assisting in the optimal system design. Our findings are comparable to those of previous studies, however tested on different databases. The highest accuracy in discriminating between benign and malignant breast lesions achieved by our PR-system was 93.5%, using the LOO evaluation method and employing both US and DM images. This accuracy corresponds to 0.92 area under the ROC curve (AUC). It should be mentioned that AUC is equal to the probability that the designed classifier ranks a randomly chosen malignant pattern higher than a randomly chosen benign pattern. Therefore, it measures the classifier's skill in ranking a set of patterns according to the degree to which they belong to the malignant or benign class (Theodoridis and Koutroumbas, 2003). Previous studies using also LOO have found: Yuan et al (Yuan et al., 2010) AUC=0.87, using digital mammography and magnetic resonance imaging, Drukker et al (Drukker et al., 2005) AUC=0.92, using mammography and sonography images, Horsch et al (Horsch et al., 2006) AUC=0.92, using mammography and sonography. In all those studies it has been shown that the employment of multimodality imaging data improves the precision of the CAD systems.

The computational requirements were enlarged by the need to assess the proposed system’s discriminatory potential to new, “unseen” by the system, breast lesions. This is actually the question often posed by the physicians when presented with a decision tool. Taking under consideration that the set of available verified data was fixed and that we would have to wait for a long period of time to attain new verified data, we employed the 10-fold external cross validation method; the available data was randomly split into two parts, 70% of the data was used to design the system and 30% to test the system’s accuracy. This procedure was repeated ten times and the mean accuracy achieved provided an assessment as the system’s precision when presented with new data in a clinical environment. Considering the dimensionality of our problem, such a procedure could only have been realised on the processors of the GPU, employing parallel processing programming techniques. One finding regarding the mean results obtained by the ECV is that the system’s overall accuracy dropped to 81% (see Table 6.5) as compared to 93.5% using the LOO evaluation method (see Table 6.4). This, however, was expected, since the optimal design method (using LOO) is an “optimistically” biased estimate (Ambroise and McLachlan, 2002), while the ECV gives a generalisation of the system’s performance. Nevertheless, the performance of the proposed system, having an estimated accuracy of 81% to new data, is reasonable enough to be considered as a second opinion diagnostic tool in a clinical environment.

6.5 System Limitations

Regarding its limitations, the proposed GPU-based PR-system has only been trained for binary classification problems, whereas an unknown pattern is assigned to one of two available classes. Specifically, the system can characterise a breast lesion either as benign or malignant, but it cannot provide any further information as to the exact type of the lesion, for example whether it is a case of Infiltrating Ductal Carcinoma or Infiltrating Lobular Carcinoma. Another limitation of the proposed system is the fact that it does not provide automatic segmentation of the breast lesions. In particular, for the purposes of this research, segmentation was performed manually by an experienced physician. Accordingly, the system requires segmented breast images with delineated ROIs as input.

Furthermore, an additional limitation is closely related to the specific type and model of the diagnostic imaging devices employed, as well as to the challenge of adapting the proposed CAD system to different models that feature diverse specifications and calibration that ultimately affect the quality of the produced images. Simply put, there is absolutely no guarantee that the proposed CAD system will continue to operate at the same level of accuracy, in case images from a different model of digital mammography device are used as input. The same applies in case images came from a differently calibrated device.

To address this challenge, a two-step approach could be adopted. Firstly, an initial assessment of the produced image quality in terms of contrast resolution, spatial resolution, contrast-to-noise ratio, etc. could be performed, employing specialised phantoms, before incorporating any new diagnostic imaging device. Thus, devices falling below a certain threshold should be excluded, as they fail to produce the minimum image quality required.

Secondly, a complete dataset of images, particular to each diagnostic imaging device, could be created, so that the proposed CAD system can be trained and adapted to the device's specific characteristics. This computationally demanding process could be easily achieved employing the proposed GPU-based CAD system.

6.6 Summary

In this chapter, results were presented showing classification accuracies for discriminating malignant from benign lesions to be 85.5% using US-features alone, 82.3% employing DM-features alone, and 93.5% combining US and DM features. Mean accuracy to new "unseen" data for the combined US and DM features was 81%. Those classification accuracies were about 10% higher than accuracies achieved on a single CPU, using sequential programming methods, and 150-fold faster. Findings regarding textural content that discriminates benign from malignant lesions indicate that benign lesions were smoother, more homogeneous, containing larger structures.

Chapter 7

System adaptation to other Medical Imaging applications

7.1 Overview

The proposed GPU-based CAD system was employed to tackle other Medical Imaging PR problems. This chapter provides a brief discussion of these applications that included classification of rare brain tumours, discrimination of patients with micro-ischemic and multiple sclerosis lesions, classification of normal and pathological knee cartilages, and discrimination between low and high grade laryngeal cancer cases.

7.2 Real time Decision Support System for diagnosis of rare cancers

Even though histological diagnosis is fundamentally important for patient management, the potential of diagnostic errors still remains substantially high, ranging from 25% to 40%, in routine conditions. Factors affecting diagnostic accuracy include experts' subjectivity and lack of experience, intra and inter observer variability and poor tumour sampling; rare cancers low prevalence and their biological complexity hinder the establishment of concrete criteria able to predict tumours' behaviour, and, thus, to administrate proper treatments. The proposed DSS is built on a GPU framework and it is capable of updating its structure in real time whenever a new verified case is uploaded on its repository. The system adjusts its design based on the exact clinical question to be answered, which is user-defined from on a list of potential outcomes according to existing clinical classifications.

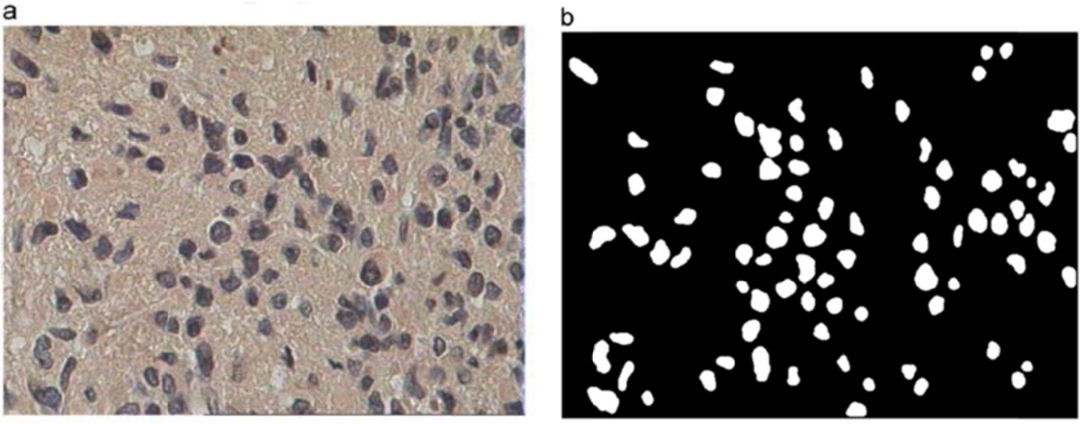


Figure 7.1: (a): An example of a high grade pleomorphic glioblastoma multiforme tumour, (b): Segmentation of Figure1a for isolation of nuclei from surrounding tissue

