Microarray Image Processing:

A Novel Neural Network Framework

Bachar Zineddin

School of Information Systems, Computing & Mathematics

Brunel University

Uxbridge, West London, UK

A thesis submitted for the degree of

Doctor of Philosophy

May 2011

Dedicated to

My parents for all of their support over the years.

My sons Safa, Wafa and Adam.

My wife Mera, her name should be next to mine on this thesis.

Special thanks to my supervisors Prof. Zidong Wang and Prof. Xiaohui Liu.

Acknowledgements

I owe my deepest gratitude to my supervisors Professor Zidong Wang and Professor Xiaohui Liu for their enthusiastic guidance and advice throughout my research. Thank you for making the overall experience of my PhD most interesting. They have made available their support in a number of ways.

I am indebted to many of my colleagues to support me; Dr Karl Fraser, Daniel Morris. I would like to thank all to all other colleagues in the IDA group. Finally, I would like to show my gratitude to Aleppo University (Syria), who through their funding opened up the possibility to begin with.

Abstract

Due to the vast success of bioengineering techniques, a series of large-scale analysis tools has been developed to discover the functional organization of cells. Among them, cDNA microarray has emerged as a powerful technology that enables biologists to cDNA microarray technology has enabled biologists to study thousands of genes simultaneously within an entire organism, and thus obtain a better understanding of the gene interaction and regulation mechanisms involved. Although microarray technology has been developed so as to offer high tolerances, there exists high signal irregularity through the surface of the microarray image. The imperfection in the microarray image generation process causes noises of many types, which contaminate the resulting image. These errors and noises will propagate down through, and can significantly affect, all subsequent processing and analysis. Therefore, to realize the potential of such technology it is crucial to obtain high quality image data that would indeed reflect the underlying biology in the samples. One of the key steps in extracting information from a microarray image is segmentation: identifying which pixels within an image represent which gene. This area of spotted microarray image analysis has received relatively little attention relative to the advances in proceeding analysis stages. But, the lack of advanced image analysis, including the segmentation, results in sub-optimal data being used in all downstream analysis methods.

Although there is recently much research on microarray image analysis with many methods have been proposed, some methods produce better results than others. In general, the most effective approaches require considerable run time (processing) power to process an entire image. Furthermore, there has been little progress on developing sufficiently fast yet efficient and effective algorithms the segmentation of the microarray image by using a highly sophisticated framework such as Cellular Neural Networks (CNNs). It is, therefore, the aim of this thesis to investigate and develop novel methods processing microarray images. The goal is to produce results that outperform the currently available approaches in terms of PSNR, k-means and ICC measurements.

Contents

List of Figures

Abbreviations

AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
BP	**B**ack**P**robagation
cDNA	**c**omplementary **DNA**
CNN	**C**ellular **N**eural **N**etwork
CNN-UM	**CNN** - **U**niversal **M**achine
CC	**C**opasetic **C**lustering
CLD	**C**omplex **D**iffusion
DIP	**D**igital **I**mage **P**rpcessing
DNA	**D**eoxyribo**N**ucleic **A**cid
ESTs	**E**xpressed **S**equence **T**ags
FFT	**F**ast **F**ourier **T**ransform
FPGA	**F**ield-**P**rogrammable **G**ate **A**rray
GMM	**G**aussian **M**ixture **M**odel
GP	**G**ene**P**ix
GPU	**G**raphics **P**rocessing **U**nit
ICC	**I**ntra**C**lass **C**orrelation
IDA	**I**ntelligent **D**ata **A**nalysis
LD	**L**inear **D**iffusion
LMS	**L**east **M**ean textbfSquare
MLP	**M**ulti**L**ayer **P**erceptron
mRNA	**m**essenger **RNA**
MSE	**M**ean **S**quare **E**rror
NIH	**N**ational **I**nstitutes of **H**ealth
NSE	**N**avier **S**tokes **E**quation
PCR	**P**olymerase **C**hain **R**eaction
PDE	**P**artial **D**ifferential **E**quation
PDP	**P**arallel **D**istributed **P**rocessing
PSNR	**P**eak **S**signal to **N**noise **R**atio
PWL	**P**iece**W**ise **L**inear
RNA	**R**ibo**N**ucleic **A**cid
SCIR	graph**S**-**C**ut **I**mage **R**econstruction
SOM	**S**elf-**O**rganizing **M**ap
SRG	**S**eeded **R**egion **G**rowing
TIFF	**T**agged **I**mage **F**ile **F**ormat

Author's Publications and Presentations

Some of the new work presented in this thesis has been previously published/submitted for publication.

Chapter 3	Zineddin, B., Wang, Z. & Liu, X., 2010. A Novel Neural Network Approach To cDNA Microarray Image Segmentation. Pattern Recognition Letters, submitted.
Chapter 4	Zineddin, B., Wang, Z. & Liu, X., 2010. cDNA Microarray Segmentation: Adaptive Approach. IEEE Transactions on Image Processing, Submitted.
Chapter 5	Zineddin, B., Wang, Z. & Liu, X., 2010. A Multi-View Approach to cDNA Microarray Analysis. International Journal of Computational Biology and Drug Design, 3(2), pp. 91-111.
	Zineddin, B. et al., 2008. Investigation on Filtering cDNA Microarray Image Based Multiview Analysis. In S. Zhang & D. Li, eds. Proceeding of the 14th International Conference on Automation & Computing. London, UK: Pacilantic International Ltd., pp. 201-206.
Chapter 6	Zineddin, B., Wang, Z. & Liu, X., 2010. A Microarray Image Multiview Analysis Based on Cellular Neural Network. IEEE Transactions on Neural Networks, Submitted.
Chapter 7	Zineddin, B., Wang, Z. & Liu, X., 2010. Cellular Neural Networks, Navier-Stokes Equation and Microarray Image Reconstruction. IEEE Transaction On Image Processing, Revised.
	Zineddin, B., Wang, Z. & Liu, X., 2010. Cellular Neural Networks, Navier-Stokes Equation and Microarray Image Reconstruction. In IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW). HK, pp. 234-239.
	Zineddin, B., Wang, Z. & Liu, X., 2010. A Cellular Neural Network for Microarray Image Reconstruction. In Proceedings of the 16th International Conference on Automation & Computing. Birmingham, UK, pp. 106-111.

Chapter 1

Introduction

1.1 Motivation

Arguably, the conquest of the unknown is the best description of human history during the last few millenniums, which is the most startling one despite all other aspects of this history. It is the journey of human to understand the life, themselves and their relation to the universe. Through several thousands of years many schools of thoughts have tried to give the right answer; or at least give a pathway to approach this answer. Starting from the far east Taoism, Indian Buddhism, Mesopotamia and ancient Egypt civilizations, through the Greek and their recent descendants, the Islamic and the last surviving, West civilizations, everyone has its own set of rules that shape these pathways.

Regardless of all the past discoveries which are parts of a long debate about accepting them as myths or mere other point of view of the same reality, it is the most sparkling milestone achievements, which have been attained during the last two centuries by some great minds in the human history, that will be the basis for this current study.

Starting from Gregor Mendel, who can be considered as the father of the modern genetics, although his work stayed obscure until the beginning of the twentieth century, to the most astonishing discoveries in the last century, the chromosomes, the structured of the Deoxyribonucleic Acid (DNA) and the rules of the genetic material within the living cell have been discovered.

Within the last thirty years, many developments have been achieved and our understanding of biology of the living systems has started to build up in wide steps. However, most of these advances until mid nineties were limited to the ability to sequence the DNA. A little has been done to improve the methods that allow human to grasp a deep insight about the functions of the genes and the relations within the genetic material and between these materials and the surrounding environment. Within this context, the microarray technology appeared as an effective throughput technology for functional genomics.

With the introduction of this technology, a new era has begun. The huge data that have been resulted from just a few experiments led to the whole novel techniques within the data analysis methodology to cope with the high demand of such outputs. The developed techniques covered many study areas from the data storage management tools, Intelligent Data Analysis (IDA), to Digital Image Processing (DIP) and many more.

The analysis of the microarray image, though it might seem an easy and straightforward task, is a difficult and challenging problem. Therefore, it is our aim in this thesis to tackle this issue by developing and applying up-to-date techniques that merge many highly potential ideas from various disciplines such as Artificial Neural Networks (ANNs), Partial Differential Equations (PDEs) and Linear Diffusion (LD) filtering techniques.

1.2 Goal of The Thesis

1.2.1 The State-of-the-art

Due to the recent success of bioengineering techniques, many large-scale analysis tools have been developed to investigate the functional organization of the cell. Among them, cDNA microarray has emerged as a powerful technology that enables biologists to study thousands of genes simultaneously within an entire organism and, thus, to obtain a better understanding of the genes' interaction and regulation mechanisms involved. However, this technology is still in the beginning of getting its potentials, and lots of improvements are required in all the stages of the microarray process.

Although microarray technology has been developed so as to offer high tolerances, there exists large signal irregularity through the surface of the microarray image. The imperfection in the microarray image's generation process causes noises of various types, which contaminate the resulting image. These errors and noises will propagate down through, and can significantly affect, all subsequent processing and analysis. Therefore, to realize the potential of such technology it is crucial to obtain high quality image data that will indeed reflect the underlying biology in the samples. One of the key steps in extracting information from a microarray image is the segmentation with aim to identify which pixels within an image represent which gene. This area of spotted microarray image analysis has received relatively a little attention comparing to the advances in proceeding analysis stages. This attitude towards image analysis was in part due to first, the believe that this stage has little impact on the later stages and, second, the need for multi-disciplinary knowledge was crucial to build effective and efficient algorithms. It is worth pointing out that the lack of advanced image analysis including the segmentation results in sub-optimal or even poor quality data being used in all downstream analysis methods.

Some of the research papers, which have played an essential role in the development of proposals presented in this thesis, are now discussed as follows.

Arena *et al.* [2002a] proposed a real-time algorithm to process the microarray image using the Cellular Neural Network (CNN) array. The CNN is an analogic processor array [Chua & Roska, 1993b; Chua & Yang, 1988b,d] that allows the application of a local strategy, with less computational complexity, to meet the task requirements. It is important to note that using spatial information leads to a robust and reliable algorithm in some applications such as microarray image analysis. Due to its architecture, the two dimensional CNN array is widely used to solve image processing and pattern recognition problems. Furthermore, the parallelism of this structure allows one to perform the most computationally expensive image analysis tasks faster than the classical CPU-based computer does.

Essentially, the algorithm in [Arena *et al.*, 2002a] used two colors, red and green. The first step consists of filtering operations; this process cleans the background noise and removes the ill-positioned spot. The next step is to remove patches and irregular size spots. After these operations, the following phase addresses the intensity analysis of both channels.

In [Samavi *et al.*, 2004], a pipeline architecture was designed particularly for real-time and semi-parallel microarray processing. This architecture, first, produces a binary map using a threshold value. Then, the map goes through gridding and morphological operators. Next, the masking operator assigns the original gray scale value to pixels that have 1's in the map. The last step is the intensity analysis; image is compared with a number of thresholds to the intensity level of each spot.

Hirata *et al.* [2002] presented a technique based on mathematical morphology that performs the segmentation. The procedure of image segmentation consists, basically, of: 1) Creating one gray-scale image from the red and the green images; 2) Rotation correction; 3) Sub-array and spots gridding; and 4) Spot segmentation. The segmentation stage, in particular, takes both the morphological gradient of the filter image and a marker image, which is composed of the approximated centers of the spots and the grid itself. Then, a watershed operator [Beucher, 1982; Soille & Vincent, 1990; Vincent & Soille, 1991] should be applied with a basic cross structuring element. Another algorithm based on mathematical morphology was proposed in [Angulo & Serra, 2003]. In this algorithm, initially, a gridding algorithm automatically produces spots' quadrants, which will be analyzed individually in later stages. Then, the analysis of the spot quadrant images is achieved in five steps including spot size estimation, background noise extraction, spot position determination, spot boundary definition which is carried out using the watershed transformation and, finally, spot signal quantification. Siddiqui *et al.* [2002] applied another algorithm to carry out the segmentation process that is based on mathematical morphology and watershed technique, followed by quantification of the shapes of the segmented spots using B-Splines [Cohen *et al.*, 2001].

Fraser [2006] proposed many mapping functions that place emphasis on certain frequencies or regions of interest. This methodology would not only be advantageous but also be more effective in terms of the overall goal. These functions are named as inverse, a summed inverse and square root, a square root and a linear function. For instance, if the image is first filtered using the inverse function to emphasize the genes, the clustering output is improved dramatically. Similarly, photographic polarizing filter allows details normally hidden behind reflections to be captured. Therefore, filtering the image in this way allows us to analyze details that would otherwise be lost.

1.2.2 Thesis Contributions

Recently, much research on microarray image's segmentation with many methods has been proposed, some methods produce results better than others. In general, the most effective approaches require considerable run time (processing power) to process an entire image. Furthermore, there has been little progress in developing sufficiently fast yet efficient and effective algorithms for the segmentation of a microarray image by using up-to-date techniques such as Artificial Intelligence (AI) approaches, specifically neural networks. It is, therefore, the aim in this thesis to investigate and develop novel analyzing methods for microarray images. In other words, our goal is to produce results that outperform the results of currently available approaches in terms of robustness, effectiveness and time efficiency. The following issues will be investigated in this thesis:

1) Fully Automation: The large number of spots, usually in thousands, as well as the shape and position irregularities [Lukac *et al.*, 2005; Wang *et al.*, 2003b] can propagate processing errors through subsequent analysis steps [Eisen & Brown, 1999; Lukac *et al.*, 2004; Wang *et al.*, 2003a]. Furthermore, the time-consuming manual processing of the microarrays has led to the interest in using a fully automated procedure to accomplish the task [Bajcsy, 2004; Jain *et al.*, 2002; Katzer *et al.*, 2003]. Automatic algorithm guarantees the elimination of user's errors and is considered as an essential step towards a unified framework for microarray image analysis.

2) Local Strategy: Considering the signal variability across the microarray image, the emphasis will be placed on techniques that benefit from the local information in order to achieve different tasks in the image analysis process. Therefore, it can be assumed that the global image properties are irrelevant and, rather, a locally adaptive strategy with less computational complexity can meet the application requirements.

3) Image Filtering: The system imperfections and the microarray generating process restrict our ability to differentiate between spots signals and background signal as well as our ability to get accurate measures of interest in the images. Therefore, it makes much more sense to produce filtered versions of the image data so that the dynamic range of the image can be increased, and hence a better ability of signal extraction could be achieved.

4) Multi-view Analysis: The mapping functions, proposed in Fraser [2006], will be investigated thoroughly in order to draw a full understanding of their effects on the image analysis process. In addition, the ability to approximate these functions in any possible hardware implementation will be addressed.

5) Image Segmentation: It has been anticipated that the spot intensity value would be independent of the segmentation algorithm if a background correction method is utilized [Yang *et al.*, 2002]. However, Lehmussola *et al.* [2006] showed that even with a correction procedure the segmentation method will significantly influence the identification of differentially expressed genes. Therefore, investigating the segmentation process will occupy a considerable part of this thesis.

6) Image Reconstruction: Considering the inherent variation between the gene and background regions, image reconstruction is one of the best techniques that can be applied to estimate the local background of every spot. By the end of this thesis, some of the most popular reconstruction techniques for the general imagery will be adopted to be applied on the microarray image.

7) PDE: The filtering techniques based on PDEs will be investigated in order to, first, benefit from their potential in the image processing applications; second, utilize the fact that PDEs can be used to build hardware architectures which approximate successfully these systems.

8) CNN: The ultimate goal of this study is to draw a road map toward a fully automated and parallel hardware implementation of the image processing steps. CNN is one of the best options to achieve this goal. Due to its architecture, the two-dimensional CNN array is widely used to solve image processing and pattern recognition problems. Furthermore, the parallelism of this structure allows one to perform the most computationally expensive image analysis tasks faster than they do in classical CPU-based computer.

1.3 Overview of The Chapters

Figure 1.1 gives a general view about the thesis structure. In the next chapter the background review of our research is presented. Chapter 2 covers the the underlying biology behind the microarray technology, the development that led to this high throughput technology, conducting the microarray experiment and, finally, a comprehensive review of the microarray image processing.

In Chapter 3, we will propose a new segmentation method utilizing a series of Multi-Layer Perceptron (MLP) and Kohonen neural networks. The presented networks have a new structure that is particularly suitable to the task at hand.

In Chapter 4, the image is filtered by applying a nonlinear anisotropic diffusion with several diffusion functions in order to increase the dynamic range of the image, and thus to increase the ability to extract desired signals. The proposed novel algorithm is based on the CNN computational paradigm integrated with median and anisotropic diffusion filters.

The investigation, in Chapter 5, is to improve the processes involved in the analysis of microarray image data. The main focus is to clarify the image features' space in an unsupervised manner. Instead of using the raw microarray image, it is suggested that

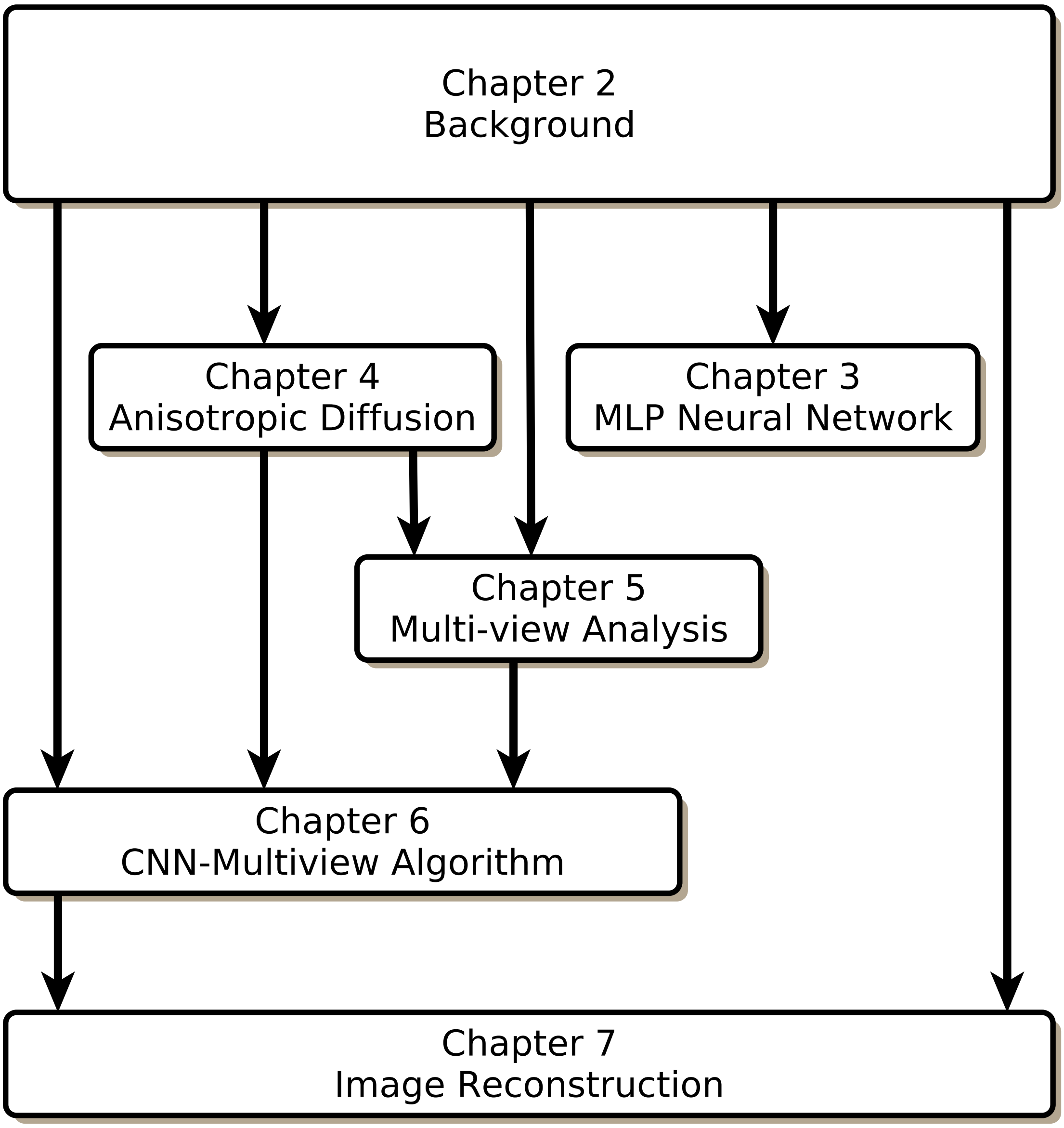


Figure 1.1: The structure of thesis

producing multiple views of the image data, i.e. highlighting certain frequencies, will yield more reliable results in the filtering stage. In addition, this methodology will be advantageous for the segmentation stage and the system as a whole. Therefore, the multi-view analysis framework is combined with many filters such as Median, Top-Hat and Complex Diffusion (CLD). Then, a thorough analysis is conducted to understand the effects of these filters on the segmentation stage.

In Chapter 6, an automatic and non-supervised algorithm for a fast and accurate pixels' classification is presented. The proposed approach in this chapter follows two lines: first, the development of CNN algorithm to approximate any mapping function and, second, the integration of diffusion based filter with an adaptive segmentation algorithm. The main advantages of this approach are that it establishes a general framework that can make the whole cDNA microarray technology fully parallel as well as it minimizes the processing time.

Two new image reconstruction algorithms based on CNN are proposed in Chapter 7. The presented methods tend to get an exemplary approximation of the background in the gene spot region either by using diffusion based algorithm or by solving the Navier-Stokes equation. These algorithms offer robust methods to estimate the background signal within the gene spot region. Subtracting this reconstructed background from the original image should lead to a more accurate quantification of genes' signals.

Finally, conclusions and suggested future works are presented in Chapter 8.

Chapter 2

Background

Imagine watching a play with thirty thousand actors.
You'd get pretty confused.

James Watson

2.1 Introduction

Certainly, Gregor Johann Mendel's results can be considered as a major milestone in the quest to understand the nature and the content of genetic information, though the theoretical realization of the genetics was not yet extended by momentous ideas that have to be presented in the twentieth century. Although Mendel did not know the physical units that represent the genetic information, his experiments showed quantitatively how to predict the underlying inheritance. His work, therefore, established a theory to explain this heredity based on assumed factors; we can now understand them as genes. Unfortunately, Mendel's paper, "Experiments in Plant Hybridization", had been ignored until many scientists rediscovered his conclusions and cited it at the beginning of the 20th century.

In the following decades, many milestone achievements can be specified: 1) Chromosome was discovered and the relation between the heredity, development and the chromosomes have gradually become significantly evident. 2) Then, as a major breakthrough, Watson & Crick [1953] revealed the molecular basis of the heredity, the DNA double helix. 3) The next astonishing establishment was unlocking the informational basis of heredity. 4) The biochemical studies, which were carried out during 1950s by a group of physicists and chemists [Roberts *et al.*, 1955], uncovered the basic mechanisms involved in the regulations of metabolic pathways. 5) The invention of the recombinant DNA technologies of cloning and sequencing [Kleppe, 1971], with the knowledge of the biological mechanism, enabled the scientists to read the genetic information.

In 1990, the National Institutes of Health (NIH) introduced a new method to accelerate the gene discovery [Adams *et al.*, 1993]. Molecules, which are called Expressed Sequence Tags (ESTs) and evolved from the development of Complementary DNA (cDNA), provide scientists with a quick and inexpensive tool for gene identification, gene expression and genome mapping [Gerhold & Caskey, 1996; Marra *et al.*, 1998; McLachlan *et al.*, 2004; Quackenbush, 2001]. Mapping genes and sequencing DNA enable researchers to draw conclusions about the gene function from its structure. In addition, the sequencing of specific genome offers a direct access to genome molecules using the Polymerase Chain Reaction (PCR). With these sequences, biologists will be able to design primers, which can amplify a particular DNA fragment, or they can effectively screen a library for a larger segment of DNA containing the region of interest. Furthermore, sequencing increases the ability to search efficiently for similar genes in other organisms.

The (PCR) [Rabinow, 1997] and automated DNA sequencing technologies [Hegde *et al.*, 2000] were remarkable achievements. While the PCR methodologies allowed researchers to amplify a very small amount of DNA, the automated DNA sequencing approach enhances the ability to obtain the entire DNA sequence of microbial genomes within few days. These achievements have initiated the human genome project [Gene, 1999]. The PCR, as an amplification tool, is an important part of DNA analysis for many reasons: 1) It provides a method to an essentially limitless supply of material. 2) The PCR is a procedure that uniformly amplifies the sample so this sample will not be altered, distorted, or mutated by the process of amplification. 3) Another rationale behind DNA amplification is the amplification selectivity of a region which is an easy way to purify that segment from the bulk. With sufficient amplification magnitude, the DNA product becomes such a major material of the amplification mixture; i.e., the starting sample is reduced to a sufficiently small percentage for most applications.

Scientists devote their efforts to sequence and assemble the genomes of various organisms, especially the human [Lander *et al.*, 2001; Ledford, 2007; Wheeler *et al.*, 2008]. However, genome sequencing is merely the transfer of the information contained within the DNA which serves as the repository of the genetic information, to another carrier. Genome sequencing provides raw data which does not provide a way to get what the data means or how the data can be utilized in various clinical applications. On the other hand, besides obtaining a genomic sequence and specifying a complete set of genes in any sequencing project, the ultimate goal is to get an understanding of the functionality of this genetic information. In other words, gaining insight about the function of a specific gene or set of genes in the normal circumstances increases our ability to understand the deceased situations. Investigating the relationships among genes in the DNA is another major goal in molecule biology [Cohen, 2005]. This new field of study, which is known as functional genomics, led to the emergence of new high throughput technologies that allowed researchers to tackle difficult problems and reveal promising solutions in many fields, i.e. cancer research [Ley *et al.*, 2008].

DNA microarray is a remarkably successful high throughput technology for functional genomics [Schena *et al.*, 1995; Shalon *et al.*, 1996]. Microarrays allow researchers to analyze the expression level, in different cell types or conditions, of many thousands of genes in a single experiment [Alizadeh *et al.*, 2000; Moore, 2001]. Basically, a DNA array [Stekel, 2003] can be defined as an orderly arrangement of tens to hundreds of thousands of unique

DNA probes of known sequence. The probes, which are either individually synthesized on the array surface or pre-synthesized by a procedure such as PCR, are attached to the array platform. The cDNA microarray aim to detect the abundance of various mRNA molecules of a cell by using hybridization of the fluorescent labeled samples to the DNA probes already existed on the array [Schena *et al.*, 1995]. This abundance can provide information about the related protein or the expressed gene [Gygi *et al.*, 1999]. The end product of the cDNA microarray experiment is a scanned array image, see Figure 2.1, [Moore, 2001; Orengo *et al.*, 2003]. These images must then be analyzed to identify the arrayed spots and measure the relative fluorescence intensities for each feature.