Clinical material comprised 140 cases of rare brain tumours, including astrocytomas (variants: gemistocytic, fibrillary and mixed), anaplastic astrocytomas, glioblastomas, oligodendroglioma and mixed gliomas. At least three Hematoxylin-Eosin (H&E) stained sections were generated from the same block of formalin-fixed paraffin-embedded tissue for each case (patient). Of the 140 cases, 61 were characterised as grade II (aggressive), 35 as grade III (more aggressive) and 44 as grade IV (most aggressive). The proposed DSS was designed employing the PNN classifier, the exhaustive search and the leave-one-out methods, Nvidia GeForce GTX580 and the CUDA parallel programming platform. Five microscopy images from each patient were digitised, using a light microscopy imaging system consisted of a Zeiss Axiostar-Plus microscope (ZEISS; Germany) connected to a Leica DC 300 F colour video camera (LEICA; Germany). A segmentation algorithm was developed to isolate nuclei from surrounding tissue (see Figure 1a and 1b). Segmented nuclei were quantified by means of 20 textural and morphological features describing the size, shape and grain texture of nuclei.

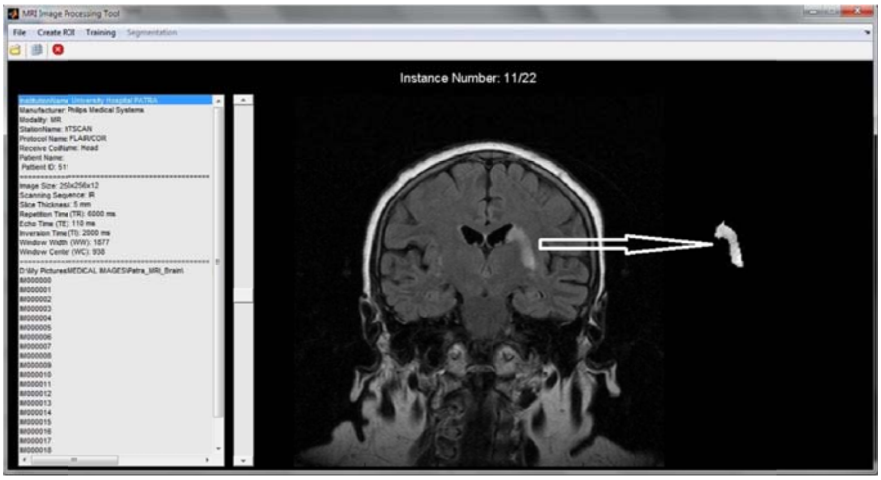
System design was optimised using a combination of four (4) textural and morphological features with 78.6% overall accuracy, whereas system generalisation to new unseen by the system data was 73.8% ± 3.2%. The proposed GPU-based DSS achieved significantly higher training speed, outperforming the CPU-based system (Intel Core 2 Quad CPU at 2.83GHz) by a factor that ranged from 267 to 288 times.

Thus, by exploiting the inherently parallel architecture of a consumer level GPU, the proposed approach enabled real time, optimal design of a DSS for any user-defined clinical question for improving diagnostic assessments, prognostic relevance and concordance rates for rare cancers in clinical practice. A detailed account can be found at (Sidiropoulos et al., 2012).

7.3 Discriminating between patients with micro-ischaemic and multiple sclerosis lesions, using MRI images

The aim was to employ state-of-art GPU technology and CUDA parallel programming to design and implement a stand-alone pattern recognition system (PR-system) to discriminate between patients with micro-ischemic (mIS) and multiple sclerosis (MS) lesions.

The dataset comprised MRI image series of 32 patients with mIS and 19 with MS lesions. The outline of each MS and mIS lesion on the MRI images was drawn by means of a custom made tool developed in Matlab (see Figure 7.2). The outlined lesion-images were used to extract textural features to be used in the design of the PR-system. The PNN classifier and forty textural features, calculated from lesions in the MRI images, were used to design the PR-system. The design applied parallel processing programming on the processors of an Nvidia GeForce 8800GT (see Figure 7.3). The involvement of GPU technology and parallel processing methods was necessary in order to deal with the high dimensionality of the problem at hand and to obtain an optimal design of the proposed system, which is capable of being redesigned on site when additional verified data are added to its depository.



```
InstitutionName: University Hospital PATRA
Manufacturer: Philips Medical Systems
Modality: MR
StationName: ITSCAN
Protocol Name: FLAIR/COR
Receive CoilName: Head
Patient Name:
 Patient ID: 51
************************************************
Image Size: 256x256x12
Scanning Sequence: IR
Slice Thickness: 5 mm
Repetition Time (TR): 6000 ms
Echo Time (TE): 110 ms
Inversion Time(TI): 2000 ms
Window Width (WW): 1877
Window Center (WC): 938
************************************************
D:\My Pictures\MEDICAL IMAGES\Patra_MRI_Brain\
IM000000
IM000001
IM000002
IM000003
IM000004
IM000005
IM000006
IM000007
IM000008
IM000009
IM000010
IM000011
IM000012
IM000013
IM000014
IM000015
IM000016
IM000017
IM000018
```

Figure 7.2:Custom-made software tool for extracting lesions

Highest classification accuracy was 90.2%, employing six textural features. It took about 135 minutes to design the PR-System on a desktop CPU featuring an Intel Core 2 Quad Q9550, against 250 seconds on the Nvidia 8800GT GPU card. By examining the textural features used to design the PR-system, it was found that MS-lesions appeared smoother than the mIS lesions but with greater variation in the textural composition from lesion-to-lesion. The latter, may be attributed to processes such as the degree of demyelination and inflammation in MS that affects the MR signal.

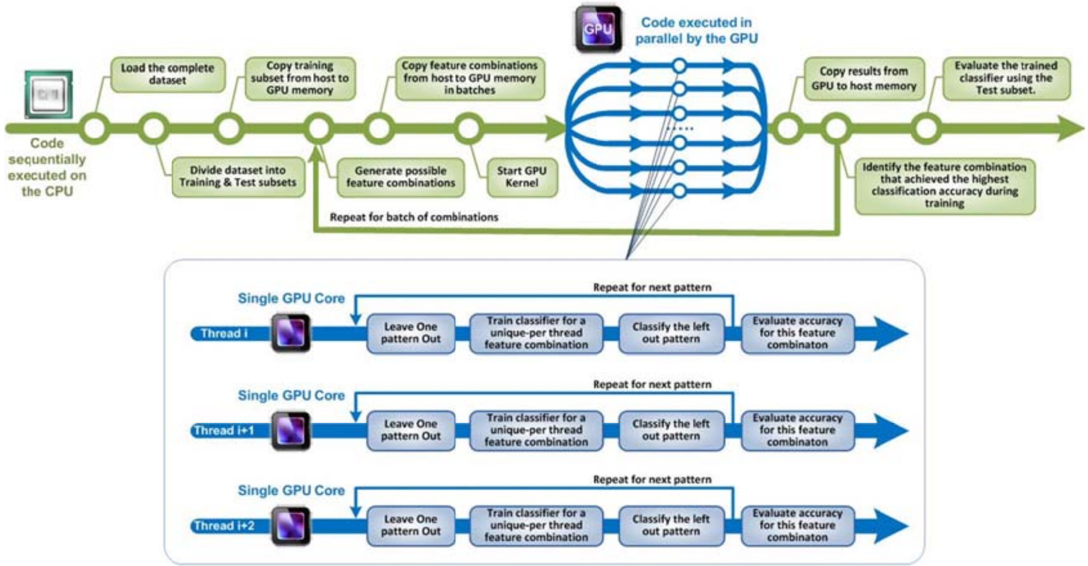

Code executed in parallel by the GPU
GPU
Load the complete dataset
Copy training subset from host to GPU memory
Copy feature combinations from host to GPU memory in batches
Copy results from GPU to host memory
Evaluate the trained classifier using the Test subset.
Code sequentially executed on the CPU
Divide dataset into Training & Test subsets
Generate possible feature combinations
Start GPU Kernel
Identify the feature combination that achieved the highest classification accuracy during training
Repeat for batch of combinations
Single GPU Core
Thread i
Thread i+1
Thread i+2
Repeat for next pattern
Leave One pattern Out
Train classifier for a unique-per thread feature combination
Classify the left out pattern
Evaluate accuracy for this feature combinaton

Figure 7.3: Flow diagram of parallel processing performed on the graphics processing unit

In conclusion, a pattern recognition system has been designed to discriminate with reasonable accuracy MS from mIS cases based on the PNN classifier and on textural features from ROIs of brain MRI images. Our findings revealed differences in imaged textural structures between MS and mIS lesions. Successful system design could only be achieved by exploiting the high processing speed and parallel processing capabilities of a commercial GPU. The proposed system can be employed in a clinical environment, as a second opinion tool with 80% confidence to new data, and its performance may be continuously improved by redesigning it, on site, with additional verified data. A detailed account can be found at (Solomou et al., 2012).

7.4 Discriminating between injured normal and pathological knee cartilage

The aim was the discrimination between normal and pathological knee articular cartilage of the medial femoral (MFC) and tibial condyles (MTC), often affected by knee injuries in young adults participating in sports activities. Knee injuries of the articular cartilage in young active adults have been associated with early cartilage changes, which may be undetected, yet, they eventually may lead to irreversible cartilage lesions and osteoarthritis. This is because articular cartilage has a very limited capacity to repair and, thus, articular injuries result in premature joint degeneration. Because chondral injuries may be asymptomatic, accurate diagnosis is important in patient management.

The dataset comprised segmented ROIs from coronal and sagittal 3T MRI images of the MFC and MTC cartilages (see Figure 7.4), of young patients, 28 with abnormality-free knee and 16 with pathological findings. The PR-system was designed employing the PNN classifier, forty (40) textural features from the segmented MFC and MTC ROIs, the exhaustive search and leave-one-out evaluation method, while the system's precision to new data was assessed by employing the ECV method. Optimal system design was accomplished on an Nvidia GeForce GTX 580 using CUDA.

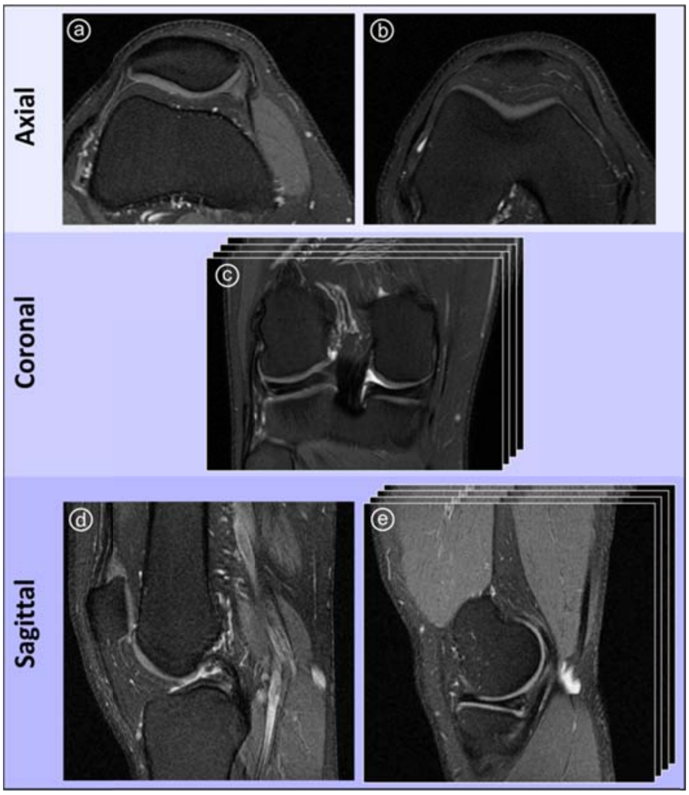


Figure 7.4: Snapshots of the three PD-FSE FS MRI planes. In these planes and in their corresponding slices, the following ROIs were extracted: from the axial planes: (a) patella and (b) femoral trochlea, from the coronal planes; (c) lateral and medial of femoral and tibial condyles, from the sagittal planes: (d) patella and femoral trochlea, and (e) lateral and medial of the MFC and MTC cartilages

PR-system design on the GPU required about 3.5 minutes against 15 hours on CPU-based system (Intel Core 2 Quad Q9550). Highest classification accuracies in discriminating between normal and pathological knee articular cartilage for the MFC and MTC were 93.2% and 95.5% and accuracies to “unseen” data were 89% and 86% respectively. Regarding findings from textural features used in the optimal design of the PR-system, the cartilage texture of the pathological knee appeared more homogeneous and comprising larger structures. A detailed account can be found at (Kostopoulos et al., 2013).