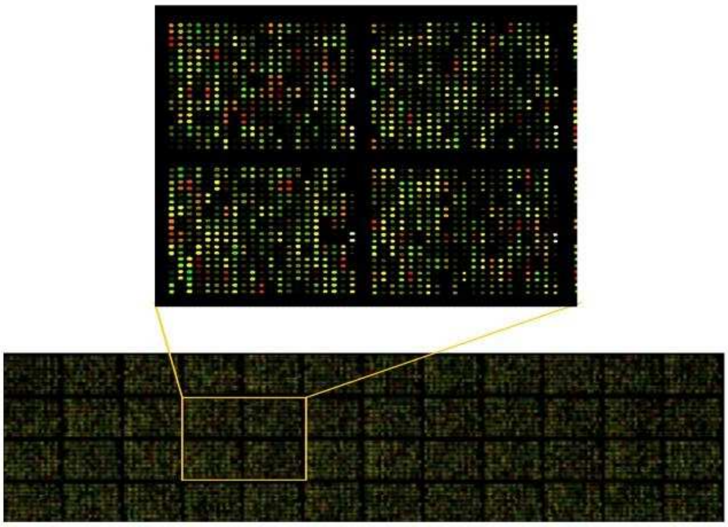


Figure 2.1: cDNA Microarray Image

2.2 Molecular Biology

2.2.1 The Structure of The DNA

By the early beginning of the 20th century, Mendel's works have become widely known [Klug & Cummings, 2003] and got fully credited by Bateson [Bateson, 2007]. A couple of years later, Sutton [1903] suggested the applicability of the Mendelian laws of inheritance to chromosomes. Sutton suggested that chromosomes contain units ("genes") which affect the heredity and the development. Furthermore, he showed that meiosis, the process by which the genetic material of eukaryotic cells is duplicated and distributed during cell division, gives a mechanism for Mendelian inheritance. Two decades later, the experiments of T.H. Morgan [Allen, 1978] on the fruit fly verified Sutton's hypothesis. Yet, the genes, their roles or how they produce physical features in organisms, were still to be discovered years later.

The discovery of the nucleic acid in the 19^{th} century by the biologist Friedrich Miescher [Allen, 1978] marked an important step toward a more advanced understanding of the heredity in the last century. By the late 1920s, chemists had discovered both ribonucleic acid (RNA) and deoxyribonucleic acid (DNA). Levene [1919] investigated the structure and the function of the nucleic acids and highlighted different forms of them, namely the DNA and RNA. However, the connection of nucleic acids with genes was yet to be established a decade later [Lorenz & Wackernagel, 1994]. The studies, back then, showed that the chromosomes contain DNA, which is suspected to be the genes' material. By the end of 1940s, DNA is proven to be the right answer, and this was finally determined experimentally.

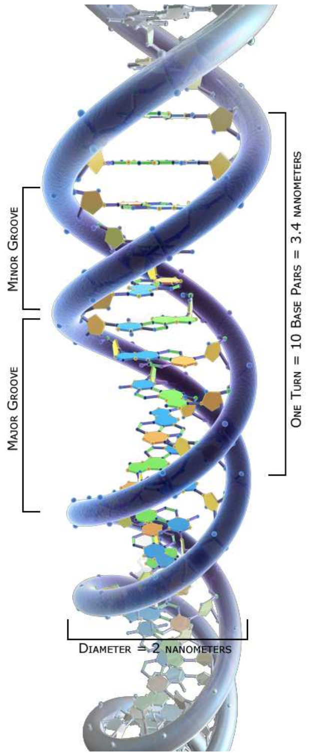


Figure 2.2: DNA structure.

Zygote Media Group.

The biggest breakthrough came when Watson & Crick [1953] proposed a model of double helix as the structure of the DNA, see Figure 2.2. This model is accepted now as the first correct structure of DNA. The model of DNA was based on a single X-ray diffraction image, which was called “Photo 51” and introduced in [Franklin & Gosling, 1953], see Figure 2.3. Those findings led to the proposal of the Central Dogma of the molecule biology, which describes the relation between DNA, RNA and proteins [Crick, 1955]. Soon later, Meselson & Stahl [1958] confirmed the replication mechanism that was implied by the double helix model.

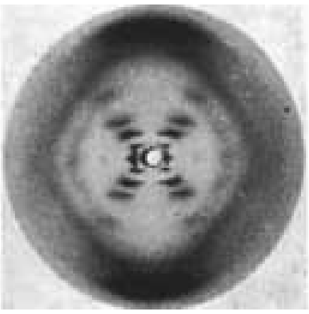


Figure 2.3: This image is a faithful digitalization of the famous historic photograph Photo 51, the name is given to an X-ray diffraction image of DNA introduced in [Franklin & Gosling, 1953]. http://en.wikipedia.org/wiki/File:Photo_51.jpg

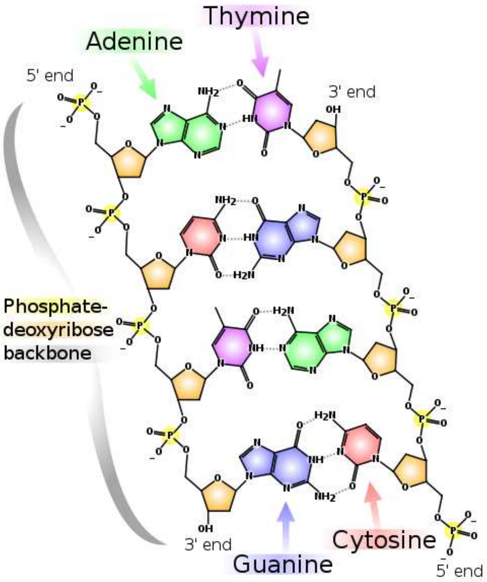


Figure 2.4: Chemical structure of DNA. Hydrogen bonds shown as dotted lines.

Madeleine Price Ball.

DNA is the basic hereditary material in all cells and contains all the necessary information to make proteins. DNA is a linear polymer made up of nucleotide units. The nucleotide unit consists of a base, a deoxyribose sugar and a phosphate, see Figure 2.4. There are four types of bases: adenine (A), thymine (T), guanine (G), and cytosine (C). In DNA, the bases are perpendicular to the helix axis and form pairs: A to T and G to C. This relation is called a complementarity and contributes to the determination of the whole DNA shape, which gives a high robust reproduction of the basis sequence in the DNA chain. This invariant relation is used in the cell to duplicate the DNA during cellular reproduction and in protein synthesis. The double helix of DNA can be divided into two helices and then recomposed with a process called hybridization. Each specific protein is built starting from a specific DNA sequence within the whole DNA chain, called a gene. RNA chemicals are very similar to the DNA ones. However, there are some differences, RNA contains uracil (U) instead of thymine (T), RNA is a single stranded and finally RNA contains ribose sugar instead of deoxyribose sugar, see Figure 2.5.

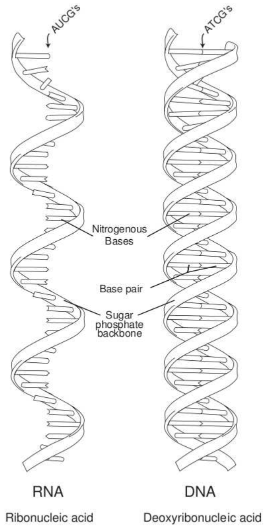


Figure 2.5: Ribonucleic Acid (RNA). National Institute of Health

2.2.2 Central Dogma

The genetic information is stored in the DNA strands. Segments of these strands encode genes. Generally, each gene produces a particular protein by coding each of the amino acids that make up the protein. Every amino acid is encoded by three nucleotide bases, for instance, the nucleotides “AAG” correspond to the amino acid phenylalanine. This three-letter code is called a codon, see Figure 2.6. In order to understand the process of producing proteins from genes, central dogma of molecular biology should be considered. Briefly, genes are transcribed into messenger RNA (mRNA) and mRNA, and then translated to form proteins, which are the building blocks and functional elements of any living cell. Gene's expression is the indication of the presence of this mRNA.

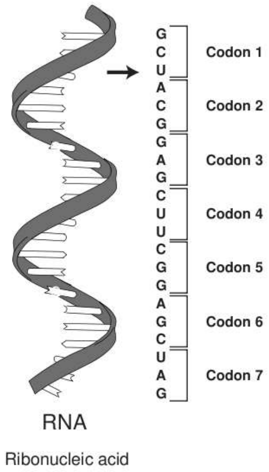


Figure 2.6: The codon. National Institute of Health

Figure 2.7 shows the genetic information flow inside the cell. The first step is the transcription which happens in the cell nucleus. From a single strand of the DNA, a protein, also called enzyme, sets the strands apart in a small section of the DNA. This enzyme then uses one of the DNA strands to create the mRNA molecule by a letter-for-letter copy of this section; i.e., in every place where the gene has (C), the mRNA has (G), and in every place where the gene has (A), the mRNA has (U). Therefore, the RNA molecule transcribed from the gene is complementary to the coding strand of that gene.

The stability is a main character of DNA. The same copy of genomic DNA exists almost in all the cells of an organism. Yet, cells are different in shape and function. These differences between them are due to the different subsets of expressed genes in each of the different cell types. In addition, different stimuli provoke different subsets of genes to be expressed. Thus, the pattern of gene expression levels reflects both the cell type and its condition. Microarrays allow researchers to detect the abundance of various mRNA molecules or transcripts in a cell at a given moment. Although the relation between the abundance of mRNA and the corresponding protein is not necessarily straightforward [Gygi *et al.*, 1999], it is accepted that the quantity of each mRNA detected in the cell can provide information on the corresponding protein.

One example that reflects how important it is to characterize protein construction is that many human diseases are due to the failed synthesis of a particular protein; for ex-

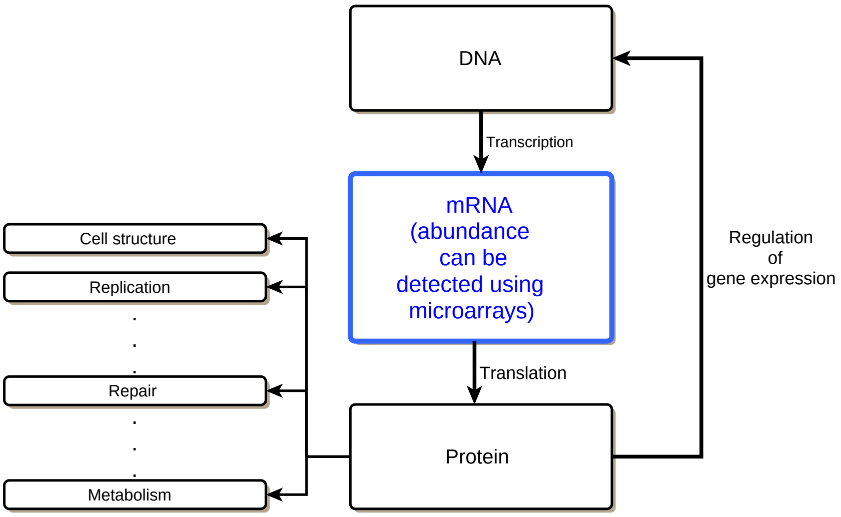

DNA
Transcription
mRNA (abundance can be detected using microarrays)
Translation
Protein
Regulation of gene expression
Cell structure
Replication
Repair
Metabolism

Figure 2.7: The information transfer between DNA, mRNA and protein (the ”central dogma”). Segments of DNA are used as a code to make mRNA, which is used as a code to make protein. Microarray experiments exploit the relation between mRNAs and the genes that encode them.

ample, insulin. The target here is to induce specific micro-organisms such as bacteria to synthesize that specific protein. Here, the first step is to know which gene is responsible for the production of the desired protein. After that, the selected gene is inserted into the micro-organism obtaining a kind of protein factory. Another example is the important exploration field that involves the study of the so-called oncogenes; i.e. genes that when mutated or expressed at abnormally high levels contribute to the conversion of a normal cell into a cancer cell. Furthermore, in [Evans, 1999], mutations in genes are used as important determinants of drug effects, which represents a key issue in the emerging pharmacogenomics discipline. These are only few examples of the huge possibilities that can be derived from the discovery, characterization and recognition of genes.

Every protein is formed from a specific sequence of the 20 amino acids. Their displacement information is transferred from the DNA through a translation code formed by a subsequence of three bases of nucleotides. So, each amino acid relates to a specific sequence of three bases, the codon. Therefore, in the classification of proteins, the major issue to be unraveled is to discover, for each protein, the sequence of bases that generated it. When a particular gene codifies a protein, it is said to be expressed into the protein. Since even the most powerful microscope is unable to distinguish among genes, new methodologies are required to gain the global gene expression profile. Many laboratories are working to make a database of gene structures as soon as they are discovered.

2.3 Microarrays Technology

2.3.1 Introduction

Microarray technology came on time to cover the need to monitor in parallel all the DNA sequences and to have the adequate sensibility to detect the variation of gene expression. Furthermore, this technology affects the data volume that can be acquired during a limited time. The crucial impact of facilitating this technology was the boost of our understanding of organisms and various biological processes. During 1990s, some parallel methods have been introduced, allowing the possibility to detect the expression level of a huge number of genes simultaneously. The first method, devised in the laboratory of Pat Brown at Stanford [Schena *et al.*, 1995], is based on the robotic micro-deposition and the fixing of DNA single-stranded fragments in microarrays mounted on microscope slides with a size of 2 cm × 2 cm. The second method depends on the high density spatial synthesis of oligonucleotides [Lipshutz *et al.*, 1999]. Other methods depend on the development of *in situ* synthesis with reagents delivered by ink-jet printer devices [Hughes *et al.*, 2001].

The development of microarray technology and its success are sparked by the introduction of many innovations in recent decades. The highly specific preferential binding of complementary single-stranded nucleic acid sequences was first exploited experimentally in the mid 1960s. This method achieved a remarkable success in the form of a technique which is called the Southern blot [Gillespie & Spiegelman, 1965; Southern, 1975]. Some other innovations are the progress in the genome sequencing, the advances in miniaturization and the high density synthesis of nucleic acids on non-porous solid supports, such as glass, nylon or silicon. Microarrays were first used to study global gene expression in [DeRisi *et al.*, 1997].

Remarkably, the microarray technique represents a real breakthrough in biological and medical fields, since all traditional gene expression detection methods provide gene information in a sequential way. The availability of genes' data-bases enables researchers to study all the genes belonging to a given organism simultaneously. Therefore, the researcher will obtain a quantitative information about cellular pathways and will observe the effect of different physiological conditions on such pathways by direct comparison between the expression levels of the genes. Microarray has already been facilitated in a wide range of applications; notably, for novel gene discovery, expression profile analysis, drug discovery and development, investigating biochemical pathways, diagnostics, therapeutics and proteomics [Holloway *et al.*, 2002; Leung & Cavalieri, 2003; Samartzidou *et al.*, 2001; Schena *et al.*, 1998].

2.3.2 Gene Expression

Gene expression is highly valuable for exploring genetic regulations such as investigating metabolism. In addition, investigation of gene expression forms a very effective methodology in the molecular medicine such as classification of disease, diagnosis, prognostic prediction and in a number of industrial and pharmaceutical applications [Cohen, 2005].

Maybe it is a common observation that biologists have found many genes to be co-regulated [Eisen & Brown, 1999; Eisen *et al.*, 1998] in an extremely efficient way. Co-regulation under various biological conditions means that there is a relative similarity among the corresponding expression profiles [Chou *et al.*, 2007]. These genes include the genes of nutrition, stress responses and metabolic pathways. Some other co-expressed genes are the genes encoding the ribosome, the proteosome and the nucleosome [Alon, 1999; Brown & Botstein, 1999; Causton *et al.*, 2001; Eisen *et al.*, 1998; Hughes *et al.*, 2000; Lashkari *et al.*, 1997].

The gene expression profile is the measurements of gene expression of the genes under study. Therefore, it can be considered as a representation of the molecular definition of a cell in a specific state [Young & Center, 2000]. Obtaining adequate information about the transcriptional profile of biological sample is very important. Expression profile is a way for describing a phenotype, which can be a complete set of observable inherited characteristics of an organism [Cantor & Smith, 1999]. Furthermore, the ability to profile and match patterns for a large number of biological samples has been used to infer the function of un-characterized genes or some supposed drug targets [Gray *et al.*, 1998; Hughes *et al.*, 2000; Marton *et al.*, 1998]. For that reason, the National Cancer Institute offers access to databases integrating gene expression profiles data from 60 human cancer cell lines in order to be used for cancer research and drug design research [Ross *et al.*, 2000; Scherf *et al.*, 2000; Staunton *et al.*, 2001; Weinstein *et al.*, 1997].

The necessity for gene expression data in fields such as oncology has been highlighted by the crucial application of gene expression for accurate and early diagnosis and treatment. In addition, gene expression data has been used to specify a tumor type in clinical samples, define a new subtype, identify misclassified cell lines and predict prognostic outcomes [Alizadeh *et al.*, 2000; Bittner *et al.*, 2000; Golub *et al.*, 1999; Perou *et al.*, 1999; Shipp *et al.*, 2002]. With such a powerful technology, the personalized medicine, in which the specific underlying problem can be identified and the prognosis can be predicted, has a high potential in the near future. Thus, the treatment can be altered based on the genetic information of the patient and the specific characteristics of the tumor in order to reduce the likelihood of unwanted side effects.

Pharmaceutical industries use microarray technology extensively at various stages of drug design starting with a high throughput screening of small molecules, then identifying possible drugs, drug target identification and assessment of toxicity. This situation has been sparked by the robustness of gene expression as a representative of biological characteristics for a wide range of samples under numerous conditions.

2.4 Microarray Experiment and Data Analysis

In general, a microarray is a chemically-treated microscope slide of glass, nylon membranes or other specialized substrates. Onto this slide, an orderly arrangement of nucleic acid samples, each representing a particular gene, is typically placed at fixed locations called spots. There may be tens of thousands of spots on an array. Each spot contains tens of millions of identical DNA molecules with lengths from tens to hundreds of nucleotides. Afterwards, the microarray slide is exposed to a set of labeled cDNA samples, which are derived from tissue of interest. With the completion of hybridization reaction, the amount of the target that bounds to each sample is measured with the aid of image capturing devices and computer technology. The measurement is based on the intensity of the spot. Theoretically, in order to carry out gene expression studies, each molecule should represent a single cDNA molecule or transcript for a specific gene. In practice, however, it is not always possible to identity sequences that monitor the expression of a specific transcript unambiguously due to the presence of families of similar genes. The spots are either printed on the microarray by a robot or ink-jet printing, or synthesized in situ by photolithography.

2.4.1 Process Summary

There are many ways by which researchers can use microarrays to measure gene expression levels. One of the most popular microarray applications is to compare the gene expression levels in two different samples. The basic idea is to label the extracted mRNA from each of the samples using two different dyes, for instance, a green label for the sample from the first condition and a red one for the sample from the second condition.

In a nutshell, the hybridized microarray is excited by a laser and scanned at wavelengths suitable for the detection of the applied dyes. The amount of fluorescence emitted relates to the amount of nucleic acid hybridized to each Spot. Assuming that the nucleic acid from the first sample is emitting green light, while the nucleic acid from the second sample is emitting red light, then, if both are equal, the spot will be: yellow, and if neither is present it will not fluoresce and so appears black. Thus, from the fluorescence intensities and colors for each spot, the relative expression levels of the genes in both samples can be estimated. Therefore, information about expression of thousands of genes can be obtained from a single experiment. On the other hand, the same principles have been facilitated in the other platforms for obtaining gene expression profiles.

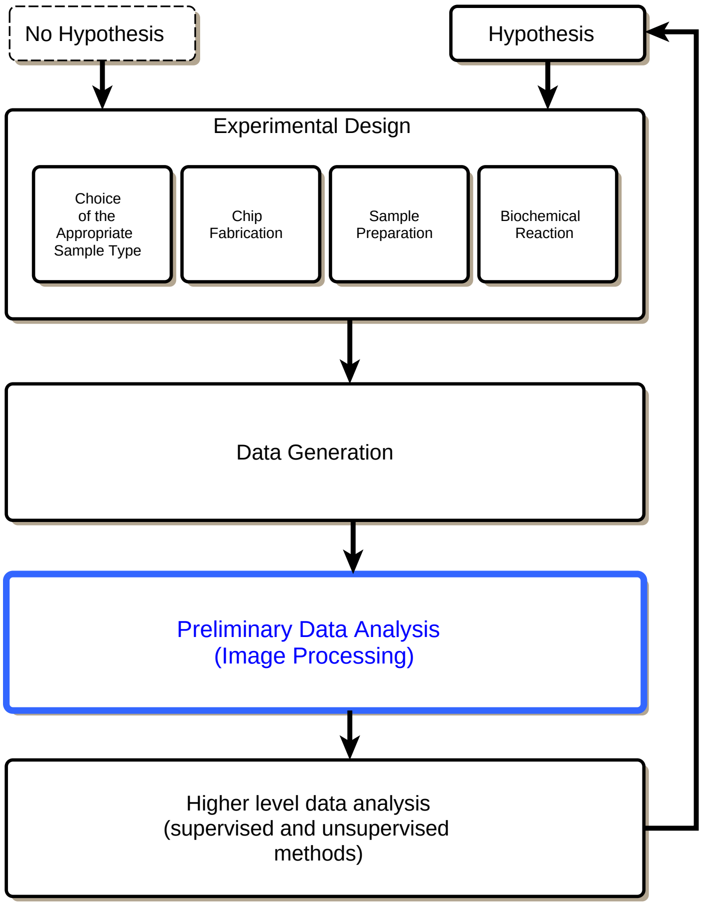

No Hypothesis
Hypothesis
Experimental Design
Choice of the Appropriate Sample Type
Chip Fabrication
Sample Preparation
Biochemical Reaction
Data Generation
Preliminary Data Analysis (Image Processing)
Higher level data analysis (supervised and unsupervised methods)

Figure 2.8: An overview of a microarray experiment and data analysis. After designing the experiment, the data should be acquired. These data must undergo a preliminary analysis basically by image processing step and quality assessment. The higher level analysis may involve various methods in order to carry out normalization and clustering. Usually, these analyses are relevant to the biological samples, the information required and the original hypothesis to be tested.

A microarray experiment generally consists of four distinct steps, see Figure 2.8. They are, 1) the design of the experiment which includes the choice of the appropriate sample type, chip fabrication, sample preparation and biochemical reaction; 2) data generation; 3) image processing; and 4) high level data analysis [Schena, 2000; Yang *et al.*, 2002].

2.4.2 Experiment Design

2.4.2.1 Choice of Sample Type

The first decision the researcher has to make when they design a specific experiment is to determine the type of the immobilized probes and the type of the sample. Since the gene expression profile is the most common application for microarray, cDNA is the most appropriate type while oligonucleotides is accounted as a second choice [Debouck & Goodfellow, 1999; Guo *et al.*, 1994; Leung & Cavalieri, 2003; Li *et al.*, 2004; Nuwaysir *et al.*, 2002]. On the other hand, when the target of the study is genes that share a large sequence identity such as genes that belong to the same family, oligonucleotides are a better choice because of their fine ability to distinguish between similar sequences [Baggerly *et al.*, 2004]. However, other probe types such as proteins and antibodies are becoming popular as researchers start realizing the potential applications of the microarray technology, which will allow proteome-wide screening of protein function in parallel [Glökler & Angenendt, 2003; Lueking *et al.*, 1999].

2.4.2.2 Chip Fabrication

To produce a microarray, pre-synthesized probes, usually PCR products, are immobilized on the array slide at a pre-defined grid location. The product of this approach is usually called "spotted microarray". Another method to produce the microarray is by using *in situ*, where each probe is individually synthesized on the surface of the slide [Debouck & Goodfellow, 1999; Yang *et al.*, 2002]. The product of this approach is usually called "Oligonucleotide microarray".

Common PCR primers can be used for the amplification of random sequences, which is very useful when there is no knowledge about the genome of the organism under study. Furthermore, microarrays can be used to perform a massive parallel sequencing [Diamandis, 2000; Zhou *et al.*, 2006]. For this purpose, the target to be sequenced is immobilized, then a very large set of short and labeled probes has to be hybridized with this target. Then, the examination of the pattern of hybridization and the computation of the original DNA sequence have to be done [Drmanac *et al.*, 1998]. Either that, or one should immobilize a very large set of short probes on a substrate and then hybridize these short probes with the labeled target of interest. Finally, biologists can infer the DNA sequence of interest by the analysis of the results [Pease *et al.*, 1994].

Other materials that can be used as probes are Oligonucleotides, which are prepared by conventional phosphoramidite pre-synthesis [Pon & Yu, 2005]. By following this approach, these oligonucleotides with their small size can minimize cross-hybridization that can possibly occur between distinct nucleic acids with sharing homology, from the same family. Given the existence of genes that share large sequence similarities in the genomes under study, oligonucleotides are often used in order to identify unique genes with greater specificity.

In the oligonucleotide microarray, *in situ* synthesis is merely a sequential addition of separate single nucleotides to a linker molecule directly onto the grid. This addition results in the synthesis of oligonucleotides, mostly with 20 nucleotides long.

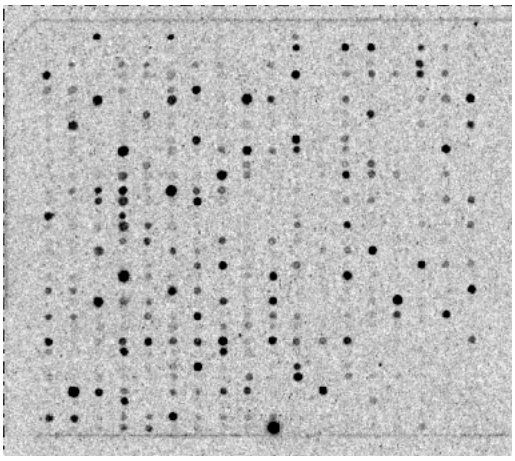


Figure 2.9: Nylon-slide Microarray.

I) Substrate: The substrates that might be used for sample spotting can be of various types such as glass (Figure 2.1), nylon membranes (Figure 2.9) or silicon chips (Figure 2.10). However, the glass slides are the most common ones because of their low cost, availability, resistant to high temperatures, low fluorescence and generally favorable chemical characteristics [Cheung *et al.*, 1999; Guo *et al.*, 1994; Moore *et al.*, 2002]

The attachment of the probes to the slide surface is based on the binding property of some chemicals. Generally, two substances are used to achieve this goal. The first material is the amine rich chemicals that give positive charges to the chip and interacts with the negatively charged probes. The second one is aldehyde chemistry where the 5' primer, which is used for generation of probes through PCR amplification of the desired sequence, carries an aliphatic-amine group which attaches it to the aldehyde coated slide [Lemieux *et al.*, 1998].

II) Types of microarray chips fabrication technologies: Three main types of advanced technologies are commonly used to prepare the microarray slide: *in situ*, mechanical spotting and the so-called ink-jet approach [Xiang & Chen, 2000; Yang *et al.*, 2002]. Each one has some advantages and some disadvantages. Thus, researchers have to choose the most suitable one of these technologies based on their own needs.

* ***In Situ*****:** In this method, oligonucleotides are built up base by base on the surface of the slide. The binding mechanism is based on the aldehyde chemistry. Usually, each nucleotide added to the oligonucleotide on the glass has a protective group to prevent the addition of more than one base at a time. The oligonucleotides in selected areas will be de-protected using chemicals or light in order to be ready before the next round of synthesis. This technology is very precise and highly automated since it allows direct fabrication of the chip using a sequence database without the need to add DNA clones, PCR products or other materials. Therefore, the risk of human error will be minimized.

Three main technologies are used for the de-protection in ”in-situ” synthesized array. Two are based on Photolithography, and the other is based on chemical de-protection:

1) Photodeprotection using masks: this is the basis of Affymetrix® technology [Brown & Botstein, 1999; Southern & Mir, 1999]. In this technology, specific masks are used in order to allow light to pass to some areas on the array but not to others. Each step of synthesis requires a different mask, and each mask is expensive to produce. This property makes this method expensive and time-consuming, and therefore limits the wider laboratory usage of photolithography. In fact, in situ synthesized microarray chips are currently produced only in commercial settings [Guo *et al.*, 1994]. However, once a mask set has been designed and made, it is straightforward to produce a large number of identical arrays.

2) Photodeprotection without masks: this method has been used by Nimblegen and Febit [Nuwaysir *et al.*, 2002]. In this method, the light is directed via micro-mirror arrays to affect the specific area which is needed to be de-protected, instead of using masks.