7.5 Differentiating between low and high grade laryngeal cancer cases

The aim was discrimination between low and high grade laryngeal cancer cases, employing immunohistochemically stained, for p63 expression, histopathology images. Laryngeal cancer has bad prognosis and diagnosis is an important step towards following the right treatment for best prognosis, the aim being to improve patient survival. However, assessing the severity of laryngeal lesions is subject to the pathologist’s experience and it has low intra- and inter-observer reproducibility rates. To assist in the assessment of the severity of laryngeal cancer, a PR system was developed to automatically characterise microscopy images of laryngeal cancer lesions as of low grade (high differentiation) or high grade (moderate and low differentiation), based on textural, morphological, and topological nuclei features.

The material comprised fifty-five verified cases of laryngeal cancer, 21 of low grade and 34 of high grade malignancy. Histopathology images were first processed for automatically segmenting p63 expressed nuclei. 52 features were next extracted from the segmented nuclei within image ROIs designated by the expert physician and concerning nuclei texture, shape, and physical topology in the image. Those features, the PNN classifier, the exhaustive search and LOO methods, and the ECV method, for assessing the PR-system’s precision to “unseen” data, were employed to design the

PR-system on the multiprocessors of the Nvidia 580GTX GPU card, using the CUDA parallel programming model and C++ programming language.

PR-system performance in classifying laryngeal cancer cases as low grade and high grade was 85.7% and 94.1% respectively. The system's overall accuracy was 90.9%, using seven features, and its estimated accuracy to "unseen" by the system cases was 80%. It required over 24 hours to reach the best design, using the four cores of the Intel Core 2 Quad Q9550 and parallel programming under the OpenMP platform, and forty-four (44) minutes, using the GPU and the CUDA programming model. Nuclei features used in the optimal PR-system design were related to texture homogeneity, contrast, contents and the dispersion of nuclei in the selected by the physician ROIs. High grade nuclei texture was found inhomogeneous, of higher local contrast, containing smaller structures, and the nuclei topology of higher in-between distance.

Optimum system design was only feasible after employing parallel processing techniques and GPU technology. The proposed system was structured so as to function in a clinical environment, as a research tool, and with the capability of being redesigned on site when new verified cases are added to its repository. A detailed account can be found at (Ninos et al., 2013).

7.6 Summary

The PR-system design was adapted for tackling other Medical Imaging problems, as a proof of its generalisation. These included classification of rare brain tumours, discrimination of patients with micro-ischemic and multiple sclerosis lesions, classification of normal and pathological knee cartilages, and separation of low from high grade laryngeal cancer cases.

Regarding PR-designs for other applications, classification results achieved for rare brain tumours were 78.6% for overall accuracy and 73.8% for estimated generalisation accuracy, accelerating system design 267 times. Similarly, in discriminating between micro-ischaemic and multiple sclerosis lesions on MRI, accuracies achieved were 90.2% and 80% with 32-fold design acceleration, in characterising injured normal and pathological knee cartilages on MRI maximum accuracies achieved were 93.2% and 89% with 257-fold design acceleration, and in separating low from high grade laryngeal cancer lesions on microscopy imaging, maximum accuracies achieved were 93.2% and 89%, with 130-fold design acceleration.

Chapter 8

Conclusions – Future Work

8.1 Overview

This chapter summarises the findings of the present thesis, draws conclusions and indicates directions for future research.

8.2 Conclusions

The application of GPU technology and parallel processing programming techniques in the design of CAD systems, presented in this thesis, has made possible to apply optimum, but considerably laborious feature selection techniques, to design high precision classification systems that, otherwise, would be impossible to implement. In particular, the employment of GPU technology not only rendered realisable the design of CAD systems based on imaging data from different modalities, whereas, the dimensionality of the problem increases due to the number of modalities involved, but also considerably accelerated the design of such CAD systems.

Additionally, since retraining of the classifier was achieved due to significant reductions in processing times, it was rendered possible to estimate the actual precision of such designed CAD system to new, “unseen” by the system data, employing time consuming external cross validation evaluation methods. Thus, trustworthy, high precision, CAD systems can be designed by limited amount of data. These CAD systems also feature the ability to be redesigned on site, once more verified data become available.

Furthermore, it has been made possible to prove that such laborious problems can be tackled by relatively low-cost GPUs, which are readily available to consumers, in contrast to previously proposed alternatives such as server clusters or supercomputers, accessible only to few.

In conclusion, by applying the methods developed in the present thesis to other medical applications, it was made possible to increase the accuracy of the respective CAD systems, while new clinical knowledge was contributed as to imaging properties of the human tissue under examination. In particular:

- In brain lesions, employing MR-imaging, MS lesions were found smoother in texture but with greater variation in textural composition, which is due to different degrees of demyelination and inflammation in MS that affects the MR signal.

- In breast lesions, benign lesions were found smoother, more homogeneous, and containing larger structures.
- In knee cartilage texture analysis on MRI imaging, pathological ROIs appeared more homogeneous, of larger structures aligned along preferred directions, and of lower textural anisotropy.
- In laryngeal cancer histopathology images it was found that grade I nucleus texture was more homogeneous, better stained, of structured nature, and of ordinary shape when compared to high grade nuclei.

8.3 Future Work

Regarding future work, incorporation of state-of-art GPU-programming capabilities, such as dynamic parallelisation offered by Kepler and Maxwell class GPUs, could be employed for further optimising design flexibility, reducing design time, and increasing precision.

In addition, expanding CAD system design to incorporate information from more imaging modalities, such as MRI, could be further investigated, while the proposed GPU-based PR system could be tested against emerging classification methods, using benchmark datasets available on the web, in order to gain further insight of its advantages and limitations.

Furthermore, research on incorporating multi-classifier ensemble schemes could be considered, especially since the increased computational power of modern GPUs allows for it. Additionally, implementation of PR systems on heterogeneous platforms exploiting multicore CPUs along with GPUs could be examined by employing the OpenCL language, instead of CUDA. Accordingly, the system’s scalability on GPU clusters could be investigated.

Finally, following the Software as a Service (SaaS) paradigm, web services technology could be employed to provide other researchers a web-based facility for designing CAD systems for various applications.

References

Ahmed, N. and K. R. Rao (1975). Orthogonal Transforms for Digital Signal Processing. Berlin - Heidelberg - New York, Springer-Verlag.

Ambroise, C. and G. J. McLachlan (2002). "Selection bias in gene extraction on the basis of microarray gene-expression data." *Proc Natl Acad Sci U S A* 99(10): 6562-6566.

AMD. (2005). "AMD Athlon™ 64 X2 Dual Core Product Data Sheet". Available at: http://support.amd.com/TechDocs/33425.pdf#search=Athlon%2064%20X2, Last accessed: 25/04/2014.