3) Chemical de-protection with synthesis via inkjet technology: in this method, the de-protection is based on a chemistry similar to the standard DNA synthesizer. The printing device works in a similar method to the conventional color printers but with the 4 DNA bases instead of the color [Allain *et al.*, 2001; Gershon, 2002; Singh-Gasson *et al.*, 1999]. Because of its flexibility, this technology has been adapted to produce microarrays with cDNA probes [Epstein *et al.*, 2002] in addition to oligonucleotides microarrays [Blanchard *et al.*, 1996].

* ***Spotted microarray*****:** In this methodology, a robotic spotting device deposits the pre-made probes on a specific location on the surface of the chip. In order to accomplish the printing task, the spotting robot contains a set of pins, which need to come close enough to the substrate to spot the probes. Those pins have to reload new samples after each deposit from a microtiter well plates.

Besides the low density of this type of microarrays, the high risk of the wrong manipulation of the pre-made probes is its main drawback. However, the low cost of the method makes it suitable for a wider range of laboratories, with limited budgets, and it is probably the predominantly used method for microarray chips' fabrication in the academic community.

* ***Ink jets*:** In a way it is similar to the spotted array methodology. The probes should be pre-prepared and then loaded into the miniature nozzle, which is controlled by a robotic system to ensure that the printing is at specific co-ordinates of all spots. After the printing of each spot, the nozzle is washed and loaded with the next sample of interest. One of the advantages of this technology is the un-necessity of the physical contact between the nozzle and the slide surface.

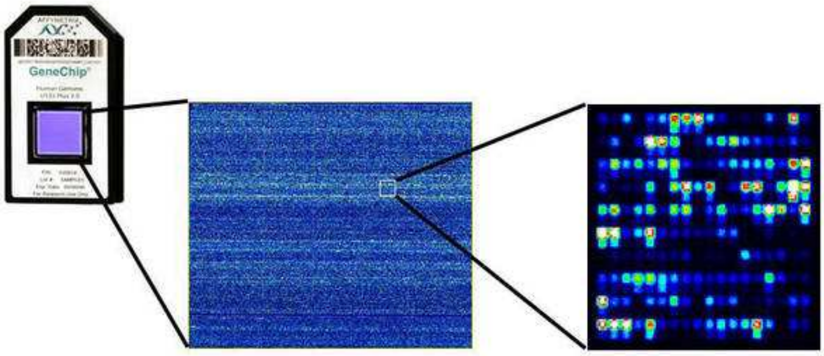


Figure 2.10: Oligonucleotides Microarray.

2.4.2.3 Target Preparation and Labeling

Many biological methods have been developed in order to prepare and label samples for microarray experiments. Basically, these methods depend on the biological query and the type of probes used in the experiment. Generally, the first step is to extract mRNA forming the cells or tissues of interest. These extracted mRNA will be used to synthesize cDNA. Most laboratories use fluorescent labeling with dyes of choice, usually Cyanine-3 (Cy_3, green) and Cyanine-5 (Cy_5, red). In the most common experiments, two samples are hybridized to the arrays. Molecules derived from the reference sample are labeled with one type of dye, say Cy_3, while the nucleic acid derived from the examined sample are labeled with a second type of dye, say Cy_5, and then the samples are mixed together. This allows the simultaneous measurement of both samples. When the resulted chip is scanned, up-regulated genes appear red (because of the high ratio Cy_5/Cy_3) while down-regulated genes appear green. Therefore, genes whose expression levels have not been affected appear yellow, as the red and green dyes are present in equal amounts.

There are generally three labeling approaches. The first is direct incorporation by reverse transcriptase; the fluorescent tags are immediately attached to the sample nucleic acid in a covalent manner. The second method is indirect labeling that also uses a reverse transcription reaction; a second molecule, commonly biotin, serves as an intermediate between the fluorescent tag and the probe. The third and the least common method for labeling is by random primed labeling using the Klenow fragment of DNA polymerase I. Many approaches have been developed to reduce the sufficient amount of RNA, which is required for an experiment to get a better quality labeling and, therefore, to improve the sensitivity of the detection process [Baggerly *et al.*, 2004].

2.4.2.4 Biochemical Reaction (Hybridization)

When the mixture of labeled samples (reference and testing) is ready, the next step is to expose the slide with the specified probes to these resulted materials. Biological bonding, which is called hybridization, is the stage in which the probes on the slide and the labeled samples form heteroduplexes bind together. Hybridization is a very complex mechanism, which is affected by many factors. The non-porous slides would increase the chance of two complementary molecules to come in physical contact in a certain amount of time. Furthermore, this type of substrate reduces the amount of samples needed for the experiment. In addition to the slide substrate properties, these conditions include temperature, humidity, salt concentrations, formamide concentration, volume of target solution and operator. The hybridization could be performed manually or, alternatively, robotic-ally by a hybridization station.

For instance, with the reference sample labeled with Cy_3 (green) and the tested sample labeled with $Cy5$ (red), the exposure of the chip to the mixture of both labeled samples will cause hybridization. The amount of red and green dyes present at any specific spot (corresponding to a particular gene) will depend on the way by which the applied treatment affects the expression level of this specific gene in the tested tissue [Brazma *et al.*, 2000].

For the genes that are not affected by the treatment, the amount of the corresponding mRNA is the same in both samples. Therefore, the red and green labeled molecules are of equal abundance and will present equal amount at these genes' assigned spots; the reflected color will be yellow. Besides, the genes that are over-expressed in the tested tissue would lead to a mixture which contains more red-labeled molecules representing those particular genes. Furthermore, the genes that are under-expressed in the tested tissue would produce a mixture with more green labeled molecules [Sherlock & Hernandez-Boussard, 2001].

Upon completion of the hybridization reaction, the array is subjected to wash in order to: 1) remove any excess hybridization solution from the array and ensure that the only labeled molecules on the array are the molecules that have specifically bound to the features on the array. 2) increase the stringency of the experiment by reducing cross-hybridization. After washing, the slide will be ready to be scanned.

2.4.3 Image Acquisition and Data Readout

At the end of the laboratory process, the image of the surface of the hybridized chip should be acquired. The heteroduplexes on the slide, where the target has hybridized to the probe, contain dye that fluoresces when excited by light of an appropriate wavelength. These dyes allow us to detect the amount of targets bound to each spot using several technologies. The most common one is the high resolution confocal laser scanners. The scanner contains one or more lasers that are focused onto the array; most scanners for two-color arrays use two lasers.

The software integrated with every scanner type extracts intensity signals from the chip and converts them into a numerical value; i.e., monochrome image. With the two-channel microarray, the output of the scanner is usually two monochrome images: one for each of the two lasers in the scanner. These images are combined to create the red (green) color images of microarrays. The images are usually stored as 16 bits Tagged Image File Format (TIFF) [Schermer, 1999]. This means that the intensity of each pixel in each channel is quantified as a 16-bit number, which takes values between 0 and 65535. Usually, background is approximately 100 and saturation can occur when the average pixel intensity is larger than 50,000. The microarray can detect intensities over an approximately 500-fold dynamic range.

The image resolution should be appropriate so that each feature has sufficient pixels to make the measurement of the intensity of the feature with the least possible noise.

The whole experimental procedures are highly important as the production of a good quality chip makes data analysis easier and substantially improves results. However, reading out and analyzing the result of a microarray experiment is probably the most difficult and the most challenging aspect of microarray experiments with vast latitudes for improvement. The size and the complexity of the generated data, in addition to the analysis of these data, form the most essential obstacle that the researchers' community faces, rather than the performance of the microarray experiment itself.

2.5 Data Generation, Processing and Analysis

Since the resulted image represents the raw data of the experiment, it is important to understand and improve the methods of extracting and analyzing the data of the image. Everything else, afterwards, is derived from those images and their initial analysis. The first step, then, would be converting the image into numerical measures that quantify the hybridization intensity of each channel for each gene or EST. This process is known as feature extraction. Typically, microarray image analysis programs give few summary statistics of the pixel intensities for each spot and for the surrounding background. The image processing involved has a major impact on the quality of data and its interpretation [Lehmussola *et al.*, 2006]. In general, there are four steps in image analysis process [Schena *et al.*, 1995; Yang *et al.*, 2002]:

- The filtering stage that is a low level image cleaning procedure. This stage can be used to remove the small contaminations or the background trend such as specks and dusts [Wit & McClure, 2004].
- The spotting stage, usually called gridding, which is used for the individual localization of spots' centers on the array.
- The segmentation stage which classifies pixels in a region immediately surrounding the gene as belonging to either the foreground or background domains.
- Feature extraction stage which is the process of analyzing every spot in order to determine the corresponding gene expression level.

2.5.1 Filtering

In this stage, the task is to identify the spots and distinguish them from spurious signals that can arise from either biological contaminations, such as a precipitated dye or other hybridization artefacts, or contaminants such as dust on the surface of the slide and other sources of nonspecific background.

Due to the existence of the high signal variability across the surface of the microarray, identifying gene spots can be counter-productive at this early stage. Therefore, rather than using the raw image, it is better to apply some filters that produce an image data such that emphasis is placed on certain frequencies or regions of interest. These results are more appropriate than the original image for the specific applications at hand. The importance of stressing "specific" is to highlight the important features of the microarray, which should be considered for designing and applying any filter during this stage. Irrespective of the applied filter, however, filtering is one of the most interesting areas of microarray image processing.

In general, regardless of the type of analyzed data, it is a normal practice to pre-process the data such that artefacts, of one type or another, are reduced. Since the noise objects could be incorrectly interpreted as valid data points, filtering stages are designed to reduce such interpretation issues. A microarray image is littered with various amounts of noise that could cause the analysis algorithm to make wrong decisions. Accordingly, a proper filtering technique must lead to a remarkable removal of noise artefacts in the microarray image surface.

Although microarray technology has been engineered to fine tolerances, there exist problems regarding contamination which can corrupt the required gene signal. The most obvious reason that causes noise is the biological processes. Since biology is based on biochemical interactions, any mutation or corruption of a single component within a sequence interaction can cause a significant effect on the final hybridization results. In general, these types of contaminations can be divided into two categories: hybridization related and washing related contaminations. Many factors affect the hybridization process including temperature, humidity, salt and formamide concentration as well as the target volume solution. The existence of the genes' families, which share sequence similarity, may lead to cross-hybridization during the hybridization process. This cross-hybridization produces misleading signal on the surface of the microarray slide.

Usually, upon the hybridization completion, the slides should be washed in order to remove these weakly bound probes and all other solutions, which have been used, from the microarray surface thus increasing the stringency of the resultant data. However, with the poor protocols of the washing stage, the washing material itself may drie onto the slide surface. These materials lead to an erroneous signal on the slide surface.

Another type of noise that can enter the process is the systematic noise. This type of noise typically has a form of internal order or structure. In the case of microarrays, it is usually a result of poor slide preparation and hardware equipment such as printing pins. The operator and the manual handling of the slide during the experiment may cause some sorts of artefacts, such as hair, scratches, dust and finger prints.

Filtering could be defined as a replacement of each pixel in the image with a value derived from the pixel and the other pixels surrounding it. It changes the dynamic range of the image, produces a smoother image and removes local noise or interference in an image. Two types of filter are highlighted in the literature, the median filter and the top-hat filter. The former allows for the removal of small contaminations which affect only a small number of pixels, and the latter allows for the robust removal of background on an array.

2.5.1.1 Median Filter

A median filter simply replaces each pixel with the median value among the pixels in a window centered on the pixel. Generally, the window would be of small dimensions [Glasbey, 2001], such as 3×3 or 5×5, for two reasons: 1) Removing small artefacts on the array so that they are fully attenuated or are less noticeable. 2) Since a hybridized spot size typically is much larger than the size of the smoothing window, the filter will not affect the overall structure of the spot.

2.5.1.2 Top-hat Filter

The top-hat filter [Buckley, 2000] is a well-known filter for estimating the trend of the image. Mainly, the top-hat filter performs morphological opening. By subtracting this trend from the image, the background contamination will be reduced [Glasbey, 2001; Yang *et al.*, 2002]. This filter, first, replaces each pixel by the minimum value of a square window centered around it. Then, it replaces each pixel of the resulted image with the maximum value in the window. Using a window of size bigger than the spot size, only the local background will be estimated; all spots will disappear in the morphological opening image. Finally, the top-hat filter subtracts the morphological opening from the original image. The important property of this filter is the preserving of non-negative estimation of the gene expression [Glasbey, 2001; Yang *et al.*, 2001]. Therefore, this filter effectively solves the brightness differences of the background which makes comparison of similar features in different parts of an image difficult.

2.5.1.3 Anisotropic Diffusion

Generally, the image has structures at different scales. In practice, however, it is not straightforward to specify the right scale for any particular application. Thus, multiple scale representation for the image is totally advantageous [Alvarez *et al.*, 1993]. A multi-scale representation of an image is an ordered set of derived images intended to represent the original image at various levels of scale [Bovik, 2000]. A proper description of these structures will lead to a smooth image processing in later stages.

Towards this end, Perona & Malik [1990] proposed anisotropic diffusion for adaptive smoothing in order to formulate the problem in terms of the non-linear heat equation. Applying anisotropic diffusion in the filtering stage of the microarray image is beneficial since it preserves edges and simultaneously represents a good approach to achieve smoothing [Weickert, 1998].

2.5.2 Gridding

In this stage, the position of every spot center on the microarray is specified. The image of the microarray has a regular structure (see Figure 2.1) since the spots are located on a uniform grid [Lonardi & Yu, 2004] with larger space between sub-grid than between the spots within each grid. In cDNA spotted microarrays, the spots are usually arranged in sub-groups representing each of the pen tips used to deposit the probe. In general, arrays have their spots arranged in a rectangular grid.

However, real microarray is rarely close to the ideal desired image. In fact, microarray has variations on the spot position, irregularities on the spot shape and size, contamination and global problems that affect multiple spots. Many issues might arise across the array, such as uneven grid positions, curve within the grid, uneven spot spacing and uneven spot size [Stekel, 2003].

Although almost all the software packages have automatic gridding algorithms, they do not produce error-free results. Practically, the manual supervision and intervention are commonly required either on all the process or, at least on a part of it. In general, an assumed grid is placed over the image with a little manual intervention to achieve the gridding task. That is, the software tries to adjust the provided fixed grid and then allows the user to tune the result. The output of this stage will be used in later stages, the segmentation and the quantification.

The importance of the grid placement is due to the usability of the grid coordinates for identifying the individual array spots and assigning identities to them. Shifting or misaligning the grid may lead to the assignment of particular expression levels to the wrong genes.

2.5.3 Segmentation

Once grids have been placed, discrimination between areas that are considered the spot signal and areas that are considered the background signal must be carried out. The process, by which each individual cell in the grid must be selected to determine the spot signal and to estimate the background hybridization, is called segmentation. That information will be put towards a quantitative measurement at each cell. Different software packages, academic and commercial, employ different segmentation algorithms. In general, the available techniques can be categorized into three main classes. These techniques could be manual, semi-automated or complex automated methods. Practically, these methods are time consuming and/or computationally expensive [Lukac *et al.*, 2004]. Regarding the segmentation process, two issues should be highlighted: 1) It has been anticipated that the spot intensity value would be independent of the segmentation algorithm if a background correction method is utilized [Yang *et al.*, 2002]. However, Lehmussola *et al.* [2006] showed that the segmentation method significantly influences the identification of differentially expressed genes. 2) The large number of spots and the irregularities of the spots' shape and position have led to the recent interest in using a fully automated procedure to accomplish the task [Bajcsy, 2004; Jain *et al.*, 2002; Katzer *et al.*, 2003].

There are four widely used approaches for segmentation:

2.5.3.1 Fixed Circle Segmentation

In this approach, a circle of fixed size is placed over the center of mass of the spot. All the pixels inside this circle are used as those that form the signal,i.e., the intensity of the spot. The area outside the circle is considered as a background associated with this spot.

This method is computationally simple and provides a reasonable estimate of the required measures about the spot. With high-quality microarrays, where the size and the shape of spots are highly consistent, the fixed circle segmentation is the best approach. However, since the microarray usually is far from ideal it can lead to mis-estimation of both the spot signal and the local background signal, in particular, when the variability in spot size and position exists [Yang *et al.*, 2001].

Fixed circle segmentation is the most commonly used method in cDNA image analysis packages including ScanAlyze [Eisen, 2010], GenePix [Anonymous, 1999] and QuantArray [Lumonics, 1999].

2.5.3.2 Adaptive Circle Segmentation

This approach is similar to the previous one but with a variable diameter of the circle. Therefore, it is able to deal with variability of the spot size, but it does not perform well with irregular spot shape .

Some packages are equipped with adaptive circle segmentation methodology such as QuantArray, GenePix and Dapple [Buhler *et al.*, 2000]. Usually, manual tuning of the diameter of the circle is used. However, The second derivative of the image is used to specify the diameter such as in Dapple package.

2.5.3.3 Histogram and Threshold Segmentation

In this approach, the histogram of all the pixels in the cell, spot and background, is produced. Typically, both extremes, highest and lowest pixels, should be omitted. With an ideal image, this will produce a bimodal distribution of pixel values, i.e., the higher mode corresponding to the spot and the lower mode corresponding to the background. This old technique is simpler than the other methods with an ability to produce reliable results for spots that shaped irregularly.Therefore, the underlying assumption of this method is that the peaks in the histogram can be utilized in order to specify a threshold for the discrimination between the gene spot and the background regions. Unfortunately, this assumption can lead to incorrect observations. For example, in noisy images there could be no peaks (valleys) that can be used to infer a threshold as the range of intensities is very small. Thus, it is almost impossible to find a threshold value or a set of threshold values that will result in a single connected region matching the set of spot pixels that a biologist would determine to be the spot pixels. Furthermore, if the spot's area is very small comparing to the cell area the histogram method will produce an unreliable result.

On the other hand, ignoring the spatial relation between pixels makes histogram segmentation prone to the contamination in the background. This can lead to a false discrimination by considering this contaminating signal as a part of the spot's signal. Note that ImaGene [Anonymous, 2008] and Quantarray have the option of using histogram segmentation.

2.5.3.4 Adaptive Shape Segmentation

As a flexible method, adaptive shape segmentation attempts to, precisely, identify the spot pixels by including only those falling within a tile boundary. The Seeded Region Growing (SRG) algorithm is the most common one among these methods [Adams & Bischof, 1994]. SRG method is perhaps the most powerful one with respect to shape identification since it allows us more precisely to identify those pixels representing real hybridization. Therefore, it can provide a better estimate of the actual intensity associated with each spot.

In the SRG methodology, seeds corresponding to foreground and background signals should be specified. The spot is then grown from its seed pixels by deciding whether adjacent pixels belong to the spot. The same is done for the background signal until all the pixels are assigned to the background or to the spot.

However, SRG is computationally difficult, causing the image processing time to increase. It relies on seed growing methodology as well as the selection of the initial seeds [Tran *et al.*, 2004]. Thus, depending on this particular algorithm, it may possibly lead to misidentification of the spot and the surrounding background areas.

Watershed segmentation is another adaptive approach to specify the spot's pixels based on mathematical morphology [Siddiqui *et al.*, 2002]. It was developed in academically developed software. Angulo & Serra [2003] compared it to GenePix and ScanAlyze. However, the watershed algorithm application to microarray image depends on the spots and background intensities distribution. In addition, watershed approach is sensitive to the noise objects. Traditional watershed algorithms have been described as excessively slow but there has been a lot of works toward improving their efficiency [Vincent, 1993].

2.5.4 Segmentation: Current State-of-the-art

To overcome the limitations of the techniques mentioned above, many approaches have been proposed. Cheriet *et al.* [1998] developed Otsu's method [Otsu, 1979] by proposing a general recursive approach for image segmentation based on discriminant analysis. In this approach, the image's histogram is calculated for every iteration with the largest peak being discriminated from the others. The process will stop when there are no more peaks in the histogram. However, the method produces unreliable results if there are more than two classes within the data. This method produces results only when the desired object corresponds to the global minimum and this object is the darkest object, otherwise it fails [Fraser *et al.*, 2010]. A gradient relaxation algorithm is proposed by Parvin & Bhanu [1983] to solve unimodal problems and compare it to a non-linear probabilistic relaxation algorithm [Ranade & Rosenfeld, 1980]. This relaxation process is iterative and tends to change the histogram properties by altering the values of the pixels in order to detect the threshold value more appropriately.

Edge based algorithms depend on any significant change in intensity of the image, which is usually spatially localized [Marr & Hildreth, 1980]. However, due to real image characteristics, these changes are not typically abrupt. In [Ahuja *et al.*, 1980] a pixel neighborhood method for image segmentation has been used where each pixels' neighbor was identified in a specific window size. This information forms a good set of features to classify the pixels. Perkins [1980] addressed the gap problem resulting from undetected edge and he proposed an expansion/contraction technique to solve it.

Marr & Hildreth [1980] utilized the second derivative of the Gaussian to detect the edge. Perona & Malik [1990] developed the work of Witkin [1984] by replacing the scale-space analysis using isotropic diffusion with the one using anisotropic diffusion. In their method, the diffusion coefficient is chosen to vary spatially in such a way as to the real edge sharp. The selection of gradient threshold in the coefficient function is essential. Pixel-level-Snakes techniques have also been applied [Ho & Hwang, 2007; Srinark & Kambhamettu, 2004]. However, these methods are sensitive to the noise degree in the image.

Although SRG algorithm is fairly robust, it has an obvious problem due to its dependency on the pixel ordering process. That is, the result of the process taking left-to-right order is different from the result of processing the opposite direction. To solve this problem, Mehnert & Jackway [1997] improved the SRG algorithm by making it independent of pixel ordering. Their algorithm has the advantages of the original SRG, but if any conflict occurs the process tries to solve it after all other pixels are labeled. The problem with this method is that the remaining unresolved conflict requires the intervention of the human to re-examine it.

Clustering is an unsupervised technique used for image segmentation that allows the discrimination between pixels and forms groups of pixels with similar intensity values. Arguably, k-means [MacQueen, 1967] and fuzzy c-means [Dunn, 1973] are the most common approaches. K-means creates k random cluster centers that correspond to k different partitions. Each data point is then assigned to the nearest cluster center which is recalculated using its current members. The process is of iterative nature, and the stopping condition can be such as no more reassignment of data points or a minimal decrease in squared error.

With the fuzzy logic based algorithms on the other hand, instead of assigning the data point in a crisp fashion to a specific group, every pixel will be assigned to every group with a specific membership value between $[0, 1]$, which describes the degree of adherence to a particular cluster. Therefore, fuzzy logic based clustering offers inherent advantages over non-fuzzy methods; as they cope better with the space problem which does not have well defined boundaries.

However, the main disadvantages of these clustering techniques are: 1) they are heavily influenced by initial starting conditions; 2) they can become trapped in local minima.

A clustering of the full microarray image area in one step has been proposed [Bozinov & Rahnenfuhrer, 2002], but this might not be computationally feasible with current processing power. To overcome such a computational issue, the authors in [Bozinov, 2003] proposed an abstraction of the k-means clustering technique. Furthermore, Bayesian approach has been applied in [Lawrence *et al.*, 2003]. Gaussian Mixture Model (GMM), utilized in a fully automated framework, was presented in [Blekas *et al.*, 2003, 2005]. Yet, all these methods ignore the spatial dependencies among adjacent pixels. More recently, Fraser *et al.* [2004] presented a copasetic analysis framework that attempted to improve the full workflow processing of microarray image analysis employing traditional clustering techniques. Other methods such as the applications of wavelets [Noda *et al.*, 2002; Wang *et al.*, 2003a] and Markov random fields [Demirkaya *et al.*, 2005; Katzer *et al.*, 2003; Li, 1994] showed great promise. Lukac *et al.* [2004] utilized the multi-channel nature of the cDNA image data. Particularly, Arena *et al.* [2002a] showed the potential of CNNs in the cDNA microarray image's analysis. The CNNs' parallelism characteristic makes them an ideal computational platform for kernel-based algorithms and image processing [Chua, 1997].

2.5.5 Background Separation

Usually, during the hybridization process some non-specific probes and other fluorescence from the glass contaminate the surface of the slide. This effect should be considered in some way in the later analysis. By accounting that the background signal is a good estimate to these contaminations, subtracting the local background signal from the foreground signal will yield a more reliable value for the hybridization intensity of each spot. However, there is no standard approach to specify an appropriate background area. Figure 2.11 highlights three most common background separation methods, which are utilized by several commercial or academic packages.

GenePix's approach is the most popular one, see Figure 2.11(1). For every spot in this method, four diamond shapes located at the valleys between the spots are considered together the background region for the center spot. The pixels of the spot area and the pixels of the background area represent the intensity of the foreground and background group respectively. Considering the position of the diamonds, one can note that these regions do not represent the real background, and then they may lead to a wrong measurement.

In Figure 2.11(2), the circular background method, such as in ImaGene package, attempts to eliminate the mentioned problem. In this approach, the background region is closer to the spot area but, on the other hand, determining the gap between the spot and the background regions has a considerable effect on the later measurement, due to the fact that the intensity of the boarder pixels of the spot represents genetic material other than the gene spot material. Therefore, including these pixels in the background estimation will yield incorrect high background signal. On the other hand, the large gap will produce either the same results as those of the GenePix or will include pixels from the neighboring spots.

When applying adaptive approach to achieve the segmentation step, the most appropriate method to represent the background region is to consider every pixel in the cell as a background one except those belong to the spot itself, see Figure 2.11(3).

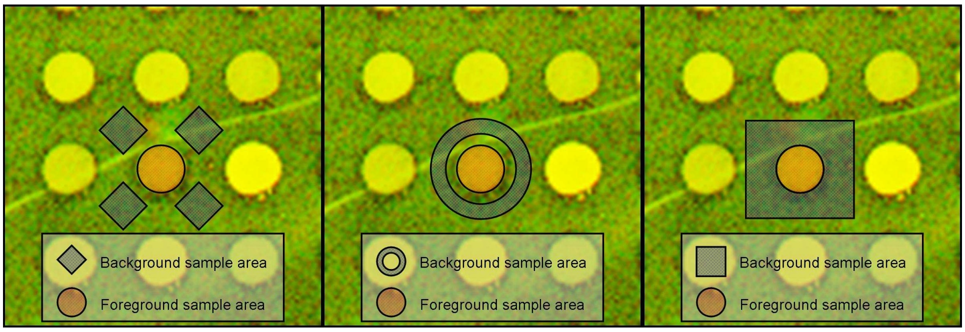


Figure 2.11: Gene Spot Background Region as Used by Common Packages: a) GenePix; b) ImaGene; c) ScanAlyze

2.5.6 Gene Quantification

Having obtained the pixels representing the hybridization region of every spot and its background region, suitable summary statistic measurements need to be calculated. The extraction of these measurements is called Quantification. Typically, this information includes, for each spot and its background: 1) the mean value; 2) the median value; 3) standard deviation; 4) diameter; 5) the number of each spots' pixels.

The rationale behind using median (mean) is that the level of fluorescence is directly proportional to the amount of hybridizations for that gene and, therefore, to the amount of RNA produced by the gene [Smyth *et al.*, 2003]. The standard deviation is a quality measure of the hybridization. This information will be the input for the next stages of data analysis.

2.5.7 Microarray Data Analysis

Having these statistical summaries computed, two points should be considered, 1) the large amount of data produced even from single experiment, and 2) the fact that the microarray measurements provide a rough estimate of the relative relation between two conditions per gene and on average over a possibly large population of cells. Therefore, some software packages should be utilized in order to get meaningful inferences from the data.

Several tools have been developed to carry out different tasks such as clustering, predicting and visualizing patterns in a high-dimensional space. The most common methods applied to microarray data are correlation based approaches. The applied analysis on a large data provides new insights about the situation under study. For instance, comparing expression profiles leads to a better understanding for molecular pathogenesis of a variety of diseases, which forms a step towards achieving a true understanding of genome function. Statistical analyzes, such as clustering and class prediction, are typical methods currently used in gene expression data analysis. Some goals of the analysis are: improving the data, finding regulated genes and representing the data in a human readable way.