Beam, C. A., P. M. Layde, et al. (1996). "Variability in the interpretation of screening mammograms by US radiologists. Findings from a national sample." *Arch Intern Med* 156(2): 209-213.

Beliakov, G. and G. Li (2012). "Improving the speed and stability of the k-nearest neighbors method." *Pattern Recognition Letters* 33(10): 1296-1301.

Berg, W. A., J. D. Blume, et al. (2008). "Combined screening with ultrasound and mammography vs mammography alone in women at elevated risk of breast cancer." *JAMA* 299(18): 2151-2163.

Bird, R. E., T. W. Wallace, et al. (1992). "Analysis of cancers missed at screening mammography." *Radiology* 184(3): 613-617.

Birdwell, R. L., P. Bandodkar, et al. (2005). "Computer-aided detection with screening mammography in a university hospital setting." *Radiology* 236(2): 451-457.

Bocchi, L., G. Coppini, et al. (1997). "Tissue characterization from X-ray images." *Medical Engineering & Physics* 19(4): 336-342.

Buck, I., T. Foley, et al. (2003). BrookGPU Project, Stanford University Graphics Lab, http://graphics. stanford. edu/projects/brookgpu/7. Microsoft. The Viewing Frustum. DirectX Documents for C++, DirectX SDK.

Burges, C. J. C. (1998). "A tutorial on Support Vector Machines for pattern recognition." *Data Mining and Knowledge Discovery* 2(2): 121-167.

Burges, C. J. C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition." *Data Min. Knowl. Discov.* 2(2): 121-167.

Chapman, B., G. Jost, et al. (2008). Using OpenMP: portable shared memory parallel programming, MIT press.

Che, S., M. Boyer, et al. (2008). "A performance study of general-purpose applications on graphics processors using CUDA." *Journal of parallel and distributed computing* 68(10): 1370-1380.

Clamp, A., S. Danson, et al. (2003). "Hormonal and genetic risk factors for breast cancer." *The Surgeon* 1(1): 23-31.

Crow, T. S. (2004). Evolution of the graphical processing unit, Citeseer.

Cupples, T. E., J. E. Cunningham, et al. (2005). "Impact of computer-aided detection in a regional screening mammography program." *AJR Am J Roentgenol* 185(4): 944-950.

Dai, Y., J. Tian, et al. (2010). "Real-time visualized freehand 3D ultrasound reconstruction based on GPU." *IEEE Trans Inf Technol Biomed* 14(6): 1338-1345.

Dean, J. C. and C. C. Ilvento (2006). "Improved cancer detection using computer-aided detection with diagnostic and screening mammography: prospective study of 104 cancers." *AJR Am J Roentgenol* 187(1): 20-28.

Devyver, P. A. and J. Kittler (1982). Pattern Recognition: A Statistical Approach, Prentice-Hall.

Di Blas, A. and T. Kaldewey (2009). "Data monster." *Spectrum, IEEE* 46(9): 46-51.

Directive, E. (1995). "95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data." *Official Journal of the EC* 23: 6.

Doi, K. (2007). "Computer-aided diagnosis in medical imaging: historical review, current status and future potential." *Comput Med Imaging Graph* 31(4-5): 198-211.

Drukker, K., K. Horsch, et al. (2005). "Multimodality computerized diagnosis of breast lesions using mammography and sonography." *Acad Radiol* 12(8): 970-979.

Duda, R. O., P. E. Hart, et al. (2012). Pattern Classification, Wiley.

Evers, K. (2001). "Diagnostic Breast Imaging." *American Journal of Roentgenology* 177(5): 1094.

Fang, J., A. L. Varbanescu, et al. (2011). A comprehensive performance comparison of CUDA and OpenCL. Parallel Processing (ICPP), 2011 International Conference on, IEEE.

Farber, R. (2011). CUDA Application Design and Development, Morgan Kaufmann.

Foley, D. (1972). "Considerations of sample and feature size." *Information Theory, IEEE Transactions on* 18(5): 618-626.

Foley, D. H. (1972). "Considerations of sample and feature size." *Information Theory, IEEE Transactions on* 18(5): 618-626.

Freer, T. W. and M. J. Ulissey (2001). "Screening mammography with computer-aided detection: prospective study of 12,860 patients in a community breast center." *Radiology* 220(3): 781-786.

Galloway, M. M. (1975). "Texture analysis using gray-level run lengths." *Computer Graphics and Image Processing* 4: 172-179.

GLOBOCAN. (2008). "Cancer Incidence and Mortality Worldwide". Available at: http://www.iarc.fr/en/media-centre/iarcnews/2010/globocan2008.php, Last accessed: 27/0/2014.

Gokhale, S. (2009). Ultrasound characterization of breast masses.

Gonzalez, R. and R. Woods (2002). Digital image processing. NY, Addison-Wesley Pub

Gose, E., R. Johnsonbaugh, et al. (1996). Pattern Recognition and Image Analysis. New Jersey, Prentice Hall PTR.

Grunberg, W. I. and O. H. Domingo (2011). "Parathyroid cyst." *Am Surg* 77(11): E256-257.

Gunal, S., O. N. Gerek, et al. (2009). "The search for optimal feature set in power quality event classification." *Expert Systems with Applications* 36(7): 10266-10273.

Gur, D., J. H. Sumkin, et al. (2004). "Changes in breast cancer detection and mammography recall rates after the introduction of a computer-aided detection system." *J Natl Cancer Inst* 96(3): 185-190.

Haralick, R. M., K. Shanmugam, et al. (1973). "Textural Features for Image Classification." *IEEE Trans Sys Man Cyb* 3: 610-621.

Haralick, R. M., K. Shanmugam, et al. (1973). "Textural Features for Image Classification." *IEEE Transactions on Systems Man and Cybernetics* 3(6): 610-621.

Harms, S. E. (1999). "Technical report of the international working group on breast MRI." *J Magn Reson Imaging* 10(6): 979.

Hayes, D. F. (2007). "Clinical practice. Follow-up of patients with early breast cancer." *N Engl J Med* 356(24): 2505-2513.

Hollander, M. and D. Wolfe (1999). Nonparametric Statistical Methods, 2nd Edition, Wiley-Interscience.

Horsch, K., M. L. Giger, et al. (2006). "Classification of breast lesions with multimodality computer-aided diagnosis: observer study results on an independent clinical data set." *Radiology* 240(2): 357-368.

Houssami, N. and S. Ciatto (2011). "The evolving role of new imaging methods in breast screening." *Prev Med* 53(3): 123-126.