2.5.8 Microarray Image Reconstruction

Regardless of the microarray image analysis methodologies, all of them deal with the same set of knowledge, for example, the position of the spot and the way to discriminate between gene signal and background signal (either edge or mask). Then, the median of the gene spot and the median of the background pixels are taken to be foreground and background intensities. Assuming that there is a little variation within the gene and the background regions, the background median is subtracted from the foreground and the result is summarized as a log2 ratio.Unfortunately, this is not always the case. A good example of the low-level signal produced in the image can be seen in Figure 2.11. The image may have many problems such as the missing or partial gene spots, shape inconsistencies and background variation, i.e, the scratch and the variation of the background illuminations around the presented genes. Therefore, several statistic techniques are employed to estimate the microarray background [Bengtsson & Bengtsson, 2006].

However, what is needed is a more specific background determination process that can account for the inherent variation between the gene and background regions. One of the first techniques applied specifically to reconstruct microarray images is the proposal of O'Neill *et al.* [2003]. In particular, the gene area is replaced by selecting pixels, which are most similar to the known border, from a known background region. The underlying assumption is that the similarity with the given border intensities guarantees the transition of the local background structures through the new region. Fraser *et al.* [2007] applied Fast Fourier Transform (FFT) on two lists representing the foreground and background regions in a specified window centered at the target spot. Then, a minimization function of the real part of the transform has been used in order to retain the subtle intensity information within the background region and to allow the gene spot area to inherit it. To produce the reconstructed area a reversed FFT should be applied on this output. Most recently, graph-Cut Image Reconstruction (SCIR) has been proposed in [Fraser *et al.*, 2008]. Particularly, this technique creates a chain of pixels through the area that has a maximal(minimal) intensity. Therefore, the algorithm replaces the high-contrast pixels (edge) within the gene spot area with the low-contrast pixels within the local background area.

2.6 Testing Dataset

The images used in this thesis are derived from the human gen1 clone set data. These experiments are designed to contrast the effects of two cancer inhibiting drugs (PolyIC and LPS) over two different cell lines. One cell line represents the control (untreated) sample, and the other represents the treated (HeLa) line over a series of several time points. In total, there are 47 distinct slides with the corresponding GenePix results presented. The size of each slide is approximately (2000×5000) pixels. Each slide consists of 24 gene blocks, with each block containing 32 columns and 12 rows of gene spots. The gene spots in the first row of each odd-numbered block are known as the Lucidea ScoreCard [Samartzidou *et al.*, 2001] and consist of a set of 32 pre-defined genes that can be used to test various experiment characteristics. The remaining 11 rows of the odd-numbered blocks contain the human genes themselves. The even-numbered blocks are duplicates of their odd-numbered counterparts. This means that each slide has 24 repeats of the 32 ScoreCard genes. Note that it is generally accepted that extreme pixel values should be ignored as these values could go beyond the scanning hardware's capabilities.

Chapter 3

A Novel Neural Network Approach for Microarray Image Segmentation

The discussion in this chapter based on:

Zineddin, B., Wang, Z. & Liu, X., 2010. A Novel Neural Network Approach To cDNA Microarray Image Segmentation. Pattern Recognition Letters, submitted.

> *If you just have a single problem to solve, then fine, go ahead and use a neural network. But if you want to do science and understand how to choose architectures, or how to go to a new problem, you have to understand what different architectures can and cannot do.*
>
> Marvin Minsky

Microarray technology has become a great source of information for biologists to understand the work of DNA, which is one of the most complex codes in nature. Microarray images typically contain several thousands of small spots, which represent different genes in the experiment. One of the key steps in extracting information from a microarray image is the segmentation whose aim is to identify which pixels within an image represent which gene. This task is greatly complicated by the noise within the image and the wide degree of variation in the values of the pixels belonging to a typical spot. In the past there have been many methods proposed for the segmentation of microarray images. In this chapter, a new method utilizing a series of artificial neural networks, which are based on multi-layer perceptron (MLP) and Kohonen networks, is proposed. The proposed method is applied to a set of real-world cDNA images. Quantitative comparisons between the proposed method and the commercial software GenePix are carried out in terms of the peak signal-to-noise ratio (PSNR). This method is shown to not only deliver results comparable and even superior to existing techniques but also have a faster run time.

3.1 Classification

Generally, there are two types of classification methods, namely, unsupervised learning and supervised learning. The former, also called clustering, establishes a number of clusters in a set of observations. The later, called classification in this thesis, aims to derive some rules based on a set of pre-classified data points. These rules will be used later to assign any new observation into one of these classes. In this chapter, both methods will be used to deal with different issues through the stages of microarray image analysis.

The rationale behind the application of classification procedures is fundamentally based on the intrinsic characteristics of microarray image as well as the general consideration related to the array production process, see Section 2.5. The crucial impact of segmentation methodology on all subsequent analysis is an important reason to ensure high standards for implementing classification approaches. Furthermore, the imperfection of the spots' quality and the huge number of spots in the image justify the tendency to develop automatic classification procedures.

There are some caveats that must be made about many issues relating to any algorithm proposed for microarray analysis. First, there should be a measure in which one would be able to specify the accuracy of the results. Every investigated technique produces a mask that classifies the pixels as belonging either to signal (the gene spot) or to noise (the local background). The reliability of the segmentation algorithm is usually represented by the percentage of correct classified pixels. However, the lack of a golden standard that defines precisely the ideal mask necessitates the development of some comparison methods; these methods are the only way to qualify the performance of any algorithm. Two general methods, subjective and objective, have been utilized in this thesis toward this goal.

The evaluation of the proposed algorithm is carried out using the dataset that highlighted in Section 2.6. The dataset will be divided into two groups, namely, the training set and the testing set. Note that the performance over the training set especially in techniques such as neural networks is usually different from the performance over the testing set. Therefore, the evaluation using the unseen data is of practical importance. For this purpose, the dataset has been divided into two groups. The first group is a randomly selected training set, and the remaining data is the testing set. The randomness of the training set selection guarantees the unbiased estimate of the accuracy.

On the other hand, in order to have a better estimate of the performance of the proposed classifier, the time issue should be considered. The time factor can be divided into two components: the training time and the speed of the classifier when applied on the testing data (or the real application of the classifier). In the microarray analysis, the time is not as essential as a real time application. However, the time for analysis should be practical yet the results are, as has been mentioned before, appropriately accurate.

Another issue is the comprehensibility of the decision of the applied tool. Neural networks are well known for their ability to model nonlinear functions; i.e., classification rules in our case. However, they use no-parametric approach. Therefore, the model obtained with neural network (black box) is not comprehensible in terms of physical parameters. To overcome this lack of understanding, analysis of the proposed system has been carried out to gain a general explanation to the results of this approach.

3.2 Neural Networks

A first interest in the ANNs [Bishop, 1995; Haykin, 1999; Rumelhart *et al.*, 1986], as parallel distributed processing, has emerged after the publication of the research of McCulloch &

Pitts [1943] about a simple neuron. McCulloch-Pitts's neuron was a model of the biological neuron as well as a conceptual component for circuits that carry out computational tasks. These neurons represented the pursuit of researchers, both in academia and industry, to mimic human abilities such as the speech and the use of language. Recent discoveries showed that the utilized models of the artificial neural networks seem to introduce over simplification of the biological models [Bower & Beeman, 1998]. Yet, this approach has been dedicated to solve many practical issues such as pattern recognition [Ripley, 2008], modeling [Polycarpou & Ioannou, 1992], and prediction [Ozkaya *et al.*, 2007].

In general, artificial neural network consists of layers of interconnected processing units called "cells" or "neurons". Each neuron has a nonlinear function, called activation function, that determines the output of the neuron based on its input. The input to a neuron may come from the input data and/or other neurons. The output of the network is identified by the output of a specified layer; a set of specified neurons which form the output layer. The network represents a very complex set of interdependencies, which can model any degree of non-linearity. Therefore, it allows approximation of very general functions at least in theory,.

In the simplest networks, the connections between the units allow propagating messages through the layers. Generally, each connection is defined by a weight that specifies the effect from each neuron on a neuron at the other end of the connection. The network becomes a recurrent neural network when the output units are connected with the earlier units. The recurrent network achieves much more complex behavior. Therefore, it is able to model a highly nonlinear system with feedback. The weights of the connections are determined usually by a combination of some statistical techniques with the machine learning techniques; this process is called learning. However, the learning process is achieved in a way that lacks the ability to make the learned rules transparent to the user.

3.2.1 Artificial Neurons

The artificial neuron is the information processing unite in the network. The neuron, basically, receives the input signal and uses it to produce the output signal which will propagate to other neurons, or will represent the network output. Generally, the neurons can be divided into three general types: input layer neurons which receive data from outside the system; hidden layer neurons which interact only with other neurons inside the system; and output layer neurons which send out the final output of the system. It should be mentioned that the whole system of neurons is totally parallel in the way that many neurons can carry out their computations simultaneously.

Figure 3.1 shows the model of a neuron. There are four basic components of the model:

1) Input Vector is multiplied by the synaptic weights' vector and connected to the neuron. The input values are a vector $\mathbf{x} \in \Re^{\mathbf{n}\times\mathbf{1}}$, with individual components given as x_i, $i = 1, \ldots, n$. Therefore, every component x_i is input to the ith synapse and connected to a neuron j through a synaptic weight w_{ji}.

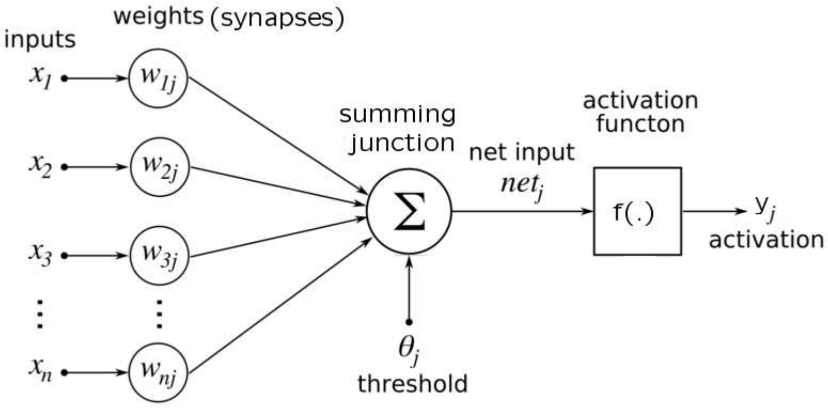


Figure 3.1: The model of an artificial neuron

2) The summing junction $\sum$ acts to add all the signals fed by the synapses. Thus, every input is multiplied by a synaptic weight and then summed. net_j is a linear combination of the input to the synapses.

3) The activation function $f(.)$ produces the final output of the neuron y_j. Usually, the activation function is a nonlinear function, but it can be a binary or a bipolar. The non-linearity enhances the ability of the network to achieve a desired outcome, such as classification and approximation.

4) The threshold, or bias, θ_j is an external signal, usually with a fixed value '-1'(or '1'). The threshold lowers the cumulative input to the activation function. Therefore, θ is subtracted from the output of the linear combination net_j before the activation is applied.

Therefore, the total input to neuron j is simply the weighted sum of the separate outputs from each of the connected neurons and the bias or threshold term θ_j:

$$net_j(\mathbf{x}) = \sum_{i=1}^{n} w_{ji}x_i + \theta_j, \tag{3.1}$$

where θ is a bias. The contribution for positive w_{ji} is considered as an excitation and for negative w_{ji} as an inhibition.

3.2.2 Activation Function

There are several different types of activation functions. Each one gives a better performance for different problems that neuron has to solve. We can define some of the most used functions as follows:

I) The Linear Function:

Identity function, as in regression problems, is a good example of this type. This function, as shown in Figure 3.2, is a continuous-valued and its output at j neuron is:

$$y_j = f(net_j) = net_j \tag{3.2}$$

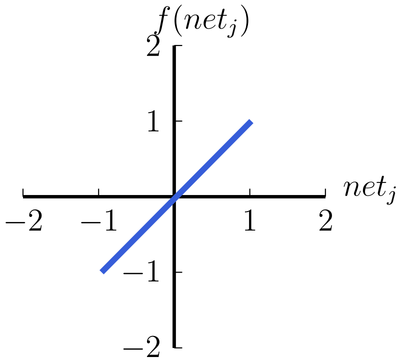


Figure 3.2: Linear activation function

II) The Hard Limiter Function:

This type can be a bipolar (or binary) function, as shown in Figure 3.3, that hard-limits the linear combination of the summing junction to a '−1' or a '1' for the bipolar function (or a '0' or a '1' for the binary function). The output of the binary hard-limiter for j neuron can be written as:

$$y_j = f(net_j) = \begin{cases} 0 & \text{if } net_j < 0 \\ 1 & \text{if } net_j \geq 0 \end{cases} \tag{3.3}$$

For the bipolar hard limiter (see Figure 3.3), the j neuron output is written as:

$$y_j = f(net_j) = \begin{cases} -1 & \text{if } net_j < 0 \\ 0 & \text{if } net_j = 0 \\ 1 & \text{if } net_j > 0 \end{cases} \tag{3.4}$$

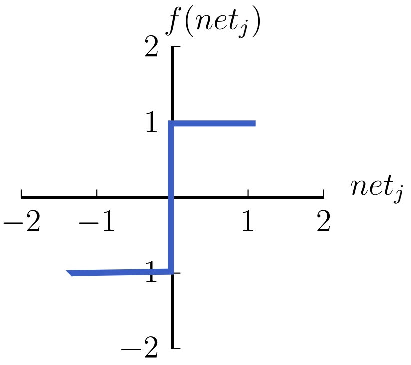


Figure 3.3: Symmetric hard limiter activation function

Usually, The neuron with hard limiter activation function is called McCulloch-Pitts model [McCulloch & Pitts, 1943]. In this model, there is no learning process and the weights are derived from analysis.

III) Symmetric Saturating Linear Function:

This function has two region types, see Figure 3.4. The first one is the saturating regions which have bipolar output; either '1' or '-1'. The other one is the linear region. The output of the j neuron can be written as:

$$y_j = f(net_j) = \begin{cases} -1 & \text{if } net_j < 0 \\ net_j & \text{if } -1 \leq net_j \leq 0 \\ 1 & \text{if } net_j > 0 \end{cases} \tag{3.5}$$

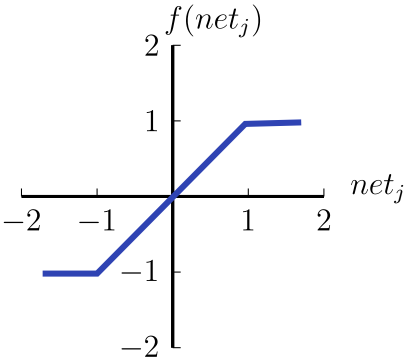


Figure 3.4: Symmetric saturating linear activation function

IV) Sigmoid Function:

The nonlinear sigmoid function is the most common activation function. Its characteristics, continuity and differentiability, guarantee the desired performance for a wide range of applications. At this stage, it should be mentioned that the derivative of the activation function plays a crucial role in the learning process of the neural network. In the binary sigmoid function (see Figure 3.5), the saturating output has a binary range. The binary sigmoid function for j neuron is:

$$y_j = f(net_j) = \frac{1}{1 + e^{-\alpha net_j}} \tag{3.6}$$

where α is the slop parameter of the function. This parameter affects the shape of the sigmoid function.

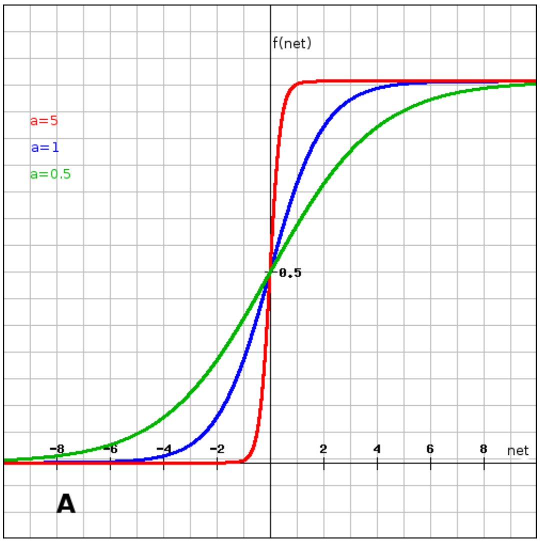


Figure 3.5: Binary sigmoid activation function.

Another sigmoidal function is the hyperbolic tangent sigmoid (see Figure 3.6), which has a bipolar form. In this function, the saturating limits have a bipolar range. The output of the j neuron can be written as:

$$y_j = f(net_j) = \frac{1 - e^{-2\alpha net_j}}{1 + e^{-2\alpha net_j}} \tag{3.7}$$

where α is the slope value. The hyperbolic tangent and the binary sigmoid functions are equivalent. With basic linear transformations to inputs and outputs, each one of these functions can be transferred into the other. Practically, the hyperbolic activation functions achieve faster convergence of training algorithms than the binary sigmoid functions.

The derivative of the binary sigmoid function with respect to the output of the summing unit can be written as in (3.8). And the derivative of hyperbolic tangent function with respect to the output of the summing junction can be written as in (3.9). Observing these

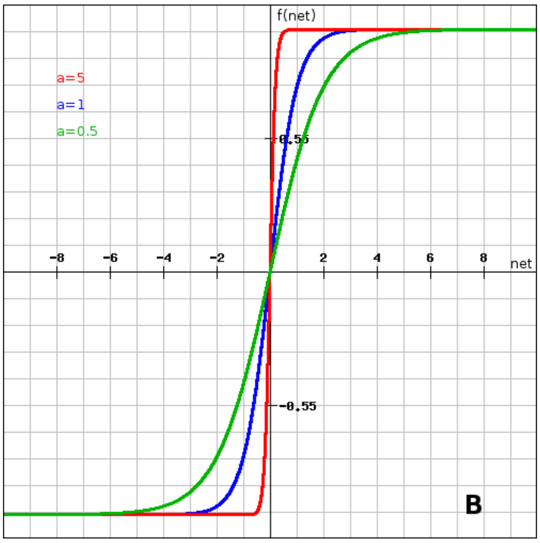


Figure 3.6: Hyperbolic tangent sigmoid function.

derivatives, Figure 3.7 shows that the main advantage of the sigmoidal activation functions is the derivative dependency only on the activation function output. This characteristic is of crucial importance for the training algorithms as we will see later.

$$g_j(net_j) = \alpha f(net_j)[1 - f(net_j)] \tag{3.8}$$

$$g_j(net_j) = \alpha[1 + f(net_j)][1 - f(net_j)] \tag{3.9}$$

By using the linear activation function, (3.1) has the following simple geometrical interpretation [Duda & Hart, 1973] (as cited in [Bishop, 1995]). Let's assume that our discussion is based on 2-dimensional input space. The decision boundary $net_j(\mathbf{x}) = 0$ is a straight line, see Figure 3.8. The vector $\mathbf{w}$ is normal to the decision line, and also $\mathbf{w}$ determines the orientation of the decision boundary. In Figure 3.8, the bias θ_j determines the position of the decision line on the x-Axe.

This interpretation can be generalized in order to cover a space of any dimension. Sigmoidal functions, on the other hand, represent a generalization form of the linear function; when $|\alpha|$ is too small, in (3.6) (or (3.7)), the sigmoidal function can be approximated by a linear function. However, it is advantageous to interpret the output of the sigmoid function as posterior probabilities [Bishop, 1995]. Therefore, the neural network will give more than a simple classification decision.

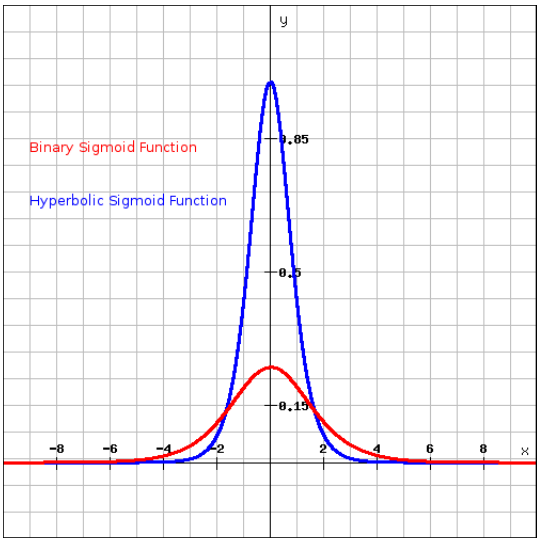


Figure 3.7: Derivatives of hyperbolic tangent and binary sigmoid functions.

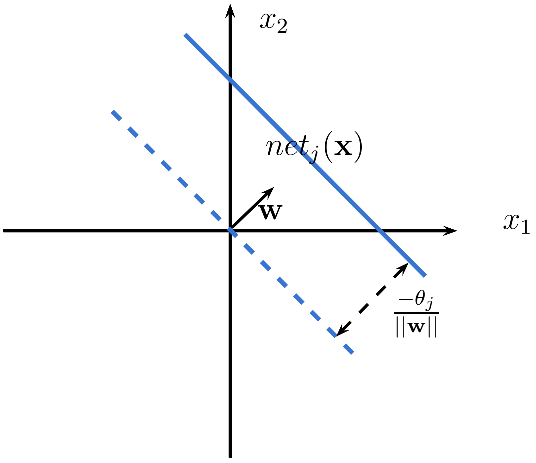


Figure 3.8: A linear decision boundary, corresponding to $net_j(\mathbf{x}) = 0$, in a 2-dimensional input space (x_1, x_2). The weight vector $\mathbf{w}$, which can be represented as a vector in $\mathbf{x}$-space, defines the orientation of the decision plane, while the bias too defines the position of the plane in terms of its perpendicular distance from, the origin [Bishop, 1995, page 79]

3.2.3 Network Topologies

Regardless of the type of the activation function and its properties that affect the design of the neural network, the model of connection between neurons is another issue that should be considered. The topology of the neural network dictates the data flow, the training algorithm and the overall performance that can be achieved using the networks under study.

Generally, there are two main patterns of connections:

I) Feedforward networks where the data propagate from input layer to output layer in a feedforward way; there are no connections to facilitate the data flow from any layer to the same layer or to any other previous layer. The Perceptron, Adaline and MLP are some examples of the feedforward networks [Fausett, 1994].

II) Recurrent networks are networks that allow feedback connections. The main advantages of these connections are the dynamical properties of the network. In some applications, the network can be designed in a way such that the system will converge to a stable state by assuming that this stable state is the desired output. On the other hand, in some applications the system undergoes oscillation behavior where the activation values change significantly. In such cases, the dynamical behavior represents the desired output [Pearlmutter, 1990]. Some examples of recurrent networks can be found in [Hopfield, 1982; Kohonen, 1977].

3.2.4 Training of Artificial Neural Networks

In order to have the desired output, the weights (the parameters) of the neural network have to be configured. In principle, two approaches are available to specify the weights' values. For explicate approach, the weight configured is based on available knowledge about the problem under study. The implicate approach is another possible methodology. In this way, the weights of the network have to be initialized by some values followed by a training process which tends to change the weights' values, based on some learning rules, to meet the requirement of the application.

In general, the implicate approaches dominate a wide range of applications of the neural networks. In such methods, the training process falls into two distinct categories:

I) Supervised Learning:

The training is achieved by providing the network with inputs and their desired outputs pairs. Usually, this information can be provided by the user. However, some systems provide the neural network, which constitutes part of the system, with the required data (self-supervised).

II) Unsupervised Learning:

This method is also called self-organization. In these techniques, the training here is the process in which the system alters the weights based on the intrinsic properties in the input data; the system is assumed to discover statistically salient patterns within the input. Therefore, with no predefined set of desired outputs to be followed, the system has to produce its own set of rules.

Regardless of the paradigm of training, there should be a rule to alter the weights' value; adapting the parameters of the network in a way that makes the system achieve the desired goal. Generally, most learning rules are a development of the Hebbian learning rule [Hebb, 2002]. The underlying principle is that if two neurons are active simultaneously, then the strength of their connection should be increased. For instance, if i gets an input form j, Hebbian rule states that the weight w_{ij} should be updated by:

$$\Delta w_{ij} = \eta y_i y_j \tag{3.10}$$

where η is a positive constant called the learning rate. Least Mean Square (LMS), also called Delta rule, is another learning rule. However, rather than using the actual output of neuron j it facilitates the error value between the j outputs and its desired output:

$$\Delta w_{ij} = \eta y_i (t_j - y_j) \tag{3.11}$$

where t_j is the target that provided by the user. This approach is widely known to be related to the Backpropagation (BP) algorithm, which will be discussed later.

3.3 Multilayer Perceptron (MLP) Neural Networks

3.3.1 FeedForward Neural Networks

McCulloch & Pitts [1943] model (MP), which slightly mimics the structure of the organic neuron, is similar to the model proposed by Fisher [1936] to carry out statistical classification. Hebb [2002] method has given MP neurons the ability to learn. Hebb's method states that if output of the network is close to the desired target, then the weights should be modified in a way that makes the network produce a similar response for similar inputs in the future. On the other hand, if the output of the network is far from the desired target, then the weights should be modified to decrease the reported error. Rosenblatt [1958] studied Perceptron network, which contains a single layer of neurons. Then, the Perceptron Learning Rule has been proposed in [Rosenblatt, 1962] to specify suitable weights for classification problems.

In spite of all these brilliant developments, the single layer network was still only able to solve quite small numbers of applications. One way to overcome these limitations was by introducing a network with many layers of neurons. This formed the MLP which is widely in use today. Minsky & Papert [1972] discussed the potential of two layers feedforward network to overcome many limitations. However, there was no solution to the learning problem, that is, there was no specified method to adjust the weights that connect the input layer to the hidden layer. Here, any layer between the input and the output layers is called hidden layer. Many algorithms were proposed later to solve this problem [Parker, 1985; Rumelhart *et al.*, 1986; Werbos, 1974]. The underlying principle of these solutions is that the errors of the hidden layer neurons are specified by feeding backwards the errors of the neurons in the output layer. Therefore, the name of this approach is the BP learning rule, and it represents a generalization of the delta rule for non-linear activation functions and multi-layer networks. By introducing this technique, a wide range of models with different connection structures or architectures can be trained. Therefore, the academic and industrial interest has emerged and developed. Currently, neural networks have many applications in many fields.

Let one data point p be represented by a vector x^p, and the target vector for this point be t^p. By feeding the network with this data point it produces the output y^p; which has the same form as t^p. The weights of the network are adjusted to minimize the total square error:

$$E = \frac{1}{2}\sum_{p}(t^p - y^p)^2 \tag{3.12}$$

The importance of this error function is that it is smooth and differentiable.

A MLP network has a layered structure. Figure 3.9 shows a model of a standard two-layers perceptron. The neurons of each layer receive their input from neurons from a directly preceding layer and send their output to neurons in a directly succeeding layer. There are no connections within a layer. In Figure 3.9, the m inputs are fed into the h layer of N hidden neurons. The input neurons are merely 'fan-out' units; no processing takes place in this layer. The activation of a hidden unit can be any one of the functions discussed in Subsection 3.2, which consists of the weighted inputs plus a bias as given in (3.1). The output of the hidden layer is distributed over the next layer, in this case K Output's neurons.