Humphrey, L. L., M. Helfand, et al. (2002). "Breast cancer screening: a summary of the evidence for the U.S. Preventive Services Task Force." *Ann Intern Med* 137(5 Part 1): 347-360.

Intel. (2005). "Intel Pentium D Processor ". Available at: http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/pentium-d-processor-brief.pdf, Last accessed: 25/04/2014.

Jemal, A., R. Siegel, et al. (2008). "Cancer statistics, 2008." *CA Cancer J Clin* 58(2): 71-96.

Kakimoto, T., K. Dohi, et al. (2012). "Performance comparison of GPU programming frameworks with the striped Smith-Waterman algorithm." *ACM SIGARCH Computer Architecture News* 40(5): 70-75.

Karahaliou, A. N., I. S. Boniatis, et al. (2008). "Breast cancer diagnosis: analyzing texture of tissue surrounding microcalcifications." *IEEE Trans Inf Technol Biomed* 12(6): 731-738.

Kecman, V. (2001). Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, Mit Press.

Khronos (2008). "The opencl specification." *A. Munshi, Ed*.

Kirk, D. B. and W. H. Wen-mei (2012). Programming massively parallel processors: a hands-on approach, Newnes.

Koepf, S. (2009). "k-combinations without repetition in lexicographic order". Available at: http://www.aconnect.de/friends/editions/computer/combinatoricode_e.html, Last accessed: 25/04/2014.

Kossoff, M. B. (2000). "Ultrasound of the breast." *World J Surg* 24(2): 143-157.

Kostopoulos, S., K. Sidiropoulos, et al. (2013). "Pattern-recognition system, designed on GPU, for discriminating between injured normal and pathological knee cartilage." *Magnetic resonance imaging* 31(5): 761-770.

Kumar, V., A. Grama, et al. (1994). Introduction to parallel computing, Benjamin/Cummings Redwood City.

Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. Hoboken, New Jersey, John Wiley & Sons, Inc.

Laine, H. R., T. Tukeva, et al. (1996). "Assessment of mammography and ultrasound examination in the diagnosis of breast cancer." *European Journal of Ultrasound* 3(1): 9-14.

Lapeer, R. J., S. K. Shah, et al. (2010). "An optimised radial basis function algorithm for fast non-rigid registration of medical images." *Comput Biol Med* 40(1): 1-7.

Lee, J. M., E. F. Halpern, et al. (2009). "Evaluating the correlation between film mammography and MRI for screening women with increased breast cancer risk." *Acad Radiol* 16(11): 1323-1328.

Luebke, D. and G. Humphreys (2007). How GPUs Work. Computer. 40**:** 96-100.

Mainiero, M. B., A. Goldkamp, et al. (2005). "Characterization of Breast Masses With Sonography: Can Biopsy of Some Solid Masses Be Deferred?" *Journal of Ultrasound in Medicine* 24(2): 161-167.

Majid, A. S., E. S. de Paredes, et al. (2003). "Missed breast carcinoma: pitfalls and pearls." *Radiographics* 23(4): 881-895.

Marieb, E. N. and K. Hoehn (2013). Human Anatomy & Physiology, Benjamin Cummings ; Pearson.

Martini, F. H., J. L. Nath, et al. (2011). Fundamentals of Anatomy & Physiology with Masteringa&p, and Laboratory Investigations in Anatomy & Physiology, Main Version, A&p Applications Manual (Valuepack Version), Practice Anatomy Lab 3.0 (for Packages with Masteringa&p Access Code) Package, Benjamin-Cummings Publishing Company.

Mattson, T. G., B. A. Sanders, et al. (2004). Patterns for parallel programming, Pearson Education.

Mavroforakis, M., H. Georgiou, et al. (2005). "Significance analysis of qualitative mammographic features, using linear classifiers, neural networks and support vector machines." *European Journal of Radiology* 54(1): 80-89.

McCool, M. D., Z. Qin, et al. (2002). Shader metaprogramming. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, Eurographics Association.

Microsoft. (2006). "Microsoft Research Accelerator Project: http://research.microsoft.com/en-us/downloads/648909e1-cb85-46c4-9a94-3cca55971b1d/". Available at: Last accessed.

Moore, G. E. (2006). "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff." *Solid-State Circuits Society Newsletter, IEEE* 11(5): 33-35.

Moore, S. K. (2010). "Multicore CPUs: Processor Proliferation". Available at: http://spectrum.ieee.org/semiconductors/processors/multicore-cpus-processor-proliferation, Last accessed: 25/04/2014.

Morton, M. J., D. H. Whaley, et al. (2006). "Screening mammograms: interpretation with computer-aided detection--prospective evaluation." *Radiology* 239(2): 375-383.

MPI. (2009). "MPI: A Message-Passing Interface Standard, Version 2.2". Available at: http://www.mpiforum.org/docs/mpi-2.2/mpi22-report.pdf, Last accessed: 25/04/2014.

Ninos, K., S. Kostopoulos, et al. (2013). "Computer-based image analysis system designed to differentiate between low-grade and high-grade laryngeal cancer cases." *Analytical and quantitative cytology and histology/the International Academy of Cytology [and] American Society of Cytology* 35(5): 261-272.

Nvidia. (2014). "CUDA C Programming Guide". Available at: http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html, Last accessed: 24/04/2014.

Oh, K.-S. and K. Jung (2004). "GPU implementation of neural networks." *Pattern Recognition* 37(6): 1311-1314.

Ohmer, J., F. Maire, et al. (2005). Implementation of Kernel Methods on the GPU. Digital Image Computing: Techniques and Applications, 2005. DICTA '05. Proceedings 2005.

OpenMP. (2005). "OpenMP Application Program Interface". Available at: http://www.openmp.org/mp-documents/spec25.pdf, Last accessed: 25/04/2014.

Pallipuram, V. K., M. Bhuiyan, et al. (2012). "A comparative study of GPU programming models and architectures using neural networks." *The Journal of Supercomputing* 61(3): 673-718.

Pang, W. M., J. Qin, et al. (2011). "Accelerating simultaneous algebraic reconstruction technique with motion compensation using CUDA-enabled GPU." *Int J Comput Assist Radiol Surg* 6(2): 187-199.

Park, I. K., N. Singhal, et al. (2011). "Design and performance evaluation of image processing algorithms on GPUs." *Parallel and Distributed Systems, IEEE Transactions on* 22(1): 91-104.

Patterson, D. (2010). "The trouble with multi-core." *Spectrum, IEEE* 47(7): 28-32, 53.

Patterson, D. (2010). "The Trouble With Multicore". Available at: http://spectrum.ieee.org/computing/software/the-trouble-with-multicore, Last accessed: 25/04/2014.

Pratt, W. K. (2007). Digital Image Processing: PIKS Scientific Inside, Wiley.

Pudil, P., J. Novovičová, et al. (1994). "Floating search methods in feature selection." *Pattern Recognition Letters* 15(11): 1119-1125.

Ruiz, A., O. Sertel, et al. (2009). "Stroma classification for neuroblastoma on graphics processors." *Int J Data Min Bioinform* 3(3): 280-298.

Saeys, Y., I. Inza, et al. (2007). "A review of feature selection techniques in bioinformatics." *Bioinformatics* 23(19): 2507-2517.

Sahiner, B., H. P. Chan, et al. (2009). "Multi-modality CADx: ROC study of the effect on radiologists' accuracy in characterizing breast masses on mammograms and 3D ultrasound images." *Acad Radiol* 16(7): 810-818.

Santhanam, A. P., Y. Min, et al. (2012). "A multi-GPU real-time dose simulation software framework for lung radiotherapy." *Int J Comput Assist Radiol Surg* 7(5): 705-719.

Sardanelli, F., F. Podo, et al. (2007). "Multicenter comparative multimodality surveillance of women at genetic-familial high risk for breast cancer (HIBCRIT study): interim results." *Radiology* 242(3): 698-715.

Schenk, O., M. Christen, et al. (2008). "Algorithmic performance studies on graphics processing units." *Journal of parallel and distributed computing* 68(10): 1360-1369.

SEER. (2011). "SEER Stat Fact Sheets: Breast Cancer". Available at: http://seer.cancer.gov/statfacts/html/breast.html, Last accessed: 26/04/2014.

Sen, R. (2012). "Developing parallel programs." *Advances in Computer Science: an International Journal* 1(1): 18-27.

Shams, R., P. Sadeghi, et al. (2010). "Parallel computation of mutual information on the GPU with application to real-time registration of 3D medical images." *Comput Methods Programs Biomed* 99(2): 133-146.

Sidiropoulos, K., D. Glotsos, et al. (2012). "Real time decision support system for diagnosis of rare cancers, trained in parallel, on a graphics processing unit." *Computers in biology and medicine* 42(4): 376-386.

Sidiropoulos, K. P., S. A. Kostopoulos, et al. (2013). "Multimodality GPU-based computer-assisted diagnosis of breast cancer using ultrasound and digital mammography images." *International journal of computer assisted radiology and surgery* 8(4): 547-560.

Solomou, E., S. Kostopoulos, et al. (2012). "Designing a pattern recognition system on GPU for discriminating between patients with micro-ischaemic and multiple sclerosis lesions, using MRI images." *International Journal of High Performance Computing Applications*: 1094342012464253.

Specht, D. F. (1990). "Probabilistic Neural Networks." *Neural Networks* 3: 109-118.

Stavros, A. T., C. L. Rapp, et al. (2004). Breast Ultrasound, Lippincott Williams & Wilkins.

Stavros, A. T., D. Thickman, et al. (1995). "Solid breast nodules: use of sonography to distinguish between benign and malignant lesions." *Radiology* 196(1): 123-134.

Sutter, H. and J. Larus (2005). "Software and the Concurrency Revolution." *Queue* 3(7): 54-62.

Tabar, L. and P. B. Dean (2008). "Thirty years of experience with mammography screening: A new approach to the diagnosis and treatment of breast cancer." *Breast Cancer Research* 10(SUPPL. 4).

Tang, J., R. M. Rangayyan, et al. (2009). "Computer-aided detection and diagnosis of breast cancer with mammography: recent advances." *IEEE Trans Inf Technol Biomed* 13(2): 236-251.

Theodoridis, S. and K. Koutroumbas (2003). Pattern Recognition. San Diego, Elsevier.

Theodoridis, S., A. Pikrakis, et al. (2010). Introduction to Pattern Recognition: A Matlab Approach: A Matlab Approach, Elsevier Science.

Weber, R., A. Gothandaraman, et al. (2011). "Comparing hardware accelerators in scientific applications: A case study." *Parallel and Distributed Systems, IEEE Transactions on* 22(1): 58-68.

WHO (2013). Global Health Estimates.

Wilt, N. (2013). The cuda handbook: A comprehensive guide to gpu programming, Pearson Education.

Wolfe, M. (2010). "Understanding the CUDA Data Parallel Threading Model, A Primer ". Available at: https://www.pgroup.com/lit/articles/insider/v2n1a5.htm, Last accessed: 25/04/2014.

Xu, F. and K. Mueller (2007). "Real-time 3D computed tomographic reconstruction using commodity graphics hardware." *Phys Med Biol* 52(12): 3405-3419.

Yuan, Y., M. L. Giger, et al. (2010). "Multimodality computer-aided breast cancer diagnosis with FFDM and DCE-MRI." *Acad Radiol* 17(9): 1158-1167.