Generally, BP can be applied to networks with any number of layers. However, many studies have shown that using one hidden layer is enough to approximate any nonlinear mapping with an acceptable degree of error, assuming that the activation function of the

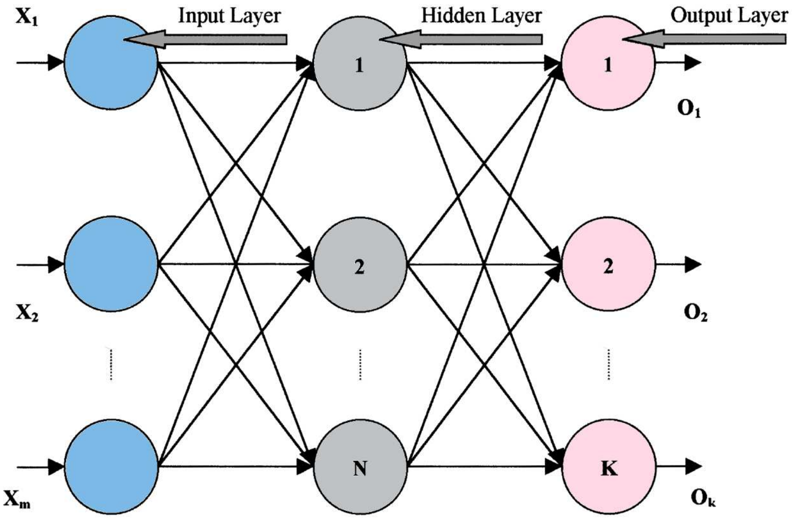


Figure 3.9: A multi-layer neural network architecture

hidden layer is non-linear, such as the sigmoidal function, [Cybenko, 1989; Funahashi, 1989; Hartman *et al.*, 1990; Hornik *et al.*, 1989]. The hidden layer differs from the other layers with its flexible and unfixed size; some techniques allow the learning rule to update the hidden layer size based on the characteristics of input data. The hidden layer is often utilized to force the network to mimic a desired system model and, at the same time, to sustain the ability to generalize to cover new data.

In Figure 3.9, let $x_0 = 1$ be the bias input and w_{i0} the bias "weight". The operation of the network can be defined by:

$$\begin{aligned} y_i^{(H)} &= f^{(H)}\left(\sum_{j=0}^{m} w_{ij}^{(HI)} x_j\right) \\ y_i^{(T)} &= o_i = f^{(T)}\left(\sum_{j=0}^{N} w_{ij}^{(TH)} y_j^{(H)}\right) \end{aligned} \tag{3.13}$$

where $\mathbf{X}$, $\mathbf{Y}^{(H)}$, $\mathbf{Y}^{(T)}$ (also called $\mathbf{O}$) are the input, the hidden layer output and the output layer output (network output) vectors, respectively. $w_{ij}^{(HI)}$, $w_{ij}^{(TH)}$ are the weights that connect hidden to input and output to hidden layers, respectively. $f^{(H)}$, $f^{(T)}$ are the activation functions of the hidden and output layer, respectively. Equations (3.13) map the input vector into an output vector through the hidden layer. The weights that connect the layers represent the parameters of this system.

3.3.2 Hidden Layer Configuration & Generalization

The number of hidden layers is an important issue in the configuration process of the neural networks. A large number of hidden neurons yield good performance when it is fed with input patterns that belong to the training data. But, the large hidden neurons' number would lead to over-fitting to the training set [Ham, 1994; Vogl *et al.*, 1988]. With the hidden layer nodes' number less than the number of degrees of freedom within the training data, the hidden neurons try to produce an orthogonal set of variables. By applying some more conditions, these internal variables construct a linear (nonlinear) principal component representation of the attribute of the input set. In this case, the final system will be able to deal with noisy data. Therefore, the multilayer perceptron has the ability to generalize to cover the new and unseen data by modeling the abstract features of the input set, and the hidden layer forms a detector for these features.

3.3.3 Independent Validation

Since using the same dataset for training and testing will lead to biased error rates, it is a common practice to divide the available data into two datasets; the training dataset is usually used to build the system (training) and the testing set is used to estimate the error rate and evaluate the classification system. On the other hand, the complex structure of the classification model may incur over-fitting problem and perform badly on the testing dataset especially with noisy data. Therefore, in order to get an effective classifier, the system should be as simple as possible to be used with noisy datasets and, at the same time, very complex to be used with noise-free datasets.

The basic idea is that randomly selected members of the dataset, considered as training data, should be used to establish the classification rule (toning the weights of the network). Another part should be used to test the resulted rule; usually the true classification of these data points are known, but have not been fed to the classifier. Then, by comparing the classifier outputs and the true classification, an unbiased error rate of the classifier can be estimated. Apart from this two sets, training and testing, there is a good procedure called independent-validation [Haykin, 1999]. Independent-validation consists of dividing the data, randomly, into three sub-sets. The first two are the training and testing sets, and the third set is validation set. In this technique, the training set is used usually to tune the network weights and the validation set is considered as a validating data; calculating the error rate which is the reference to stop the training process. Finally, the test data is used to asses the generalization of the network.

3.3.4 Back-Propagation Algorithm (Delta Rule)

A simple network, such as in (3.1), is able to form a linear relation between the input and the output. As has been highlighted, the network defines a hyperplane in the high dimensional input space, and this fact can be generalized to multi inputs (outputs) situation. Let our objective be to train the network to make this hyperplane fit an input patter x^p and its target t^p. Now, for every input sample the network output o^p will differ from the target t^p by $(y^p - t^p)$. Finally, let's call the summation output s, equivalent to *net* in Figure 3.1. The delta rule uses an error function based on this difference value to tune the weights.

Based on the error function in (3.12), which is the summed squared error, the total error is defined by:

$$E = \sum_p E^p = \frac{1}{2} \sum_p (t^p - y^p)^2 \tag{3.14}$$

where index p refers to patterns in the input sets, and E^p is the error of patter p. The LMS, the delta rule, searches for weights' values that minimize the error function by using a method called gradient descent. The idea is to make a change in the weight proportional to the negative of the derivative of the error as measured on the current pattern with respect to each weight:

$$\Delta_p w_j = -\eta \frac{\partial E^p}{\partial w_j} \tag{3.15}$$

where η is the learning rate, a constant of proportionality. The derivative is:

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial y^p} \frac{\partial y^p}{\partial w_j} \tag{3.16}$$

When the linear Eqn. (3.1) is used:

$$\frac{\partial y^p}{\partial w_j} = x_j \tag{3.17}$$

and

$$\frac{\partial E^p}{\partial y^p} = -(t^p - y^p) \tag{3.18}$$

we have

$$\Delta_p w_j = \eta \delta^p x_j \tag{3.19}$$

where $\delta^p = t^p - y^p$ is the difference between the target and the neuron outputs for data point p.

In order to deal with neurons with nonlinear activation function, this rule should be generalized. Let's define the function by

$$y_k^p = f(s_k^p) \tag{3.20}$$

where s_k^p is calculated by (3.1). To get the desired generalization of the delta rule, we must set

$$\Delta_p w_{jk} = -\eta \frac{\partial E^p}{\partial w_{jk}} \tag{3.21}$$

The error measure E^p is defined as the total quadratic error for pattern p at the output neurons:

$$E = \sum_p E^p = \frac{1}{2} \sum_{i=1}^{K} (t_i^p - y_i^p)^2 \tag{3.22}$$

We can write:

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{jk}} \tag{3.23}$$

By (3.1), we see that the second factor is

$$\frac{\partial s_k^p}{\partial w_{jk}} = y_j^p \tag{3.24}$$

When we define

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} \tag{3.25}$$

we will get an update rule which is equivalent to the delta rule as described in (3.19), resulting in a gradient descent on the error surface if we make the weight changes according to:

$$\Delta_p w_{jk} = \eta \delta_k^p y_j^p \tag{3.26}$$

To compute δ_k^p from (3.25), we can write

$$\delta_k^p = -\frac{\partial E^p}{\partial y_k^p} \frac{\partial y_k^p}{\partial s_k^p} \tag{3.27}$$

By (3.20), we see that

$$\frac{\partial y_k^p}{\partial s_k^p} = f'(s_k^p) \tag{3.28}$$

Assuming that the neuron k is an output neuron, we will have

$$\frac{\partial E^p}{\partial y_k^p} = -(t_k^p - y_k^p) \tag{3.29}$$

which is the same result we obtain with the standard delta rule. Substituting this and (3.28) into (3.27), for any output neuron, we get

$$\delta_k^p = (t_k^p - y_k^p) f'(s_k^p) \tag{3.30}$$

This will work fine for the output neuron, but what about the hidden neuron? Let h the hidden neuron, then we should specify the contribution of the neuron h to the output error of the network. However, the error measure can be written as a function of the net inputs s_j from hidden to output layer; i.e., $E^p = E^p(s_1^p, s_2^p, \ldots, s_j^p, \ldots)$ and we use the chain rule to write

$$\begin{aligned} \frac{\partial E^p}{\partial y_h^p} &= \sum_{o=1}^{K} \frac{\partial E^p}{\partial s_o^p} \frac{\partial s_o^p}{\partial y_h^p} \\ &= \sum_{o=1}^{K} \frac{\partial E^p}{\partial s_o^p} \frac{\partial}{\partial y_h^p} \sum_{j=1}^{N} w_{ho} y_j^p \\ &= \sum_{o=1}^{K} \frac{\partial E^p}{\partial s_o^p} w_{ho} \\ &= \sum_{o=1}^{K} \delta_o^p w_{ho} \end{aligned} \tag{3.31}$$

Substituting this into (3.27), for any hidden neuron, we get

$$\delta_h^p = f'(s_h^p) \sum_{o=1}^{K} \delta_o^p w_{ho} \tag{3.32}$$

Equations (3.30) and (3.32) give a recursive procedure for computing the δ's for all neurons in the network, which are then used to compute the weight changes according to (3.26). This procedure constitutes the generalized delta rule for a feed-forward network of non-linear neurons, and its significance is distributing the error of an output neuron to all the hidden neurons that it is connected to, weighted by this connection.

Generally, many experiments have to be conducted to specify the largest value for η that will work. However, large learning rate will lead to significant change in the gradient. A practical technique is used usually to overcome this problem. The idea of this modification is to update the weights in the direction that is a linear combination of the current gradient of the current error surface and the one obtained in the previous step of the training. This is accomplished by adding a momentum term to (3.21) involving a parameter $\alpha \lll 1$:

$$\Delta_p w_{jk} = -\eta \frac{\partial E^p}{\partial w_{jk}} + \alpha \Delta_{old} w_{jk} \tag{3.33}$$

where $\Delta_{old} w_{jk}$ refers to the most recent weight change. The momentum term improves the performance of the convergence of the standard algorithm by introducing stabilization in weight changes. Based on (3.33), if the current change is in the same direction as the previous one, the algorithm will increase the rate of change. On the other hand, if the current change is not in the same direction as the previous one, the algorithm will decrease the rate of change.

3.4 Kohonen Neural Networks

3.4.1 Self-Organizing Neural Networks

All different types of self-organizing neural networks have the ability to assess the input patterns that are fed to the network and to learn the inherent features presented in the inputs on their own. Therefore, they have the ability to categorize the input patterns into groups each having members of similar characteristics. The absence of the exemplar gives these algorithms the name, i.e., unsupervised learning algorithms. Generally, the lack of classes' information that guide the clustering process gives the unsupervised learning its importance. The unsupervised methods can often detect some unknown features in the data.

In this type of learning, the network performs frequent modifications to its weights in response to the input data. The weights' adjustment is achieved by a set of learning rules. Basically, from random patterns that applied to the network, a global order will emerge. This global order represents the desired outcome of these types of neural networks.

Some classes of self-organizing networks are based on competitive learning where the output neurons compete among themselves to specify a winner. One type of these networks is the Kohonen self-organizing map (SOM) [Kohonen, 1982, 1989]. In Kohonen network, the order of the output neurons is an application dependent. The order of the neurons specifies the neighboring neurons for every neuron. When input patterns are fed into the network, the weights of the network will be adjusted in a way that keeps the order present in the input space. The input patters, which are close to each other in the input space, would be mapped to close output neurons in the output space. In other words, Kohonen network transforms input patterns, of any dimension, into 1- or 2-dimensional map of features in a topologically ordered fashion.

During training process, the winning neuron, the nearest neuron to a given training pattern, adjusts its weights to be closer to that pattern and, at the same time, affects the neighboring neurons in the network topology. This leads to a smooth distribution of the network topology in a non-linear subspace of the training data.

3.4.2 Competitive Learning Algorithm

In the competitive learning, the learning procedure clusters the input patterns into groups, each has members with shared characteristics among themselves. The competitive learning network, such as Kohonen network, is fed only by input patterns, and it applies an unsupervised learning algorithm. In this algorithm, when a specific input pattern is provided, one output neuron will be activated. In the resulted network, all the patterns in one cluster will activate the same winner. In general, there are two learning rules that are used to specify the winner: 1) dot-product; 2) Euclidean distance.

Let $\mathbf{x}$ and $\mathbf{w}_j$ be normalized input and weight vectors. In dot-product winner selection, each output neuron j calculates its activation value y_j according to the dot product of input and weight vector:

$$y_j = \sum_i w_{ij} x_i = \mathbf{w}_i^T \mathbf{x} \tag{3.34}$$

Then, the winner neuron k would be:

$$\forall i \neq k : \quad y_i \leq k \tag{3.35}$$

Therefore, $y_k = 1$ and the rest are 0. Once the winner k has been selected, the weights updating rule will be:

$$\mathbf{w}_k(n+1) = \frac{\mathbf{w}_k(n) + \eta(\mathbf{x}(n) - \mathbf{w}_k(n))}{||\mathbf{w}_k(n) + \eta(\mathbf{x}(n) - \mathbf{w}_k(n))||} \tag{3.36}$$

where all $\mathbf{w}$ are normalized. And only the winner's weights are updated.

Based on the rule (3.36), each time an input $\mathbf{x}$ is presented, the nearest weight vector to this input is selected and therefore rotated towards the input.

On the other hand, in order to eliminate the normalization, the nearest neuron k is selected such that the weight vector $\mathbf{w}_k$ closest the input $\mathbf{x}$ based on the Euclidean distance measure:

$$\forall i \neq k : \quad k : ||\mathbf{w}_k - \mathbf{x}|| \leq ||\mathbf{w}_i - \mathbf{x}|| \tag{3.37}$$

where only the winner neuron's weights are updated. Here, instead of rotating the weight vector, the weight updating must be changed to implement a shift towards the input:

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) + \eta(\mathbf{x}(n) - \mathbf{w}_k(n)) \tag{3.38}$$

However, in the applications of Kohonen neural network, not only the winner neuron's weight k is updated but its neighbors are also adapted using the learning rule:

$$\forall o \in S_k : \qquad \mathbf{w}_o(n+1) = \mathbf{w}_o(n) + \eta g(o,k)(\mathbf{x}(n) - \mathbf{w}_o(n)) \tag{3.39}$$

where S is a predefined neighborhood of neuron k. $g(o,k)$ is a decreasing function of the defined distance between o and k such that $g(k,k) = 1$. For instance, $g(o,k)$ can be a Gaussian function (in one dimension) such that $g(o,k) = \exp(-(o-k)^2)$. Therefore, the behavior of this function will tend to make the input's pattern, which are similar (close), to be mapped on neighboring neurons in output layer.

3.5 Microarray Image

Based on the discussions made in Chapter 2, we have the following observations. 1) Microarray image segmentation, see Fig. 3.10, is an important yet challenging problem since the discrimination between the foreground and the background signal strongly affects the gene expression value for every spot. 2) Improving the quality of the extracted gene expression is one of the ultimate goals of microarray image processing. 3) It is essential that the segmentation stage be given significant attention. 4) Many microarray image segmentation (clustering) methods have been proposed with some producing better results than others. In general, the most effective approaches require considerable run time (processing power) to process an entire image. 5) Although many approaches have been proposed in the literature, there has been little progress on developing sufficiently fast, efficient yet effective algorithms to segment a microarray image by using up-to-date techniques such as ANN approach. Based on these observations, we, in this paper, aim to propose a novel method for segmenting microarray images with hope to produce results that are as good as, if not better than, the results of most advanced microarray segmentation algorithms *but with less running time.*

In this chapter, a new segmentation method, which is based on artificial neural networks, is investigated. Towards the end of this chapter, the method is tested on a set of real cDNA microarray images. Quantitative comparisons between the proposed algo-

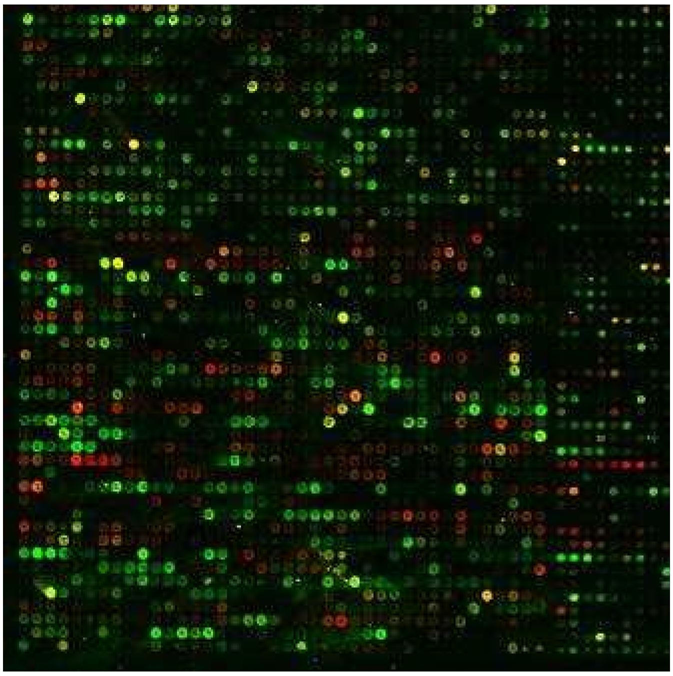


Figure 3.10: Two-channel cDNA microarray image

rithm and commercial GenePix software are carried out in terms of subjective comparison methods. It is shown that the proposed technique is not only very powerful in clustering but also very efficient in terms of runtime. To deliver the proposed method and justify its advantages, the remainder of the chapter is arranged as follows. 1) The method of gridding images used to create training and test data for the neural networks is briefly discussed. 2) The approach of determining the optimal number of spot region classes as well as the competitive neural network with kohonen learning algorithm, which is used to classify the spot's region, is described. 3) The proposed ANN for the segmentation of the spot's local area is put forward. 4) The results of the new clustering approach are analyzed. 5) The results from this method are validated on real-world microarray images and also compared with the commercially available software GenePix.

3.6 The Proposed Approach

The task of spot segmentation falls within the category of classification, that is, assigning pixels into spot and non-spot classes. ANNs are a well established tool for classification problems [Bishop, 1995]. Once trained, ANNs can produce very impressive classification results in a significant short runtime. ANNs are motivated by an interest in modeling the working of neurons [Hebb, 2002; McCulloch & Pitts, 1943], the cells that comprise the brain. Just like a brain, neural networks can be taught new skills on how to specifically recognize patterns. The training of a MLP neural network (with delta rule learning algorithm) involves the use of a training set, a set of input values and the desired response for each. A natural question is that how to obtain the quality training data in the case of the spot segmentation problem. The most logical answer is to train the network with output from one of the best spot segmentation algorithms developed to date, namely the CC approach [Fraser *et al.*, 2010]. In addition, GenePix is used to produce another training set in order to validate the proposed method.

3.6.1 Creation of Training Sets

The training sets for the neural networks, which are used in the subsequent stages of our new segmentation approach, have to be created for the use of ANN. These sets contain: 1) a set of inputs as spots' regions - areas taken from a raw microarray image with each containing a single spot as well as some background and possibly noise pixels; and 2) the desired outputs, which are corresponding to these inputs, as binary images.

A complete blind microarray image gridding framework developed in [Morris, 2008] is used to accomplish the spotting task. The input of the framework is the microarray image that can be at any resolution, and the gridding is accomplished without any prior assumptions. The framework includes an Evolutionary Algorithm (EA) and several methods for various stages of the gridding process including sub-grid detection. Actually, it is not critical which method is used to accomplish these tasks as long as the result is appropriate. Nevertheless, the results of [Morris, 2008] have experimentally demonstrated to be both robust and effective for this task.

The chosen image features many of the characteristics that hinder traditional segmentation approaches. Such characteristics include, but are not limited to, poorly expressed spots, malformed spots, high valued noise artefacts and uneven background. A square region is taken around each spot center as the spot region. These regions are taken from the raw microarray image to form the input set. The corresponding regions are taken from the binary output of the CC algorithm [Fraser *et al.*, 2010] in order to form the desired output set. A second output set is created by taking output of GenePix package, which will be used to validate our developed ANN-based segmentation approach.

3.6.2 Single Neural Network Implementation

The first attempt made at segmentation is performed using a three layer neural network with the BP learning algorithm. This network uses 50% of the CC dataset as training set and 30% of the same set for validation purposes.

The output of the neural network for a given spot region is a real valued image of the same size. The higher a given pixel value is the more likely it should belong to the foreground. Each output region is, by threshold, assigned into either the foreground or the background. The threshold value is obtained using Otsu's method [Otsu, 1975].

The trained network is then tested using approximately 20% of the spots from the CC dataset. The outcome for the majority of the spots is a very poor match to the desired output. This result highlights the inability of the network to cope with poorly expressed spots whose values are close to or below their local background values.

With a typical microarray image, there is a wide variety of spots. Intensity wise, the spots range from ideal spots, which are valued well above their local background, down to very poorly expressed spots, which may be valued below their local background. Considering the spot's shape, the spots range from perfectly formed round shape spots with clearly defined edges down to disfigured spots with blurred edges and even disconnected regions. Clearly, it is inconceivable for a single BP neural network to learn how to segment all these contradictory types of spots. A way to simplify this task, with the ability to process all the spots within an image, is to classify the spots within a microarray image into one of the several classes. For each class, a BP neural network can achieve the segmentation for that specific class, see Figure 3.11.

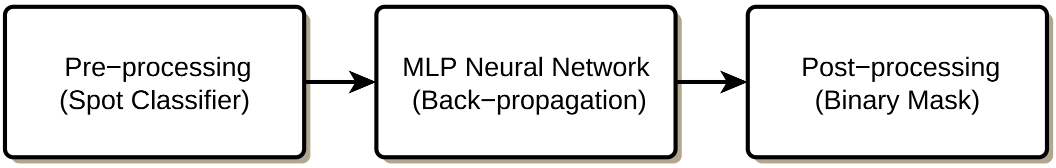


Figure 3.11: Microarray image processing steps

3.6.3 Spot Classifier

One appropriate solution to classify the spot's regions is to employ an unsupervised competitive network. This network seeks to find patterns/regularities, known and unknown, in the input data. Figure 3.12 shows an illustration of the architecture of the competitive layer [Demuth *et al.*, 2007].

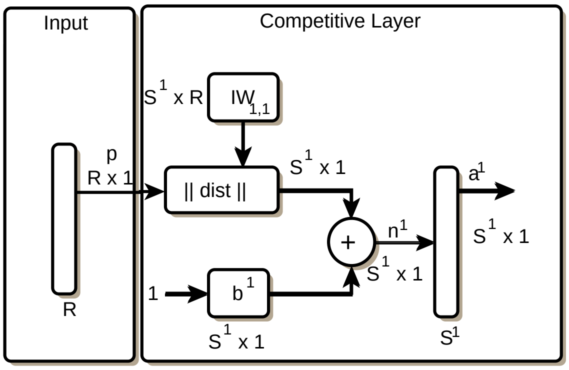


Figure 3.12: The architecture for the competitive later.

In order to obtain better results, the training data is pre-processed in the way depicted in Figure 3.13. 1) The extreme values within each spot region are eliminated by applying a median filter with window size 3×3. 2) The values within each spot region are normalized between values of 0 and 1. 3) A Laplacian pyramid filtering technique [Burt & Adelson, 1983] is applied in order to extract gradient information of the cropped image. Therefore, information of less importance in the image has been eliminated.

The learning algorithm used to train the network is the Kohonen Learning algorithm (Winner-Take-All) [Fausett, 1994; Ham & Kostanic, 2000]. In this kind of learning, the units of the network update their weights by forming a new weight vector that is a linear combination of the old weight vector and the current input vector. Typically, the cell whose weight vector is the closest to the input vector is allowed to learn. The measure of closeness/distance can be specified by two methods, both of which are based on the assumption that the weight vector of each cluster cell serves as an exemplar for the input vectors which have been assigned to that cell in the learning stage. In the first method, the smallest squared Euclidian distance between the input vector and weight vectors marks the winning cell. In the second method, the largest dot product marks the winning cell, where the dot product can be interpreted as a correlation measure between the input and weight vectors [Haykin, 1999]. Using Kohonen training method, the individual neurons of the network learn to specialize on ensembles of similar patterns; in doing so, they become feature detectors for different classes of input patterns.

In order to determine how many classes the spot regions could be grouped into, an iterative training approach is used. The network is trained to classify the spot regions into

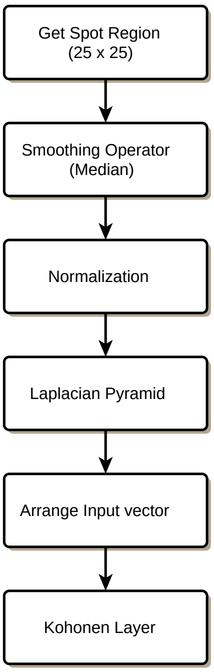


Figure 3.13: Spot classifier.

$(3, 4, 5, ..., N)$ classes. Working with all the spots from real microarray images, this method can define 9 classes of spot regions. Figure 3.14 shows ten spot regions from each of the 9 classes.

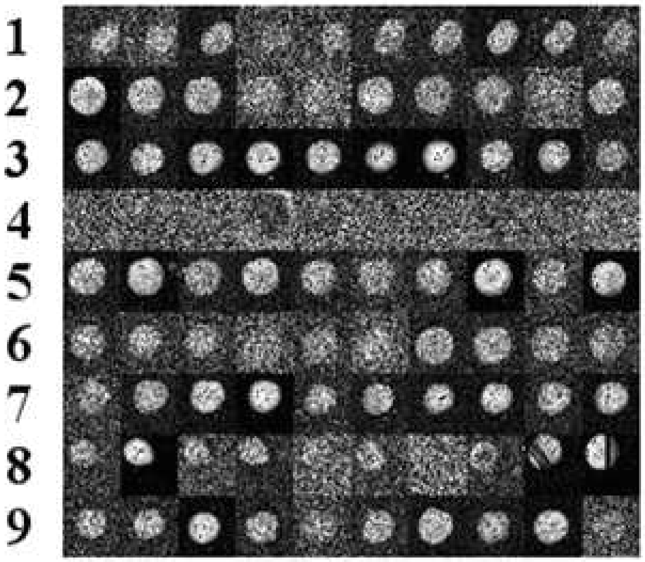


Figure 3.14: 10 examples from each of the 9 classes of spots determined by the network.

By investigating the classification outputs, some key rules for the membership can be inferred such as the degree of noise, the size of spot, the position of the spot and the distribution of the pixels' values. However, in some of the classes the rules for membership are more obvious than they are in the others. Class 4, for example, is clearly the class that contains the noisiest spot regions while class 3 contains spots that are in high contrast to their background. Not only has the iterative training approach detected the number of classes (9) to divide the many spot regions within an image into, but it has also provided a mean to classify quickly a given spot region into one of the 9 classes.

3.6.4 Multiple Neural Networks Implementation

Having determined the number of classes to divide the spot regions into (9) classes, the next task is to train a neural network for each class. This is accomplished in the same manner as the single neural network experiment described above. However, it is worth mentioning that each network is trained with its own custom training set where all the spot regions, within each training set, belong to the same spot region class.

One possible way to accomplish this task is to use Pattern Association [Fausett, 1994]. In such a type of neural networks, learning is the process of forming association between related patterns. A key question for all associative nets is that how many patterns can be stored before the net starts to forget the learned pattern. Many factors influence the number of patterns that can be learned. The complexity of the patterns and the similarity of the input patterns, which are associated with significantly different output patterns, both play a role.

Based on the previous discussion, the associative nets are not a good solution for the addressed problem. Therefore, a new, partially connected and 3-dimensional MLP topology is proposed to tackle this problem, see Figure 3.15. In addition to the pre-processed intensity value for each pixel, additional inputs are determined to result in a better segmentation process. A 3 × 3 window is placed around every pixel, and the mean and the standard deviation of the window are used as inputs to the network. The network, therefore, features 1875 inputs as well as a hidden layer with 625 neurons and 625 outputs. Each of the output and hidden layers is arranged in 2D 25×25 array. More specifically, every pixel (represented by pixel's value, the mean and the standard deviation) is connected to one hidden cell. Every cell in the output layer is connected to the corresponding hidden cell and its neighboring cells located within a prescribed sphere of influence N_r of radius $r = 1$ centered at this hidden cell.

Two sets of nine neural networks are trained. The first is trained with the segmentation

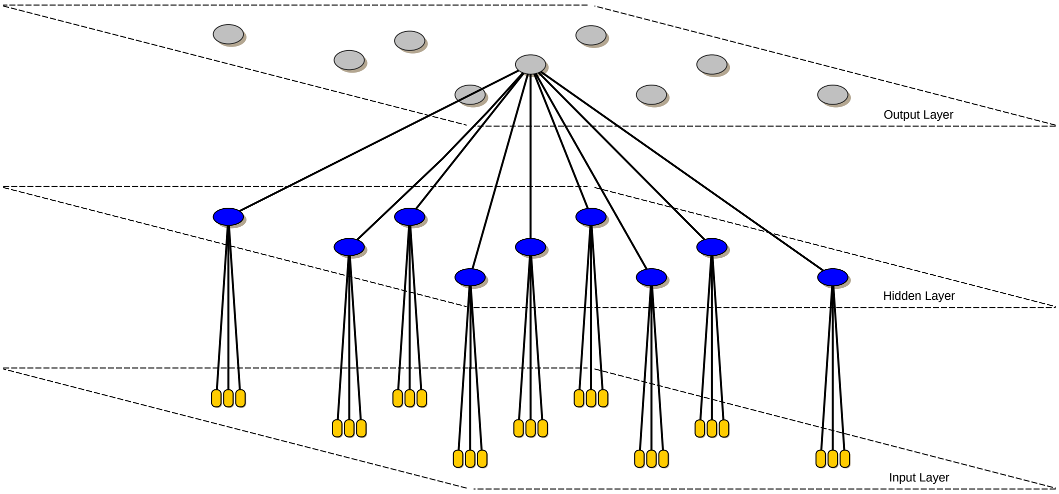


Figure 3.15: The proposed network architecture

results of the GenePix package, then the second is trained with the segmentation results from the CC algorithm. Training data is taken from the raw image, GenePix output image and the CC output image, which are used to produce an image with pixels' values falling within $[0, 1]$. Therefore, further processing is required to get the final binary image.

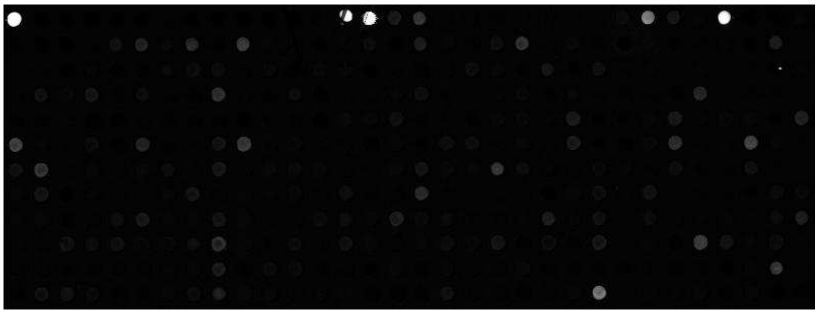


Figure 3.16: Microarray image sample input

In order to produce a binary mask, the Otsu's thresholding technique [MathWorks, 2007] is used. Figures 3.16 and 3.17 show the input sample and its output.

3.7 Experimental Results

In order to quantify the performance of different filtering methods, a quality measure is required in order to evaluate the validity of the pixels selected for a given spot. Note that

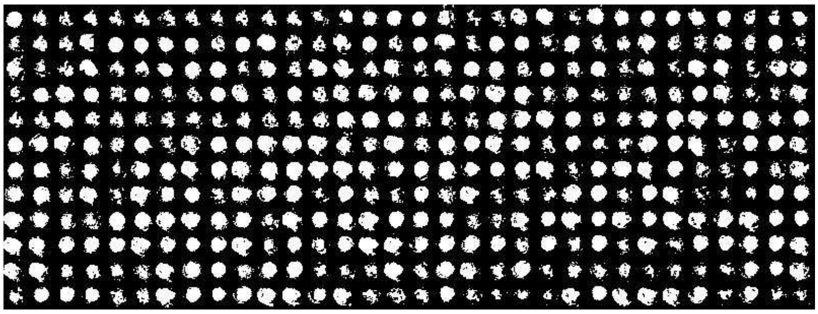


Figure 3.17: Example of the MLP segmentation algorithm output.

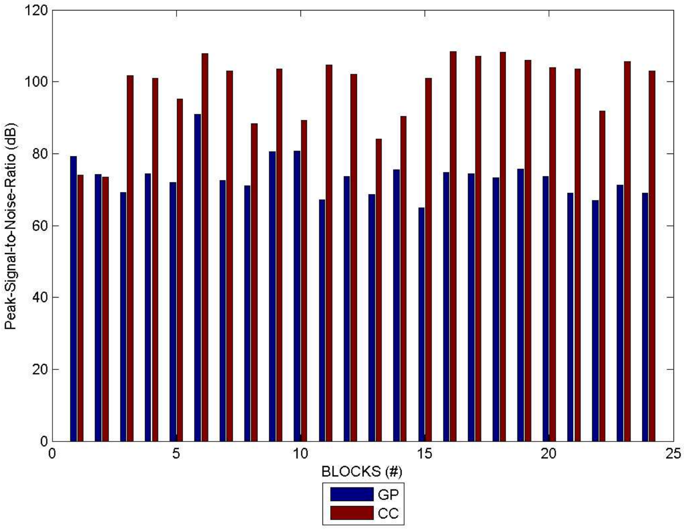


Figure 3.18: Comparison of segmentation results for 24 subgrids between GenePix (GP) and the copasetic clustering (CC) algorithm.

the proposed algorithm produces a binary mask that classifies the pixels as belonging to either signal (the gene spots) or noise (the local background). For this purpose, an image quality measurement, known as the Peak Signal-to-Noise Ratio (PSNR), is used and the rational is justified as follows.

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are two error metrics frequently used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image. The lower the value of MSE, the lower the error. The PSNR represents a measure of the peak error. The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g. for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. Though a higher PSNR would normally indicate that the reconstruction is of a higher quality, in some cases one reconstruction with a lower PSNR may appear to be closer to the original than another.

To compute the PSNR, the block first calculates the mean-squared error using the following equation:

$$MSE = \frac{\sum\limits_{m,n}[I_1(m,n) - I_2(m,n)]^2}{M * N} \tag{3.40}$$

where M and N are the numbers of rows and columns in the input images, respectively, I_1 is the grayscale and I_2 is the mask image. Then, we obtain the PSNR using the following equation:

$$PSNR = 10 \log \left[\frac{R^2}{MSE} \right] \tag{3.41}$$

where R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

The PSNR returns the ratio between the maximum value in the signal and the magnitude of the signal's background noise. The output of the PSNR is decibel units (dB). An increase of 20dB corresponds to a ten-fold decrease in the MSE difference between two images. The higher the PSNR value the more strongly the binary spot mask fits with the raw image surface. The PSNR is a much better measure of the suitability of a spot mask to the raw image area containing the spot as it is more resilient against large intensity ranges within the spot area compared to the MSE [Fraser *et al.*, 2010]. Figure 3.18 shows the PSNR comparison between the results of segmenting an image using the CC method and the GenePix software. Comparisons are made between all 2 subgrids within the image. Clearly, using copasetic clustering gives a much better segmentation outcome.

Having trained the two sets of nine neural networks, we use them to segment the image that they are trained from. The results are shown in Figure 3.19, where the four columns represent the PSNR for GenePix, CC, the neural network trained with the GenePix data, and the neural network trained with the CC data. In the first two subgrids, the neural network results are approximately equal to the results of segmentation using GenePix and the CC algorithm. Both neural networks even slightly surpass the results of the approaches from which they are trained in the second subgrid. In all other subgrids, the CC algorithm still produces the best results, but the ANN results are clearly very competitive with the GenePix results.

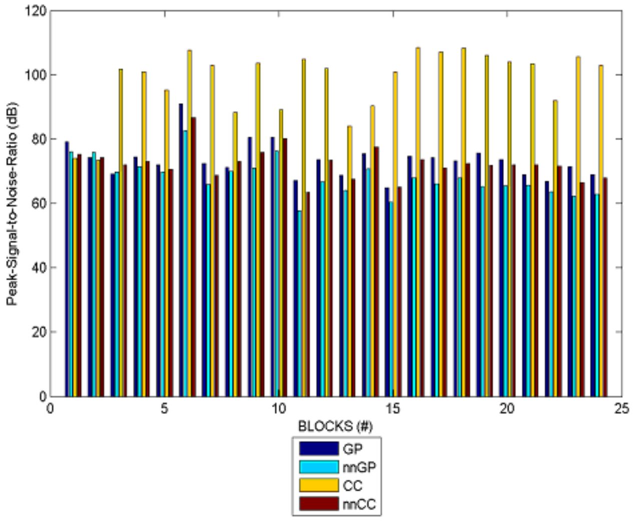


Figure 3.19: Comparison of segmentation results between genepix, the neural network trained with the genepix data (nnGP), CC and the neural network trained with the CC data (nnCC).

An advantage of the ANN based approach is the computational time saving. Once a network has been trained, it can be applied to many other images and achieve the task faster than GenePix and CC approaches. Figure 3.20 shows the results of the clustering of 48 subgrids in an image. The neural networks used are the same ones as those used to produce Figure 3.19. The CC algorithm does produce the best segmentation result for all 48 subgrids, but the NN results are clearly comparable to the results obtained using the GenePix package.

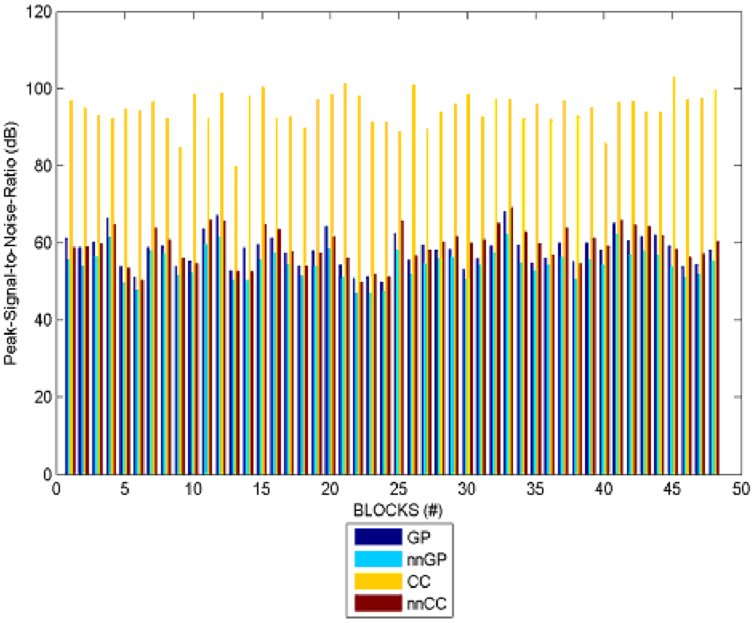


Figure 3.20: Comparison of segmentation results for 48 subgrids. The neural networks used did not see this image during their training.

3.8 Conclusions and Future Work

The task of spot segmentation is a critical one in the processing of any microarray image. Also, the impact of the accuracy and consistency with which spot pixels are identified is very significant. In this chapter several methods of segmenting microarray images have been discussed. One of the most advanced approaches is the so-called CC algorithm [Fraser *et al.*, 2010], which has been shown to surpass regularly the clustering results of the widely used GenePix package that relies heavily upon manual interaction. The main contribution of this chapter is the development of a new neural network based method for spot segmentation. This method not only produces very impressive results that are very competitive to the results obtained using the GenePix package, but it can also produce the outputs more quickly than previous approaches. Clearly, this is a valuable addition to the area of microarray segmentation.

Future work in this area will allow an improvement on the results of the segmentations achieved using a neural network. It may even be possible to train a network to produce consistently better segmentation results than the CC algorithm.

In order to improve the algorithm, many alternatives could be utilized. Among others, we list three future topics here.

- Our dataset, although big, is not vast enough to be sure that a single training would be enough for analyzing several microarray experiments. For instance, the slide might be a bit different when prepared by another protocol. The approach, that has been followed in this chapter, based on the assumption that the BP network is able to infer the rules by which the classification of the pixels will be achieved. However, the generalization study should be discussed in much more details during the design phase of the system.

  One possible improvement is to think about a second hidden layer in the BP networks, as it has been mentioned in Section 3.3.2. If the network has only one hidden layer, the neurons seem to interact with one another [Haykin, 2001]. In such a case, it is difficult to improve the approximation for one point in the mapping without degrading it at some other point. Therefore, considering two-hidden-layers BP network can be a potential development that could lead to a better generalization.

- In this chapter, a one-dimension Kohonen network has been used to divide the spots' regions into a maximum number of class. This can be considered in two ways: 1) Instead of the one-dimension Kohonen network, a two-dimensions self-organizing map competitive neural network [Haykin, 1999] could be used where the neurons are placed at the nodes of a lattice that is usually one or two dimensional, thus, the pixel's neighborhood will be squared shape. The neurons become selectively tuned to various input patterns or classes of input patterns throughout a competitive learning process. The locations of the neurons so tuned become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice. The self-organizing map is, therefore, characterized by the formation of a topographic map of the input patterns in which the spatial locations of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns. 2) Optimizing the number of spots' regions' classes, by conducting more experiments on a different number of classes, can be advantageous.

- A 5×5 window can be used to calculate some properties based on "sum and difference histogram" [Sridhar *et al.*, 1993]. Then, these properties would be used as input for fully connected neural network with target value either from CC or GenePix outputs. Furthermore, The employment of Maximum Covariance Technique may improve the generalization ability [Lehtokangas *et al.*, 1996].

Finally, a road map should be specified based on the outcomes of this chapter. Considering the results in Section 3.7, it should be mentioned that the neural network approach has given results outperforming the results of GenePix software, but it is unable to repeat the results of CC. In addition, the supervised applied methodology can only deal with cases for which they have been trained. Thus, when all the networks are learned from the output of CC or GP, the proposed method can only be at best as good as the method from which the training notation comes. On the other hand, the hardware implementation of the neural networks (BP and Kohonen) has been improved [Botros & Abdul-Aziz, 2002; Gadea *et al.*, 2000; Kim & Jung, 2004; Omondi & Rajapakse, 2006]. In particular, the currently available neurocomputers based on Field-Programmable Gate Arrays (FPGAs) have achieved a good capacity and performance, though it would be advantageous to regard a more specified architecture that is dedicated to image processing and much more suitable for local processing strategies.

In the following chapters, the investigation will be focused on: 1) The local adaptive strategies are the main approaches that will lead a highly successful developed system to deal with the problems at our hand. 2) Pre-filtering stage is a main step which helps to improve the outcome of the microarray image processing system. 3) The proposed method assumes no gold standard segmentation results. Therefore, it follows an unsupervised approach. 4) An analogic computer [Chua & Roska, 1992b] will be considered as a potential alternative to assure the hardware implementation, particularly, the suitability of these computers to analyze the images using a local strategy.

Chapter 4

Adaptive Segmentation of Microarray Image

The discussion in this chapter based on:

Zineddin, B., Wang, Z. & Liu, X., 2010. cDNA Microarray Segmentation: Adaptive Approach. IEEE Transactions on Image Processing, Submitted.

> *It makes all the difference whether one sees darkness through the light or brightness through the shadows.*
>
> David Lindsay

In this chapter, the approach will be built based on the conclusions of the previous chapter, particularly, the importance of the filtering stage and its implications on the final outcome of the microarray image processing. However, the proposed filtering methodologies have no effects on the quantification of the image. Their only role is to prepare the image for feature detection purpose. The principal objective of filtering is to process an image so that the result is more adequate than the original image for the specific task at hand, namely the segmentation process. Specifying the task is crucial, since the developed techniques are a problem oriented to a high degree. For instance, a method that is quite useful for enhancing X-ray images may not necessarily be suitable for enhancing microarray images. In spite of that, however, image filtering is one of the most interesting and visually appealing areas of image processing.

Referring to the background of Chapter 2, DNA microarray technology has enabled biologists to study all the genes within an entire organism to obtain a global view of genes' interaction and regulation. This technology has a great potential in obtaining a deep understanding of the functional organization of cells. Yet, it is still early in its development, and needs improvements in all the main stages of the microarray process. Therefore, this chapter is concerned with improving the processes involved in the analysis of microarray image data. The main focus is to clarify an image's feature space in an unsupervised manner. Rather than using the raw microarray image, it suggests to produce filtered versions of the image data by applying nonlinear anisotropic diffusion (see Section 4.2), so that the dynamic range of the image could be increased and, hence, a better ability of signal extraction could be achieved.

In this chapter, a novel segmentation algorithm is proposed. This algorithm is based on CNN computational paradigm (see Section 4.1), integrated with median and anisotropic diffusion filters. The AnaLogic CNN Simulation Toolbox for MATLAB (InstantVision Toolboxes for MATLAB) is used during the segmentation process. Quantitative comparisons among the proposed methods and GenePix are carried out in terms of objective and subjective point of view. It is shown that the analogic algorithm integrated with Complex Diffusion filter is the best one to be applied to achieve the segmentation.

4.1 Cellular Neural Networks

A Cellular Neural/Nonlinear Network is defined by two mathematical constructs [Chua & Yang, 1988b,d]: 1) A spatially discrete collection of continuous nonlinear dynamical systems called cells, where information can be encrypted into each cell via three independent variables called input, threshold and initial state; and 2) A coupling law relating one or more relevant variables of each cell to all neighboring cells located within a prescribed sphere of influence $N_{r(ij)}$ of radius r centered at ij. Analog circuit has played a very important role in the development of modern electronic technology. Even in our digital computer era, analog circuits still dominate such fields as communications, power, automatic control, audio and video electronics because of their real-time signal processing capabilities.

Conventional digital computation methods have run into a serious speed bottleneck due to their serial nature. To overcome this problem, a new computation model called "neural networks" has been proposed, which is based on some aspects of neurobiology and adapted to integrated circuits. The key features of the neural networks are asynchronous parallel processing, continuous-time dynamics and global interaction of network elements. Some encouraging, if not impressive, applications of neural networks have been proposed for various fields such as optimization, linear and nonlinear programming, associative memory, pattern recognition and computer vision.

4.1.1 Application Potential

The CNN paradigm provides a flexible framework to describe spatiotemporal dynamics in discrete space. Especially, the acceptance of the CNN approach as a computational paradigm [Chua & Roska, 1993b] and the design of the hardware architecture (i.e., the CNN Universal Machine CNN-UM [Chua & Roska, 1992b]), allow efficient VLSI implementation of analogue array-computing structures. Such devices possess a huge processing power that can be employed to solve numerically expensive problems. The CNN-UM is the first parallel, analogic stored program and visual array microprocessor that can be fabricated on a single chip [Cruz & Chua, 1998; Dominguez-Castro *et al.*, 1994, 1997; Linan *et al.*, 2003]. These devices are programmed by analogic algorithms [Csapodi & Roska, 1996; Rekeczky & Chua, 1999; Rekeczky *et al.*, 1995, 1999; Zarandy *et al.*, 1996], i.e. using analog operations in sequence combined with local logic at the cell level.

Over the last two decades, CNNs have been applied in a wide range of applications. In signal processing, CNNs show great promise in solving many complex problems that cannot be solved satisfactorily using conventional approaches [Chua *et al.*, 1991; Crounse, 1997; Crounse & Chua, 1995; Krieg *et al.*, 1990; Tanaka *et al.*, 1992]. In image processing, CNNs can be applied to perform many tasks, such as feature extraction and classification [Feng *et al.*, 2006; Fernández-Muñoz *et al.*, 2006; Haung *et al.*, 2009; Namba, 2008; Shitong & Min, 2006; Sziranyi & Csapodi, 1998], image thining [Matsumoto *et al.*, 1990a], motion detection or estimation [Dumontier *et al.*, 1999; Roska & Chua, 1990; Roska *et al.*, 1990, 1992; Shi *et al.*, 1993], objects counting [Bertucco *et al.*, 1998], process color images [Inoue & Nishio, 2009], detect and identify microscopic organisms[Tokes *et al.*, 2008], enhance the image [Abrishambaf *et al.*, 2008; Aizenberg *et al.*, 2001; Park & Nishimura, 2007; Su & Jhang, 2006], moving-object segmentation [Rodriguez-Fernandez *et al.*, 2008], robotic motion [Fasih *et al.*, 2008] and inn solving partial differential equations [Chedjou *et al.*, 2009; Chua, 1997; Kozek & Roska, 1996; Kozek *et al.*, 1995; Roska *et al.*, 1995].

4.1.2 Architecture of Cellular Neural Networks

The structure of cellular neural networks is similar to that found in cellular automata; namely, any cell in a cellular neural network is connected only to its neighbor cells. The adjacent cells can interact directly with each other. Cells not directly connected together may affect each other indirectly because of the propagation effects of the continuous-time dynamics of cellular neural networks, see Figure 4.1.

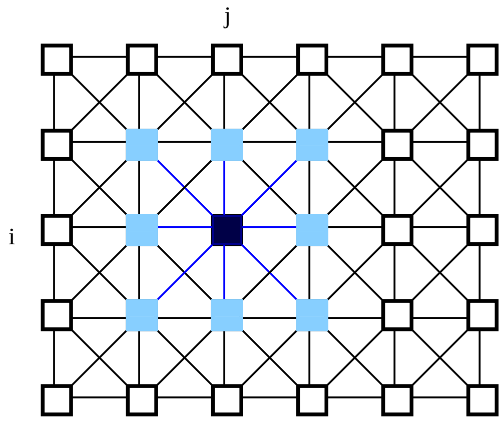


Figure 4.1: A 2-dimensional CNN defined on a squared grid. The ij-th cell of the array is colored by black, cells that fall within the sphere of influence of neighborhood radius $r = 1$ (the nearest neighbors) by blue

The basic circuit unit of cellular neural networks is called a cell. It contains linear and nonlinear circuit elements, which are typically linear capacitors, linear resistors, linear and nonlinear controlled sources, and independent sources. All the cells of a CNN have the same circuit structure and element values. A typical circuit of a single cell is shown in the Figure 4.2.

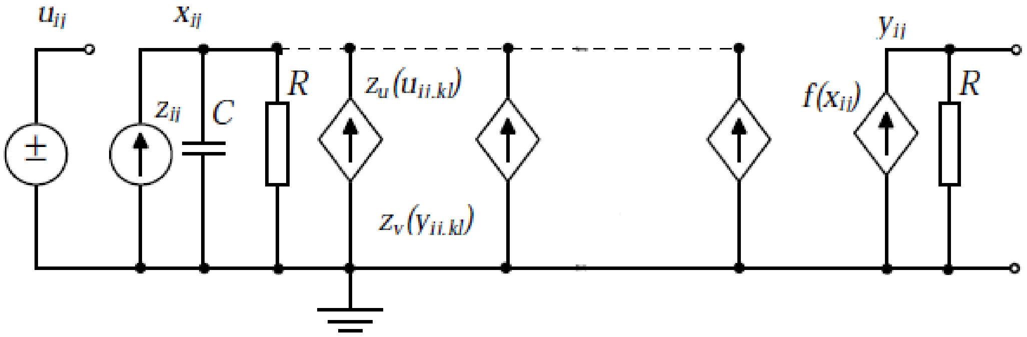


Figure 4.2: A CNN base cell circuit

Each cell contains one independent voltage source u_{ij} (Input), one independent current source z_{ij} (Bias), several voltage controlled current sources $Z_u(u_{ij,kl})$, $Z_y(y_{ij,kl})$, and one voltage controlled voltage source y_{ij} (Output). The controlled current sources $Z_u(u_{ij,kl})$ are coupled to neighbor cells via the control input voltage of each neighbor cell. Similarly, the controlled current sources $Z_y(y_{ij,kl})$ are coupled to their neighbor cells via the feedback from the output voltage of each neighbor cell. The time constant of a CNN cell is determined by the linear capacitor (C) and the linear resistor (R) and it can be expressed as $\tau = RC$.

As the basic framework, let us consider a two-dimensional $M \times N$ CNN array in which the cell dynamics is described by the following nonlinear ordinary differential equation with linear and nonlinear terms:

$$\begin{aligned}
C\frac{d}{dt}x_{ij}(t) &= -R^{-1}x_{ij}(t) + \sum_{k,l\in N_r} A_{i,j;k,l}y_{kl}(t) \\
&\quad + \sum_{k,l\in N_r} B_{i,j;k,l}u_{kl} + z_{ij} \\
&\quad + \sum_{k,l\in N_r} D_{i,j;k,l}(\Delta v_{nn}) \\
y_{ij}(t) &= f(x_{ij}(t))) = 0.5(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \qquad (4.1)
\end{aligned}$$

where

$$\begin{aligned}\Delta v_{nn} &= n_{kl} - n_{ij} \\ n &\equiv x(t), y(t), u \\ |x_{ij}(0)| &\leq 1 \\ |u_{ij}| &\leq 1 \\ |z_{ij}| &\leq z_{\max} \\ 1 \leq i \leq M \quad &, \quad 1 \leq j \leq N\end{aligned}$$

where x_{ij}, u_{ij} and y_{ij} are the state, input and output voltages of the specific CNN cell, respectively. The state and output vary in time, the input is static (time-independent), ij refers to grid point associated with a cell on a $2D$ grid, and $kl \in N_r$ is a grid point in the neighborhood within a radius r of the cell ij. Term $A_{ij,kl}$ represents the linear feedback, $B_{ij,kl}$ is the linear control, $D_{ij,kl}$ represents the nonlinear template and z_{ij} is the cell current (also referred to as bias or threshold), which could be space and time variant. A CNN cloning template, the program of the CNN array, is given with the linear and nonlinear terms completed by the cell current. The block diagram of a cell $C(i,j)$ is shown in the Figure 4.3. The piecewise-linear function $f(x_{ij}(t))$ is a widely used nonlinearity, see Figure 4.4.