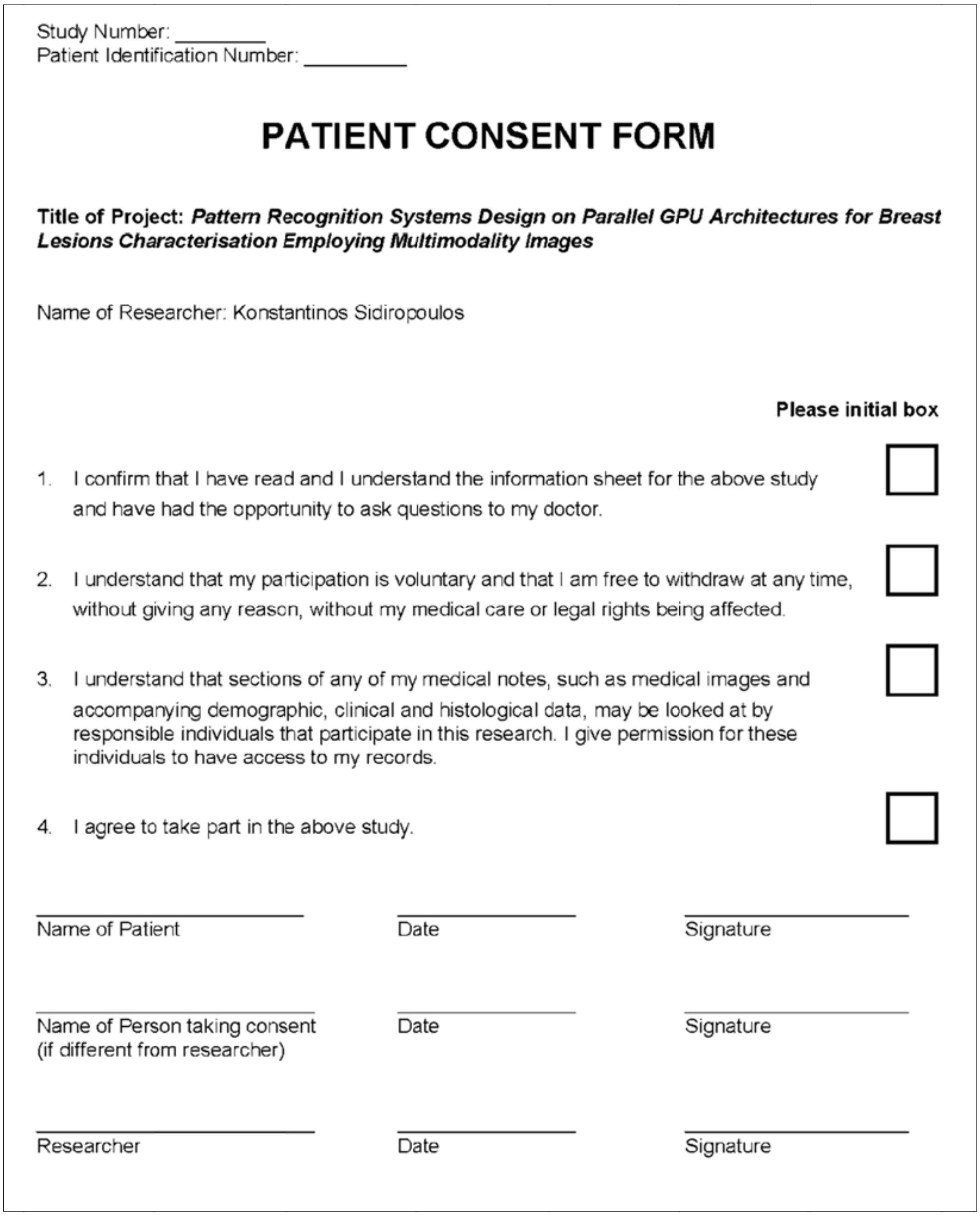
Appendix A

XML structure used to store patient data and delineated ROIs

```
<?xml version="1.0" encoding="utf-8"?>
<DicomDir>
  <Patients>
  <Patient>
    <PatientName>Jane Smith</PatientName>
    <PatientID>10.04.2007</PatientID>
    <PatientBirthDate>Not Available</PatientBirthDate>
    <PatientSex>F</PatientSex>
    <Studies>
      <Study>
        <StudyInstanceUID>1.2.840.113619.2.66.2218493336.4356070410091952.20000</StudyInstanceUID>
        <StudyDate>10.04.2007</StudyDate>
        <StudyTime>09:19:52</StudyTime>
        <StudyDescription>Not Available</StudyDescription>
        <StudyID>354</StudyID>
        <Serieses>
          <Series>
            <SeriesInstanceUID>1.2.840.113619.2.66.2218493336.4356070410092200.10003</SeriesInstanceUID>
            <SeriesNumber>1037</SeriesNumber>
            <SeriesModality>MG</SeriesModality>
            <SeriesDescription>Not Available</SeriesDescription>
            <SeriesDate>10.04.2007</SeriesDate>
            <SeriesTime>09:22:06</SeriesTime>
            <SeriesStationName>aws1</SeriesStationName>
            <Instances>
              <Instance />
              <fromImage>C:\Dimitropoulos DICOMs\00006285.dcm</fromImage>
              <comments>Lmin=456  Rmax=1424</comments>
              <date>04/01/08 03:43:18</date>
              <drawings>
                <drawing id="1" type="line">
                  <point>
                    <x>299</x>
                    <y>49</y>
                  </point>
                  <point>
                    <x>295</x>
                    <y>51</y>
                  </point>
                  <point>
                    <x>294</x>
                    <y>51</y>
                  </point>
                  <point>
                    <x>0</x>
                    <y>0</y>
                  </point>
                </drawing>
                <drawing id="2" type="rect">
                  <x>947</x>
                  <y>947</y>
                  <width>540</width>
                  <height>487</height>
                </drawing>
                </drawing>
              </drawings>
            </Instances>
          </Series>
        </Serieses>
      </Study>
    </Studies>
  </Patient>
  <Patient>
    <PatientName>John Doe</PatientName>
    <PatientID>12.03.1954</PatientID>
    <PatientBirthDate>Not Available</PatientBirthDate>
    <PatientSex>F</PatientSex>
    <Studies>
      <Study>
        <StudyInstanceUID>1.2.840.113619.2.66.2218496716.15919070102162245.20000</StudyInstanceUID>
        <StudyDate>02.01.2007</StudyDate>
        <StudyTime>16:22:45</StudyTime>
        <StudyDescription>Not Available</StudyDescription>
        <StudyID>36</StudyID>
        <Serieses>
          <Series>
            <SeriesInstanceUID>1.2.840.113619.2.66.2218496716.15919070102162245.10001</SeriesInstanceUID>
            <SeriesNumber>67</SeriesNumber>
            <SeriesModality>MG</SeriesModality>
            <SeriesDescription>Not Available</SeriesDescription>
            <SeriesDate>02.01.2007</SeriesDate>
            <SeriesTime>16:22:45</SeriesTime>
            <SeriesStationName>aws1</SeriesStationName>
            <Instances>
              <Instance />
              <fromImage>C:\euromedicaimages\cache2\00000ce7.dcm</fromImage>
              <comments />
              <date>06/13/08 01:24:43</date>
              <drawings>
                <drawing id="1" type="rect">
                  <x>447</x>
                  <y>661</y>
                  <width>540</width>
                  <height>487</height>
                </drawing>
              </drawings>
            </Instances>
          </Series>
        </Serieses>
      </Study>
    </Studies>
  </Patient>
</Patients>
</DicomDir>
```

Appendix B

The patient consent form used for the purposes of this research. It should be noted that this is a translated version of the original form, which was in Greek.



Study Number: ________
Patient Identification Number: _________

PATIENT CONSENT FORM

Title of Project: ***Pattern Recognition Systems Design on Parallel GPU Architectures for Breast Lesions Characterisation Employing Multimodality Images***

Name of Researcher: Konstantinos Sidiropoulos

Please initial box

1. I confirm that I have read and I understand the information sheet for the above study and have had the opportunity to ask questions to my doctor. ☐
2. I understand that my participation is voluntary and that I am free to withdraw at any time, without giving any reason, without my medical care or legal rights being affected. ☐
3. I understand that sections of any of my medical notes, such as medical images and accompanying demographic, clinical and histological data, may be looked at by responsible individuals that participate in this research. I give permission for these individuals to have access to my records. ☐
4. I agree to take part in the above study. ☐

Name of Patient	Date	Signature
Name of Person taking consent (if different from researcher)	Date	Signature
Researcher	Date	Signature