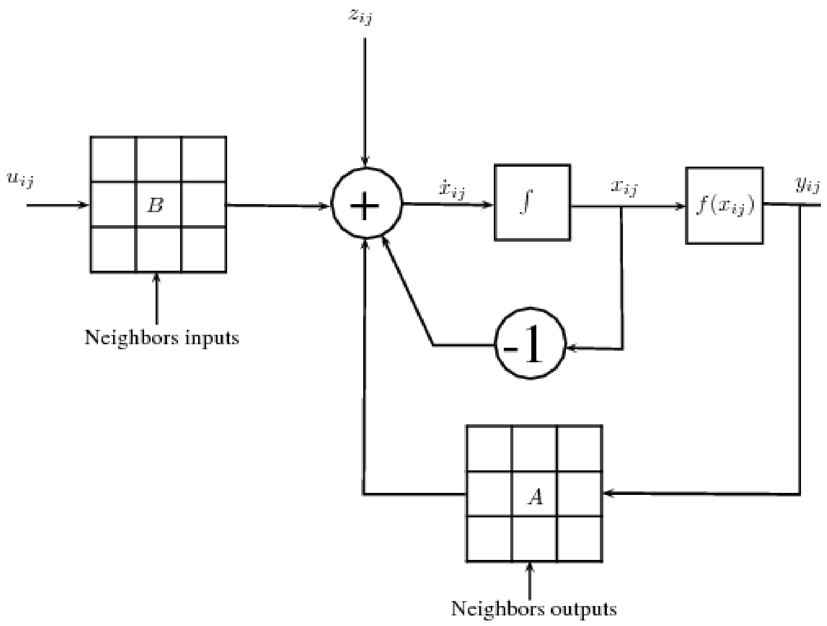


Figure 4.3: A block diagram representing CNN

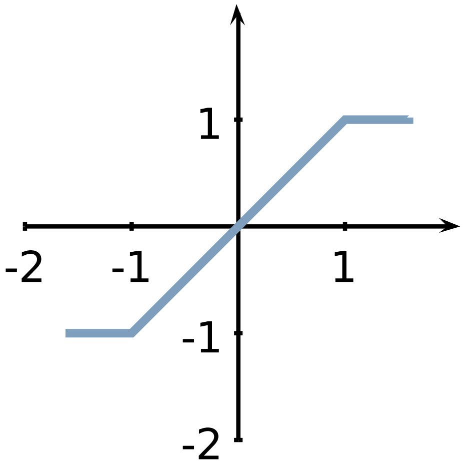


Figure 4.4: A piecewise-linear function, CNN cell output function

The bias (also referred to as the **bias map**) of a CNN layer is a gray-scale image. The bias map can be viewed as the space variant part of the cell current. Using pre-calculated bias maps, the linear spatial adaptivity can be added to the templates in CNN algorithms. If the bias map is not specified it is assumed to be zero.

Any cell that belongs to a neighborhood of any cell yet not part of $N \times M$ grid is called virtual cell. These cells must be initialized by one of the boundary conditions, which are most commonly used for a 3×3 neighborhood.

1. Fixed (Dirichlet) boundary conditions.

2. Zero-flux (Neumann) boundary conditions.

3. Periodic (Toroidal) boundary conditions.

4.1.3 Global Behavior of Cellular Neural Networks

In image processing applications, a $M \times N$ rectangular grid array usually represents the image, where M and N are the number of rows and columns, respectively. Each cell in a CNN corresponds to a pixel in the image.

With local connectivity and the space invariant assumptions, the template set, which contains 19 coefficients (A, B, z), represents the program of the neural network. The behavior of the CNN is completely determined by these set values.

To define these programs, several methods have been utilized. New template can be specified by defining the global task's rules, local rules or by training algorithms. Usually, the local rules are used to specify the equilibrium state of a cell based on the inputs and outputs in the local neighborhood. The inputs and the outputs of the neighbor cells are assumed to be constant. The dynamics of the cell is not specified. Simulation, usually, is a useful method to test the dynamic global behavior of the entire clone of cells.

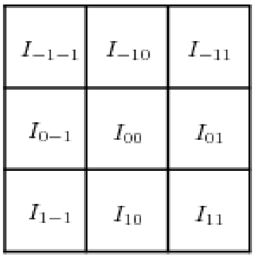
Optimal coefficient calculation leads to solutions, which converge after a short time. That is, the output of every cell reaches its final output after a specific short time.

4.2 Nonlinear Diffusion Filtering

4.2.1 Introduction

The importance of approximation and noise filtering comes from its implication on the image analysis and computer vision applications. Its target is to replace the image by its smother version by removing undesired data that may complicate the image processing steps. Dealing with the noise has evoked a lot of research over the years [Mitra & Sicuranza, 2000; Pitas, 1993, 2000; Pitas & Venetsanopoulos, 1990; Yaroslavsky & Eden, 1996]. Recently, nonlinear techniques have attracted an increasing attention in order to process both the impulsive and the Gaussian noises. Furthermore, it is inevitable to attenuate the corrupted pixels to facilitate other image processing operations such as edge detection and image segmentation.

In the literature, there is a considerable number of techniques that have been used to filter an image. Regardless of the detailed algorithms of these techniques, the underlying idea is to preserve edges and other important details, and to eliminate different kinds of noise. In particular, the crucial importance of the edges' information for human perception makes their preservation and enhancement very important for judging the performance of image filters, both linear and nonlinear.



I_{-1-1}	I_{-10}	I_{-11}
I_{0-1}	I_{00}	I_{01}
I_{1-1}	I_{10}	I_{11}

Figure 4.5: The filtering mask of size 3 × 3 with the pixel I_{00} in the center

The earliest and the most simple methods are the linear filters. Basically, these filters are based on the convolution of the image with the filter kernel of constant coefficients.

Thus, with a 3 × 3 window in Figure 4.5, the filter will replace the central pixel value I_{00}, from the set of pixels I_{xy} ($x = -1, 0, 1;\ y = -1, 0, 1$), with the weighted average of the gray-scale values of the central pixel I_{00} and its neighbors [González & Woods, 2008; MCDONNELL, 1981]. The result of the convolution I^*_{00} of the kernel A is

$$I^*_{00} = \frac{1}{Z} \sum_{x=-1}^{1} \sum_{y=-1}^{1} A_{xy} I_{xy}, \qquad Z = \sum_{x=-1}^{1} \sum_{y=-1}^{1} A_{ij} \tag{4.2}$$

Although most linear filters are simple and remarkably fast, they cause blurring to the edges throughout the image. This drawback can be overcome by introducing an appropriate adaptive nonlinear filter kernel. The nonlinear filters are usually applied on a selected neighborhood. The adaptivity refers to the fact that the coefficients' values are set according to the pixels' values in every neighborhood to be smoothed [Erler & Jernigan, 1994; Haykin, 2001]. Therefore, adaptive filtering can be considered as a nonlinear process that attenuates the noise while preserving the important image's features such as edges.

A powerful adaptive filter has been proposed in [Saint-Marc *et al.*, 1989, 1991]. In this technique, which is similar to the anisotropic diffusion, the central pixel I_{00} is replaced by a weighted sum of all the pixels contained in the filtering mask

$$I^*_{00} = \frac{1}{Z} \sum_{x=-1}^{1} \sum_{y=-1}^{1} W_{xy} I_{xy} \tag{4.3}$$

where

$$W_{xy} = \exp\left(-\frac{|G_{xy}|^2}{\beta^2}\right) Z = \sum_{x=-1}^{1} \sum_{y=-1}^{1} w_{xy} \tag{4.4}$$

Here $|G_{xy}|$ is the magnitude of the gradient calculated in the local neighborhood of the pixel I_{00} and β is a smoothing parameter.

Anisotropic diffusion is a powerful filtering technique. It has been proposed by Perona and Malik [Perona & Malik, 1990] in order to selectively enhance the image required details and reduce noise using a modified heat diffusion equation coupled with the concept of scale space [Witkin, 1983]. Since then, Anisotropic diffusion became a popular tool for a wide range of applications, for example, medical image processing [Bajla *et al.*, 1993; Gerig *et al.*, 1992; King & Glick, 1993; Lamberti *et al.*, 2002; Loew *et al.*, 1994; Sijbers *et al.*, 1997; Steen & Olstad, 1994], improved sub-sampling algorithms [Ford *et al.*, 1992], post-processing of fluctuating [Weickert, 1996], blind image restoration [Kaveh, 1996], computer-aided quality control [Weickert, 1995], segmentation of textures [Whitaker & Gerig, 1994; Whitaker, 1993], remotely sensed data [Acton & Crawford, 1992; Acton *et al.*, 1994], multigrid methods [Acton, 1998], mathematical morphology inspired techniques and many others [Biswas, 1996; Fischl & Schwartz, 1997; Gerig *et al.*, 1992; Kimia & Siddiqi, 1994; Maeda *et al.*, 1998; Shah, 1996; Torkamani-Azar & Tait, 1996].

4.2.2 Isotropic Diffusion

Basically, the anisotropic diffusion is developed, based on the modification of isotropic diffusion (4.6), in order to prevent blurring effect at the image edges. This modification is done by introducing a conductivity coefficient function that encourages the intra-region smoothing over inter-region smoothing.

Physically, we can define diffusion as a transport process that tends to compensate concentration differences. Thus, it leads to equalization of the spatial concentration differences. Fick's law of diffusion states that flux density Γ is directed against the gradient of concentration I in a given medium $\Gamma = -c\nabla I$, where c, the diffusion coefficient, describes the relation between the gradient and the flux. In the isotropic case, the coefficient function is replaced by a positive scalar-valued. The transformation process can be expressed by using the continuity equation and applying Fick's law

$$\frac{\partial I}{\partial t} + \mathrm{div}\Gamma = 0, \;\Rightarrow\; \frac{\partial I}{\partial t} = \mathrm{div}\left[c\nabla I\right] \tag{4.5}$$

where t is the time. This equation represents many physical transport processes [Crank, 1975], including heat transfer where it is called heat equation. Having a space independent and constant scalar coefficient, the technique will be homogeneous filtering and it is called isotropic diffusion. On the other hand, the space-dependent diffusion coefficient produces an inhomogeneous filtering, also called anisotropic diffusion.

Assuming that $I(x, y, 0)$ is the initial state, if $I(x, y, t)$ denotes a real valued function representing the digital image in 2D, and the grey value at any point represents the concentration, then the equation of linear and isotropic diffusion can be written as

$$\frac{\partial I(x,y,t)}{\partial t} = c\left[\frac{\partial^2 I(x,y,t)}{\partial x^2} + \frac{\partial^2 I(x,y,t)}{\partial y^2}\right] \tag{4.6}$$

where x, y are the image coordinates, t denotes time, $c = 1$ is the diffusion coefficient.

Isotropic diffusion is a well established filtering technique. Although Witkin [1983] paper is considered as the first study to discuss this technique, there is, however, a previous study that considered isotropic diffusion [Iijima, 1962] (in [Weickert, 1998]). More detailed investigations can be found in [Lindeberg, 1993; Weickert *et al.*, 1997].

The remarkable properties of the isotropic diffusion have been reflected by the vast applications of this technique. However, two main disadvantages have to be dealt with. Firstly, since isotropic filter does not consider the natural boundaries of the objects, it does not incorporate the semantically meaningful descriptions of the image which help toward achieving the desired target [Perona & Malik, 1990] such as edge detection. The isotropic diffusion filter, in addition to the noise filtering, destroys the edge junctions which contain the spatial information of the edges drawing. Secondly, the dislocation of the edges at the coarse scale [Bergholm, 1987; Witkin, 1983] needs further analysis for the resultant structure.

4.2.3 Anisotropic Diffusion

In order to tackle blurring and localization issues of isotropic diffusion, Perona & Malik [1990] suggested that conductivity coefficient c should be dependent on the image structure such that the diffusivity at large gradient locations, most likely edges, is minimized. If $\Omega \in \mathbb{R}^2$ is a rectangular image domain, the image $U(x, y) : \Omega \to \mathbb{R}$. Therefore, for the filtered image $I(x, y, t)$, they proposed the following partial derivative equation (PDE)

$$\frac{\partial I(x, y, t)}{\partial t} = \operatorname{div}\left[c(x, y, t)\nabla I(x, y, t)\right] \tag{4.7}$$

where

$$I(x, y, 0) = U(x, y) \tag{4.8}$$

The diffusion coefficient $c(x, y, t)$ is a monotonically decreasing function of the image gradient magnitude and usually contains a free parameter K, which determines the gradient threshold value which marks the amount of smoothing introduced by the non-linear diffusion process. Many functions of $c(x, y, t)$ have been suggested in the literature [Acton, 1998; Alvarez *et al.*, 1992b; Black *et al.*, 1998; Charbonnier *et al.*, 1994; Guillermo Sapiro, 2001; Rudin *et al.*, 1992; ter Haar Romeny, 1994]. The most popular functions are those introduced in [Perona & Malik, 1990], i.e.,

$$c(x, y, t) = \exp\left(-\left(\frac{|\nabla I(x, y, t)|^2}{K^2}\right)\right) \tag{4.9}$$

$$c(x, y, t) = \frac{1}{1 + \left(\frac{|\nabla I(x,y,t)|}{K}\right)^2} \tag{4.10}$$

Note that $c(x, y, t)$ is a time and space-varying function, and its value is large in homogeneous regions to encourage smoothing and small at edges to preserve image structures.

One of the major practical drawbacks of the anisotropic approach is that, even though many methods have been proposed to estimate the parameter K value [Kim, 2006; Perona & Malik, 1990; Szatmari *et al.*, 2000; Voci *et al.*, 2004; Yi *et al.*, 2005; Zhang *et al.*, 2007], the optimal value of this parameter is unknown. Also, some stopping criterion is required to finish the iteration process before the image converges to a staircase solution or to the average value of the image pixels [Weickert, 1998; You *et al.*, 1996].

Furthermore, the properties of the diffusion function make it difficult to differentiate between the semantically important edge and the impulsive noise with large gradient that might contaminate the image. Increasing the value of K in (4.9) and (4.10), to get rid of this type of noise will increase the blurring effects on the edges, which is the main problem that leads to anisotropic diffusion in the first place. In our approach, to deal with these issues, many components have been added to the algorithm.

In addition to these practical problems, there is a theoretical problem regarding the PDE solution: ill-posedness. That is, the solution of (4.7) should have a unique solution that depends on the initial condition. The main manifestation of this problem is the appearance of so-called staircase effect. However, Weickert & Benhamouda [1997] has proven that using spatial finite difference discretization is sufficient to transfer the Perona-Malik equation (4.7) into a well-posed system. However, to eliminate the numerical implementation effect, regularization has been added to (4.7) in [Catte *et al.*, 1992; Nitzberg & Shiota, 1992]. Thus, In order to improve the efficiency of the original process, a regularization was introduced [Alvarez *et al.*, 1992b; Catte *et al.*, 1992]. This addition makes the original equation have a unique solution and, apparently, more robust against noise. In this method, the diffusion coefficient is a function of the gradient convolved with a Gaussian linear filter G_σ with standard deviation $\sigma > 0$. Therefore, the new equation is

$$\begin{aligned} \frac{\partial I(x,y,t)}{\partial t} &= \operatorname{div}\left[\hat{c}(x,y,t)\nabla I(x,y,t)\right] \\ \hat{c}(x,y,t) &= g\left(|\nabla(G_\sigma * I(x,y,t))|\right) \\ G_\sigma &= G_0\frac{1}{\sqrt{\sigma}}\exp\left(-\frac{|z|^2}{4\sigma^2}\right) \end{aligned} \tag{4.11}$$

where G denotes the Gaussian kernel with standard deviation σ, $*$ denotes the convolution and g is a decreasing function. The advantages of this formulation are that it can enhance the edges [Weickert & Benhamouda, 1997], reduce the staircase effect [Nitzberg & Shiota, 1992] and it is less prone to discretization effect [Froehlich & Weickert, 1994] (in [Weickert, 1998]). Yet, this technique leads to a higher computational complexity of the anisotropic diffusion process.

Other approaches have been proposed in which robust statistic norms were chosen to design the anisotropic diffusion process [Black *et al.*, 1998; Scharr *et al.*, 2003; You *et al.*, 1996]. In these methods, the diffusion coefficient function preserves the edges and improves the automatic stopping of the diffusion. However, these diffusion coefficient functions are not effective in case of strong Gaussian or impulsive noise.

4.2.4 Complex Diffusion

Based on simplified Schrödinger equation, a generalization of linear and nonlinear diffusion in the complex domain has been proposed in [Gilboa *et al.*, 2001, 2004]. The underlying idea is to facilitate the complex diffusion-type processes and the time-dependent Schrödinger equation from the quantum mechanics. In the simplest case, the Schrödinger operator is

$$H = -\frac{\hbar^2}{2m}\Delta + V(x) \tag{4.12}$$

where x is the spatial coordinates, $\hbar$ is Planck's constant, $V(x)$ is the external field potential and Δ is the Laplacian operator. The relation between the diffusion equation and Schrödinger equation has been investigated in [Nagasawa, 1993].

Gilboa *et al.* [2001] considered the following problem

$$\begin{aligned} \frac{\partial I(x,y,t)}{\partial t} &= c\nabla^2 I, \qquad t > 0,\ x,y \in \mathbb{R} \\ I(x,y,0) &= I_0 \in \mathbb{R}, \qquad c, I \in \mathbb{C} \end{aligned} \tag{4.13}$$

In order to solve the generalized equation (4.7), with the same conditions, they utilize the Laplacian operator to produce the diffusion coefficient function. The second derivative (Laplacian) operator is a suitable choice since it has a high magnitude near the edges and low magnitude everywhere else. Therefore, it enables the nonlinear diffusion process to reduce noise within a ramp. The system equations can be written as

$$\begin{aligned} \frac{\partial I(x,y,t)}{\partial t} &= \operatorname{div}(c(\Im(I)\nabla I) \\ c(\Im(I)) &= \frac{e^{i\theta}}{1 + \left(\frac{\Im(I)}{K\theta}\right)^2} \end{aligned} \tag{4.14}$$

where $i = \sqrt{-1}$, K is a threshold parameter. The phase angle θ should be small ($\theta << 1$), and $\Im(I)$ is the imagery part of I.

4.3 Microarray Image

To summarize the idea presented in this chapter, we have the following points. 1) Microarray image processing is an important yet challenging problem due to the system imperfections and microarray generating process, which restricts both our ability to differentiate between spots' signals and background signal and our ability to get accurate measures of interest in the images. 2) Instead of using the raw image, it makes much more sense to produce filtered versions of the image data by applying nonlinear anisotropic diffusion so that the dynamic range of the image could be increased, thereby achieving a better ability of signal extraction. 3) The ability to transform images, in complex and non-linear ways, makes CNNs ideal for microarray image processing. 4) Although many approaches have been proposed, there has been little progress in developing effective algorithms that automatically clarify an image's features by using up-to-date techniques such as integrated diffusion filtering and CNNs approach. It is, therefore, the aim of this chapter to investigate the microarray image analysis by implementing filtering techniques to reduce the microarray images' noise and, at the same time, to enhance the position of gene spots.

Motivated by the above discussions, in this chapter, novel segmentation algorithms are investigated. These algorithms are based on the CNN computational paradigm combined with median and anisotropic diffusion filters. The proposed methods are applied to a set of real-world cDNA images. Quantitative comparisons are carried out among different filters in terms of objective and subjective evaluation methods. It is shown that the proposed algorithm with complex diffusion filter surpasses other methods as well as the output of GenePix software. The presented algorithms can also be applied on CNN-UM [Chua & Roska, 1993b]. These algorithms, even when applied on GPU [Dolan & DeSouza, 2009; Ho *et al.*, 2008; Soos *et al.*, 2008], offer a view of parallel computation that remains reasonably efficient.

4.4 The Framework

In order to demonstrate the proposed algorithm, a standard model of microarray image analysis is followed, see Figure 4.6. The test dataset features many of the characteristics that hinder traditional segmentation approaches. These characteristics include poorly expressed spots, malformed spots, high valued noise artefacts and uneven background. With such a test dataset, the effectiveness, efficiency and robustness of the algorithm can be evaluated.

To get the preliminary coordinates of the spots, the robust girdding algorithm proposed

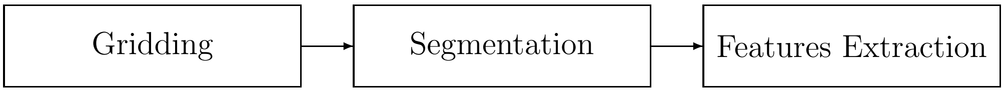


Figure 4.6: Microarray image analysis

by Morris [2008] is used. When the girdding information is ready, each Region Of Interest (ROI) is put to the two-stage process. In the first stage (see Algorithm 1), the filtering stage is carried out by applying different versions of Anisotropic Diffusion techniques. Ideally, the output of this stage would be the best candidate as input for the segmentation stage. However, analyzing the performance of different diffusion functions has led to some conclusions and future recommendations to be discussed later. In the second stage (see Algorithm 2), the intermediate output is fed into the segmentation stage. A novel CNN algorithm is then proposed and performed using MatCNN Matlab toolbox from AnaLogic Computers Kft. The algorithm can be applied on CNN-UM.

4.4.1 Gridding

A completely blind microarray image gridding framework [Morris, 2008] is used to accomplish the spotting task. The input of the framework is the microarray image that can be at any resolution, and the gridding is accomplished without any prior assumptions. The framework includes an Evolutionary Algorithm (EA) and several methods for various stages of the gridding process, including sub-grid detection.

Remark 1. *The output of this stage is the coordinates of all ROIs, i.e., the region of every spot in the input image. Therefore, the input of the later steps will be the microarray raw image and these coordinates .*

4.4.2 Segmentation Algorithm

Incorporating the global information is very important for successful segmentation algorithms. However, the objective of our application is a two-level segmentation. Therefore, it can be assumed that the global image properties are irrelevant and, rather, a locally adaptive strategy with less computational complexity can meet the application requirements. It is important to note that using local information leads to a robust and reliable segmentation algorithm in some applications such as microarray image analysis.

Algorithm 1 The Main Body Algorithm of the proposed approach

```
Require: $C_y3$ {Image: The green channel}
Require: $C_y5$ {Image: The red channel}
Require: $Gridd$ {Rough spots' coordinates}
Ensure: $I_o$ {Image: the output MASK}
while There are more spots to be processed do
    $tG \leftarrow \text{median}(G_i,\ 5)$
    $tR \leftarrow \text{median}(R_i,\ 5)$
    $tG \leftarrow \text{diffusion}(tG)$
    $tR \leftarrow \text{diffusion}(tR)$
    $tg \leftarrow \text{median}(tG)$
    $tG \leftarrow tG - tg$
    $tr \leftarrow \text{median}(tR)$
    $tR \leftarrow tR - tr$
    $mdg \leftarrow \text{median}(tG)$
    $mdr \leftarrow \text{median}(tR)$
    $mng \leftarrow \text{mean}(tG)$
    $mnr \leftarrow \text{mean}(tR)$ {Check the noise degree in the current ROI} {produce one
    representative image}
    if $mdg > mng || mdr > mnr$ then
        $I \leftarrow tG + \frac{mdg}{mdr} tR$
    else
        $I \leftarrow \max(tG,\ tR)$
    end if
    $I \leftarrow \text{AdSeg}(I)$
    $I_o i \leftarrow I$
    return $I_o$
end while
```

Algorithm 2 AdSeg Algorithm: Adaptive segmentaion steps

```
Require: $I$ {Image: specify gene spot region pixels}
Ensure: $I_o$ {Image: the output mask}
$tI \leftarrow \text{mean}(I, 5)$
$mdI \leftarrow \text{median(I)}$
$mnI \leftarrow \text{mean(I)}$
$\Theta \leftarrow tI - \max(mdI,\ mnI)$ {Estimate the threshold}
$tI \leftarrow \text{THRESH}(I,\ \Theta)$ {Adaptive threshold}
$tI \leftarrow \text{checkSize}(tI,\ \alpha,\ \beta)$ {Eliminate objects that are smaller than $\alpha$ or bigger than
  $\beta$}
$tI \leftarrow \text{checkHoles}(tI)$ {Eliminate objects that are with holes}
$I_o \leftarrow \text{checkGridd}(tI)$ {Eliminate objects that ovarlaps the borders}
```

The algorithm, illustrated in Figure 4.7, consists of five stages. 1) In the pre-filtering stage, every channel (the red and the green) is put to three processing steps. First, a basic median filter with window's size (5 × 5) is applied, followed by further smoothing stage, which could be either diffusion filtering or unitary operation (the output of the previous stage transferred to the next one). Then, median filter is used to estimate the background level of the ROI. 2) The red and the green channels are combined to produce a gray-scale image. 3) The gray-scale image is used to estimate the local thresholds. 4) Locally adaptive segmentation is applied on the gray-scale image taking into account the estimated threshold. 5) A series of binary operation is applied in order to remove the small and the big objects (based on a specified threshold), i.e., the objects that have holes and the objects that overlap the edge of the ROI.

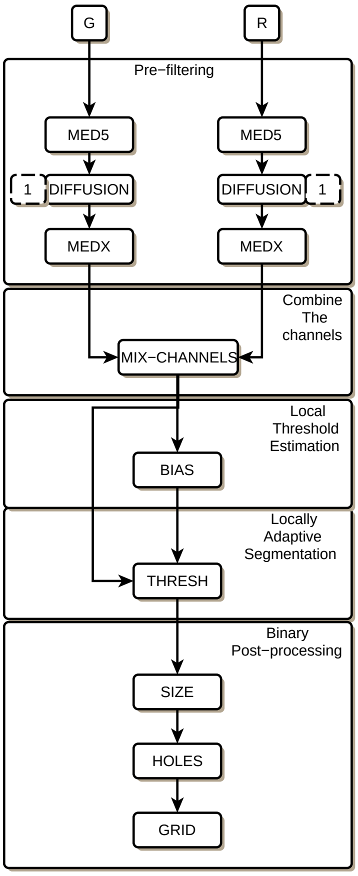


Figure 4.7: Microarray segmentation algorithm

4.4.2.1 Median 5×5

The fundamental property of the median filter is its ability to eliminate impulsive noise. Basically, this filter replaces the value of the current pixel, assumed corrupt, by the median of the ordered input sample (values of the window that centered by current pixel). Median filter has been extensively investigated and extended into many different versions.

The median filter, as a nonlinear filter, is able to attenuate the strong impulsive noise and, at the same time, preserve image edges. However, this filter will affect the objects that have the size similar to or smaller than the size of the filter window, thus it distorts the texture of the filtered image. On the other hand, median filter, in addition to the noisy areas, will affect the noise free areas since it does not differentiate between noisy and noise-free pixels. Therefore, many methods have been proposed to accomplish the balance between attenuation of the noise and the preservation of image fine details, i.e., a trade-off between the suppression of noise and preservation of minute details [Alparone *et al.*, 1996; Arce, 1998; Arce *et al.*, 1986; Chen, 2001; Ko & Lee, 1991].

A typical microarray image contains several kinds of artefacts and noise (hair, scratches and fingerprints, for example). If the operator of traditional microarray image analysis software is to be removed from the analysis process, the input images must be cleaned (have noise artefacts removed). However, this cleaning process should not affect the gene spot intensities themselves as later stages will use such information to help determine the spots' locations.

The first median filter simply parses the red and the green channels independently with a sampling window of 5×5 pixels centered on each pixel in turn. This center pixel is thus calculated as the median of all the pixels in the 5×5 region. The 5×5 region of the median filter process effectively removes the small artifacts on the array. Since a hybridized spot is much larger than the smoothing window, the filter will reserve the overall structure of the spot.

Remark 2. *The smoothing process can remove the small region with high intensity pixels that are of no value in a gene spot's identification context. Therefore, applying a smoothing operator as a first stage will be of additional benefit.*

4.4.2.2 Anisotropic Diffusion

It is generally accepted that images contain structures at different scales. Practically, it is usually ambiguous to specify the right scale to obtain the desired information. Therefore, it is beneficial to have an image representation at multiple scales [Alvarez *et al.*, 1993]. A multi-scale representation of an image is an ordered set of derived images intended to represent the original image at various levels of scale [Bovik, 2000]. Having these structures eases the image's processing in later stages.

Basically, the Gaussian representation introduces a scale dimension by convolving the original image with a Gaussian noise with a standard deviation $\sigma = \sqrt{2t}$. This is analogous to solving the linear diffusion equation:

$$I_t = c\nabla^2 I, \quad I|_{t=0} = I_0, \quad 0 < c \in \mathbb{R} \tag{4.15}$$

with a constant diffusion coefficient $c = 1$.

A major breakthrough comes from Perona & Malik [1990] who proposed anisotropic diffusion for adaptive smoothing in order to formulate the problem in terms of the non-linear heat equation. The main benefit of the anisotropic diffusion is the edge preservation, which is achieved by the introduction of the coefficient function $c(\mathbf{x})$ [Weickert, 1998]. This function encourages the intra-region smoothing over the inter-region smoothing [Bovik, 2000]. If $c(\mathbf{x})$ is allowed to vary according to the local image gradient, then we have an anisotropic filter. Here is a basic anisotropic diffusion PDE:

$$\frac{\partial I_t(x)}{\partial t} = div\{c(x)\nabla I_t(x)\} \tag{4.16}$$

with $I_0 = I$ [Bovik, 2000]. Recently, Gilboa *et al.* [2001] generalized the linear and the non-linear scale spaces to the complex diffusion processes by combining the diffusion and the free Schrödinger equation.

In this chapter, three diffusion coefficient functions are used, that is, 'Diff1' [Perona & Malik, 1990], 'Diff2' [Perona & Malik, 1990] and 'CDiff' (complex valued - ramp preserving Gilboa *et al.* [2001]), denoted as follows, respectively,

$$c = \exp(-(\frac{|\nabla(J)|}{K})^2) \tag{4.17}$$

$$c = \frac{1}{1 + (\frac{|\nabla(J)|}{K})^2} \tag{4.18}$$

$$c = \frac{\exp(i * \theta)}{1 + (\frac{Im(\nabla(J))}{K * \theta})^2} \tag{4.19}$$

where K is a threshold parameter and Im is the imaginary part of a complex number. The phase angle θ should be small ($\theta \ll 1$).

4.4.2.3 The Median of ROI

The second median filter is essentially the same as the first except that the second filter uses a larger window region. In this case, the window is slightly bigger than a gene spot local region. This filter results in a simple estimation of the image's background features. Such smoothing operators have two positive effects on the image data: 1) low-level background noise is either reduced substantially or removed altogether from the image; and 2) the large scale artefacts have their internal structure been removed from the image data.

4.4.2.4 Two Channels to One Channel

By applying the pre-filtering operators, the 'salt and pepper' type artefacts and the large artefacts regions have been attenuated or completely removed. In order to proceed with the segmentation process, a single combined view has to be created by merging the processed information of both channels. Hence, the computational complexity would be reduced. The merging approach is used to reduce the effects of the artefacts and get the most essential information for the later stages. The mean and median of the ROI for every channel are used to estimate the quality of the area.

If $median(I) > mean(I)$, where I is the green or the red channel, then

$$J = GREEN + (\frac{MED(GREEN)}{MED(RED)}) * RED \tag{4.20}$$

otherwise

$$J = \max(GREEN, RED) \tag{4.21}$$

Remark 3. *The median and mean values of both channels are used to specify the degree of contamination in each one of them. Investigating the dataset gives rise to the formulation of this relationship. (4.20) is used to exploit the most available information in both channels and (4.21) is used when at least one of the channels is less contaminated. Hence, the effect of the noisy channel can be reduced as much as possible.*

4.4.2.5 Local Threshold Estimation

The threshold estimation can be described by the following equation:

$$\Theta = T_r(U) = \mu_r(U) - \max(MED(U), MEAN(U)) \tag{4.22}$$

where $\mu_r(U)$ is the image which is composed of the mean in local neighborhood N_r, U is the input image, Θ is the threshold estimate which is used as a bias map of the CNN.

(4.22) defines space-variant threshold levels. This information should support an optimal separation of the objects from the background [Sezgin & Sankur, 2004; Venkateswarlu & Boyle, 1995].

4.4.2.6 Locally Adaptive Segmentation

The segmentation process performs the following gray-scale to binary mapping:

$$Y' = S_r(Y, \Theta) \tag{4.23}$$

where Y is the gray-scale image, Θ is the space-variant threshold level (local threshold estimation) and Y' is the binary output of the mapping. S_r compares the image to the threshold in the local neighborhood N_r and specifies the binary output.

$$ADTHRES_A = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 2.0 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}; \quad ADTHRES_B = 0; \quad ADTHRES_I = 0.7; \tag{4.24}$$

CNN is used to achieve the adaptive segmentation with a single template operation `ADTHRES` (4.24), which gives the binary segmentation output.

4.4.2.7 SizeClass

In this stage, two CNN templates are used to remove the small objects from the output image of the previous step. The morphological operator "EROSION" is applied to peel off from the input image one layer of boundary pixels every time. The algorithm tends to remove any object with 5 pixels' diameter or less. Then, the reconstruction operation, which uses the so-called "RECALL" template, is used. The later template operates on two images: 1) the segmentation result contains the objects which are regarded as a spot signal; and 2) the output of the "EROSION" operator contains the markers. where n is

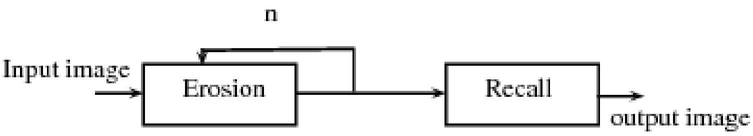


Figure 4.8: SizeClass algorithm

the number of times the EROSION operator will be applied and, thus, it is the number that specifies the smallest size for detected objects.

Remark 4. *In the marker's image, if at least one active pixel of an object survives the peeling process, the whole object will be restored.*

4.4.2.8 HoleClass

CNN template, for connected component detection [Matsumoto *et al.*, 1990b], is used to detect any holes or irregular object results from the artefacts.

4.4.2.9 GridClass

Assume that the spots do not overlap the boundary of the ROI (the spot should roughly be in the middle area). In this case, it is essential to treat any segmentation result that overlaps the borders as a noise. Two CNN templates are used to this end: 1) "LOGAND" operator is applied with two input images, the first one is the output of the previous step and the second is an empty image with active boundary pixels; and 2) the reconstruction operator is applied ("RECALL" operator) to get an image with an object assumed to be the spot's signal.

4.4.3 Features' Extraction

There are many methods to separate the background in order to calculate the statistics of the spots and their local background. The most popular one is that associated with GenePix. This method uses the valleys' regions (the center of four gene spots' regions) as a background of the central circular gene spot. However, using these valleys may lead to incorrect representation of the background. Another popular method is the circular background region around the spot, which adopted by ImaGene. The background is specified by a circular region around the gene area with a gap between both. Yet, the main drawback of this method is that one has to specify the size of the gap which may result in inaccurate measures.

By regarding the whole ROI (except the spot's pixels) as the background, the true background distribution should be represented more rigorously. The intense variations in a microarray image background make the last method the representative of a more global background calculation. Therefore, it is our choice to facilitate the last approach in order to separate the background and to calculate the required statistic measurements.

4.5 Results and Evaluation

In order to quantify the performance capabilities of the different techniques, a quality measure is required to allow the judgment of how well the calculated template fits the gene's spot position. Every investigated technique produces a mask that classifies the pixels as belonging either to signal (the gene spot) or to noise (the local background). For this purpose, 1) two subjective quality measures are used. The first is known as PSNR, Section 3.7, and the second is the k-means's objective function; and 2) a systematic objective method, which is based on the descriptive statistic Interclass Correlation Coefficient (ICC) measures, is used in order to compare the results produced by different techniques. The rational is justified as follows.

Remark 5. *Throughout the discussion the algorithm has been given the following four names to highlight four integrated filtering techniques that are used in the pre-processing stage: 1) Median method where only the median filter is used; 2) Diff1 method where the anisotropic diffusion filter with (4.17) is used as diffusion coefficient function; 3) Diff2 method where the anisotropic diffusion filter with (4.18) is used as diffusion coefficient function; and 4) CDiff method where the complex diffusion filter with (4.19) is used as diffusion coefficient function.*

4.5.1 The Peak Signal-to-Noise Ratio (PSNR)

From Figure 4.9, we directly compare PSNR values determined by GenePix and the proposed techniques for the dataset images and on average. Our algorithm shows a marked 3 to 6 dB improvement. Essentially, the algorithmic process has consistently outperformed the human expert using GenePix in terms of gene spot identification. Note that Diff1 and Diff2 methods are slightly different while CDiff outperforms all the other techniques. The performance of CDiff is related to the edge preserving characteristic of the diffusion coefficient function (4.19).

4.5.2 The k-means Objective Function

K-means [MacQueen, 1967] is one of the simplest unsupervised learning algorithms that solve clustering problems. The procedure follows a simple and easy way to classify a given dataset through a certain number of clusters (k). The main idea is to define k centroid, one for each cluster. The next step is to take each point in the dataset and associate it with the nearest centroid. When no point is pending, the first step is completed and an

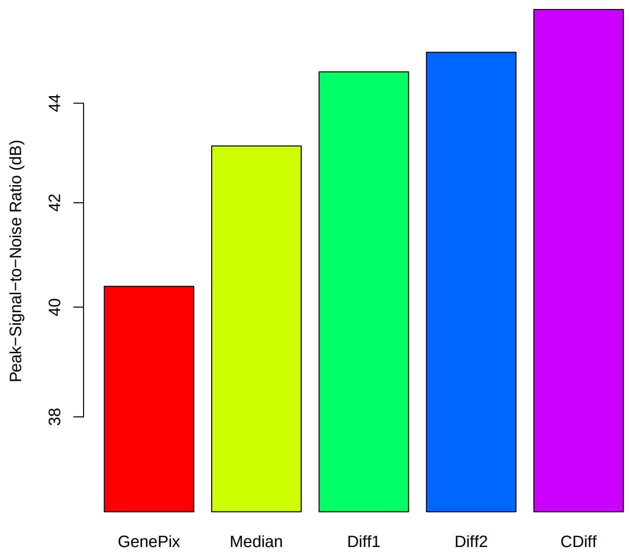


Figure 4.9: PSNR for the Dataset. Comparison of segmentation results between GenePix (GP), Median, Diff1, Diff2 and CDiff (CLD) algorithms.

early groupage is done. At this point, we need to re-calculate k new centroid (4.25) for the clusters resulting from the previous step.

$$\mu_i = \frac{1}{S_i} \sum_{x_j \in S_i} x_j \tag{4.25}$$

With these k new centroid, a new binding has to be done between the same dataset points and the nearest new centroid. Therefore, a loop has been generated. Pursuant to this loop, we may notice that the k centroids change their location step by step until no more changes is done. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function:

$$\Omega = \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{4.26}$$

where μ_i is the center of S_i.

In our case, clustering is already done with $k = 2$. Therefore, (4.25) is used to calculate the centers and (4.26) is used to evaluate the results of the segmentation process. In Figure 4.10, we can directly compare the k-means values determined by GenePix and by our techniques for the dataset images and on average. The proposed algorithms show an improvement. Essentially, the superiority of the algorithmic process over the human expert using GenePix, in terms of gene spot identification, is experimentally demonstrated.

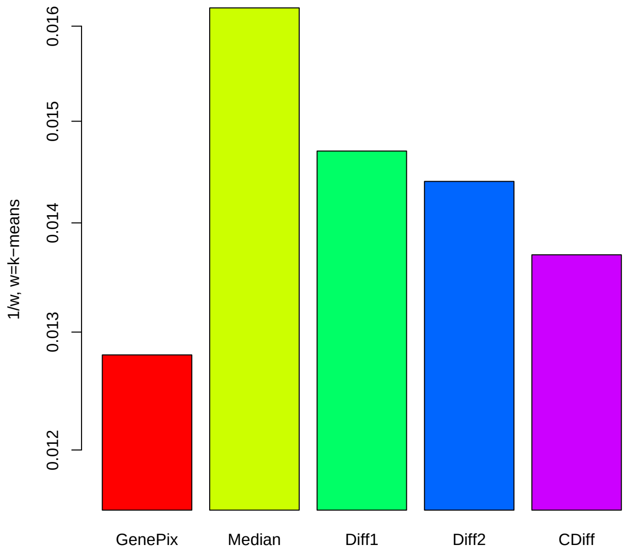
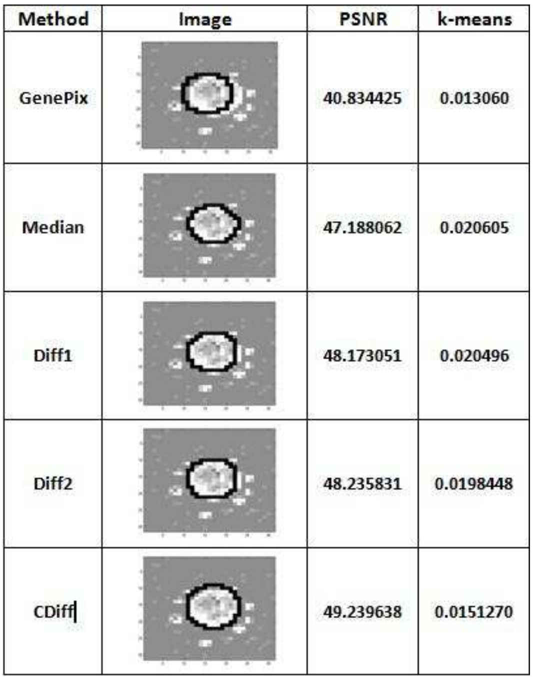


Figure 4.10: k-means for the Dataset. Comparison of segmentation results between GenePix (GP), Median, Diff1, Diff2 and CDiff (CLD) algorithms.

Figure 4.11 shows the segmentation results for one spot with the values of the k-means and the PSNR. Although using only the median filter in the smoothing stage gives the best value when comparing the segmentation results using k-means's objective function, it is better to use the diffusion filter (compare the PSNR values and the corresponding edge). We note that, with diffusion filter, especially complex diffusion, we reduce the chance of getting false negative segmentation (losing valuable information of the spot signal).

4.5.3 The Objective Comparison

Although we have compared the segmentation processes using subjective methods, it is now essential to compare them using an objective method. Due to the difficulties of comparing the estimated expression levels from cDNA microarrays by using the gold standards such as Northern blot analysis or quantitative PCR, we decide to achieve the objectivity by using the Interclass Correlation Coefficient (ICC) measures [Fleiss, 1986; McGraw & Wong, 1996]. Therefore, by comparing the results both between methods within arrays and within



Method	Image	PSNR	k-means
GenePix		40.834425	0.013060
Median		47.188062	0.020605
Diff1		48.173051	0.020496
Diff2		48.235831	0.0198448
CDiff		49.239638	0.0151270

Figure 4.11: Examples of the PSNR and k-means values for a spot

methods between replicated arrays (replicate spots on the same array in our dataset), and by assessing the observed variations relative to the variations between genes, we can objectively compare image processing methods [Korn *et al.*, 2004].

The dataset images have been obtained from experiments that carried out using Locidea Microarray ScoreCard [Anonymous, 2001]. Therefore, a specific part of the microarray image data is used in the comparison process. The microarray ScoreCard reagents consist of control spotting samples and control mRNA solutions (spike mixes). The control spotting samples have been designed to be replicated 24 times per array. Using these controls, we base our analysis on the following assumptions. 1) The better the segmentation is, the higher the correlation within the same control should be (minimum σ_e^2). 2) The better the segmentation is, the lower the correlation between the genes within the array should be (maximum σ_g^2). 3) The better the segmentation is, the higher the ICC value should be.

In order to compare the segmentation methods, we estimate the reliability of each method for each experiment using components of the variance model Korn *et al.* [2004],

$$Y_{ij} = g_i + e_{ij} \tag{4.27}$$

where Y_{ij} is the log expression ratio for the ith spot and jth replicate. The error variance component σ_e^2, associated with e_{ij}, represents the reproducibility of the method. The variance component σ_g^2, associated with g_i, represents the true spot-to-spot (gene-to-gene) variability. Then intra-class correlation Coefficient (ICC) represents the reliability of the method [McGraw & Wong, 1996]. ICC is used as a measure of reproducibility over measures such as the error variance or its square root σ_e alone, because it guards against algorithms that produce ratio estimates that all shrunk to a central value.

The variance component, which is the error within-gene and between replicates, is estimated by,

$$\hat{\sigma}_e^2 = \sum_{i=1}^{n_g} \sum_{j=1}^{n_a} (Y_{ij} - \bar{Y}_{i.})^2 / [n_g(n_a - 1)] \tag{4.28}$$

where n_a is the number of replicate arrays, n_g is the number of genes, and

$$\bar{Y}_{i.} = \sum_{j=1}^{n_a} Y_{ij}/n_a$$

The between-gene variance component is estimated by,

$$\hat{\sigma}_g^2 = \sum_{i=1}^{n_g} (\bar{Y}_{i.} - \bar{Y}_{..})^2/(n_g - 1) - \hat{\sigma}_e^2/n_a \tag{4.29}$$

where

$$\bar{Y}_{..} = \sum_{i=1}^{n_g} \sum_{j=1}^{n_a} Y_{ij}/(n_g n_a)$$

The estimated ICC is

$$ICC = \frac{\hat{\sigma}_g^2}{\hat{\sigma}_g^2 + \hat{\sigma}_e^2} \tag{4.30}$$

Figures 4.12-4.14 present the estimated variance components and ICC for the dataset images and on average. The reliabilities of all methods are high, with our methods appearing on average to be perhaps slightly more reliable than GenePix. Note that even the within-spot variability $\hat{\sigma}_e^2$ (the noise) is smaller for our method, and the between-spot variability (the signal) is bigger for our method.

Although Diff1, Diff2 and CDiff have almost the same σ_e^2 and ICC, CDiff has a bigger between-spot variability, and therefore, it outperforms the other proposed techniques as well as the human expert using GenePix in terms of gene spot identification.

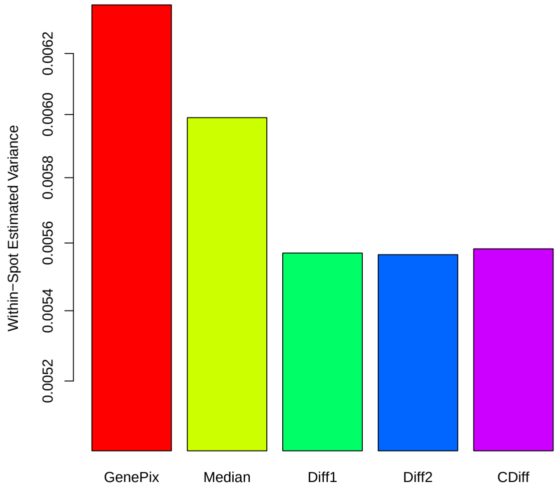


Figure 4.12: Average within-spot estimated variance over the dataset. Comparing between GenePix (GP), Median, Diff1, Diff2 and CDiff (CLD) algorithms.

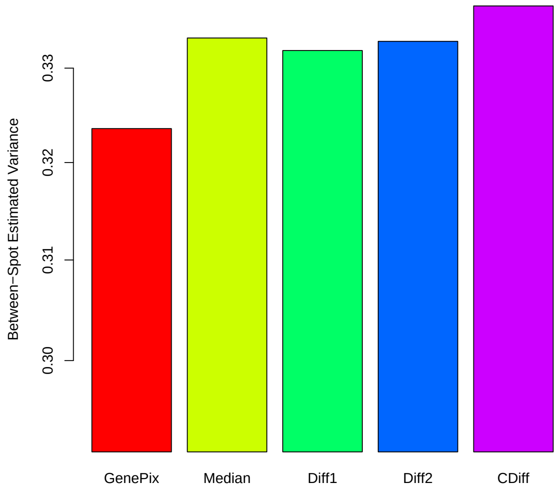


Figure 4.13: Average between spot estimated variance over the dataset. Comparing between GenePix (GP), Median, Diff1, Diff2 and CDiff (CLD) algorithms.

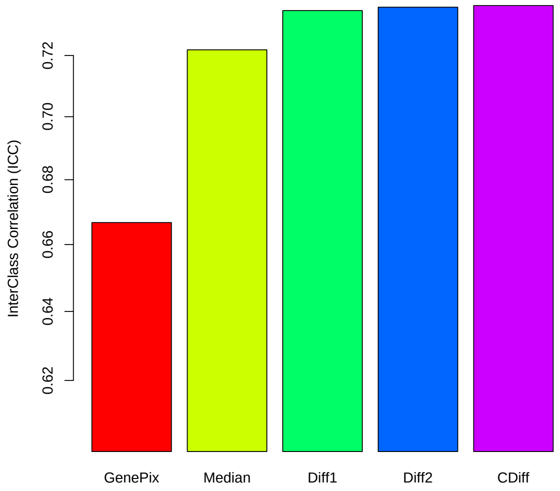


Figure 4.14: Comparing the results of the proposed adaptive segmentation algorithms. ICC over the dataset

4.6 Conclusions

In this chapter, we have presented a novel segmentation framework that attempts to improve the full workflow processing of microarray image analysis. Specifically, the framework consists of several components that process a microarray image from its raw 16-bit scanned representation to the final log ratios and related statistics without human intervention.

The proposed adaptive segmentation methodology based on Cellular Neural Networks (see Fig. 4.7) offers several advantages. In the following, we will discuss some properties of current implementations:

1. The proposed algorithm is a simple and a robust one for the segmentation of the microarray image. Our approach to analysis the image is by dividing the problem into two parts, the first part is to filter the image in a way that maps the image into a better representation for the second part. The second part, on the other hand, is to achieve the segmentation.

   However, there is another way to look at the approach. In this way, the approach can be divided into two parts. The first part is the one that deals with the low level information of the image. In other words, any process applied on the pixels' values, either to map them into new values or by classifying them into two defined categories, is considered to be a low process. On the other hand, dealing with the detected objects in order to decide whether the object belongs to the spot or merely a high noise signal is considered as a meta level process.

The robustness can be referred to both these parts. The proposed filtering techniques are well established and proved to be suitable for the real microarray image with several degrees of noise. Meta's analysis of the objects of the binary image, although it is simple, has shown a good degree of robustness. Nevertheless, many things can be done to improve this stage and this will be considered later in this thesis.

2. The algorithm is totally blind, and it takes only the raw data as the input. The importance of the blind algorithm has been discussed in Section 2.5.3 in Chapter 2. The algorithm requires no prior knowledge about the image, the intensity distribution as a global information and the noise distribution. Furthermore, no information or assumption have been made about the shape of the image or the strength of the signal in different parts on the image surface.

   The only assumptions about the spots are some general points which have been used to develop the algorithm, particularly the meta-level analysis. These general assumptions are: 1) there should be a spot somewhere in the middle of the ROI, where the gridding process has guessed the existence of this spot; 2) there should be no two spots or more within the single ROI; 3) there is, at least, a slight difference between the spot signal and the background signal with no preference on which one has the higher value.

   These assumptions are general and applied to all cDNA microarray images no matter in what machines and which laboratory they have been conducted, or which operator has carried out the experiment. However, improvements can be done to the system in general if these assumptions have been considered again to cover more situations that might arise in the real images. This is especially important since the algorithm has been developed to be adaptive.

3. Applying anisotropic diffusion filtering in the pre-processing stage improves the segmentation output and the subsequent gene expression analysis. In Chapters 2 and 3, the importance of the filtering stage has been highlighted. In this chapter, Section 4.2, the properties of nonlinear diffusion have been discussed before proving the applicability of this technique for the filtering of the microarray image.

   However, in Section 4.2.3, one particular drawback has been mentioned. The inability of the anisotropic diffusion to detect the impulsive noise was dealt with by adding one median filter before the application of diffusion filter. The median filter windows were small enough to remove the impulsive noise and, at the same time, to keep the spatial information around the edges without any significant modification. Furthermore, the bigger noise areas, with a pattern similar to a spot signal, were smoothed so the meta analysis will detect and separate them from the spot signal.

4. In the result section, we have demonstrated the potential of the algorithm using direct comparisons between our proposed approach and GenePix approach, the commercially accepted package, over a real imagery dataset. First of all, the GenePix mask (output) was achieved by a semi-automated method. Recently, another version of the GenePix has been released with the ability to carry out fully automated algorithm. Part of the future work should be testing our algorithm against the new GenePix system.

   Due to the lack of golden standard, which would enable us to evaluate the output of any algorithm dealing with the segmentation problem of the microarray image, comparison rules have to be defined in order to evaluate the relative performance of different algorithms.

   In this chapter, three methods have been used to compare the results with the GenePix segmentation output. The first is PSNR, which is usually used to test the compression algorithm. In this context, however, the argument is that PSNR is assumed to give an indication about what the difference between methods might be. The stress on indication is important, for some reasons: 1) with this method, when the area under study is noisy the ratio will increase with the size of the mask (pixels with '1' as a value); 2) when the spot signal is weak- pixels with small values- the PNSR does not give a significant difference between the applied method.

   The second comparison method is the K-means. This method outperformed PNSR by giving a better estimate for the difference between applied algorithms in most cases. However, when the spot signal has some variability the objective function will give the evaluation for the method that takes the spot partially better than the one that captures the whole spot. Finally, in order to have a full overview about the proposed algorithms, ICC method has been applied.

   In conclusion, the three methods indicate, with a slight difference, that the proposed algorithms outperformed the well-known GenePix method. In addition, the three comparison methods suggested that the complex diffusion gives a better result than the other proposed combinations.

Chapter 5

A Multi-View Analysis of Microarray Image

The discussion in this chapter based on:

Zineddin, B., Wang, Z. & Liu, X., 2010. A Multi-View Approach to cDNA Microarray Analysis. International Journal of Computational Biology and Drug Design, 3(2), 91–111.

Zineddin, B. et al., 2008. Investigation on filtering cDNA microarray image based multiview analysis. In S. Zhang & D. Li, eds. Proceeding of the 14th International Conference on Automation & Computing. London, UK: Pacilantic International Ltd., 201-206.

> *Looking for patterns trains the mind to search out and discover the similarities that bind seemingly unrelated information together in a whole.*
> *. . . A child who expects things to 'make sense' looks for the sense in things and from this sense develops understanding. A child who does not see patterns often does not expect things to make sense and sees all events as discrete, separate, and unrelated.*
>
> Mary Baratta-Lorton

In Chapter 4, We have presented a novel segmentation framework that attempts to improve the microarray image processing stage. The main part of the algorithm is the filtering components. For filtering purpose, the idea was to facilitate diffusion model in order to enhance the ability to segment the image. Three diffusion coefficient functions have been used, see Section 4.4.2.2. These functions smooth the surface of the image and preserve the edges considerably. Integrating median filter with diffusion model is advantageous since it surpasses the impulsive noise yet with a minimum edge disturbing phenomena.

This chapter is concerned with improving the processes involved in the analysis of microarray image data. The main focus is to clarify an image's feature space in an unsupervised manner. Rather than using the raw microarray image, it is suggested that producing multiple views of the image data such that emphasis is placed on certain frequencies or regions of interest would not only be advantageous but also be more effective in terms of the overall goal. In this chapter, the multi-view analysis combined with Median, Top-Hat and CLD filters is investigated. The proposed image processing methods are applied to a set of real-world cDNA images. The AnaLogic CNN Simulation Toolbox for MATLAB (InstantVision Toolboxes for MATLAB) is used during the segmentation process. Quantitative comparisons among different filters are carried out in terms of PSNR. It is shown that the CLD filter is the best one to be applied with the image transformation engine.

5.1 Introduction

The underlying principle of the microarray technology is that the measured color intensity of the hybridized mRNA gives an indication for the transcribed mRNA, i.e., the gene expression. Conducting a microarray experiment involves many steps with accumulating errors introduced in each step. As we have seen in the previous chapters, the aim of the filtering stage is to smooth the image surface and remove some small contamination. In addition, filtering might help in predicting the trend of the background. The inherent characteristics of the microarray image oblige us to use some transformation techniques. These transformations will help us getting the most of the image information for later stages.

To summarize the discussions made so far, we have the following conclusions:

1. Image processing for analysis of microarray images is an important yet challenging problem since imperfections and fabrication artifacts often spoil our ability to measure accurately the quantities of interest in the images.

2. Rather than using the raw microarray image, it makes much more sense to produce multiple views of the image data so that emphasis can be placed on certain frequencies or regions of interest.

3. Although many approaches have been proposed in the literature, the requirement of multidisciplinary knowledge resulted in little progress on developing efficient and effective algorithms to automatically clarify an image's feature space by using up-to-date techniques such as multi-view analysis [Fraser *et al.*, 2010], filtering and CNNs. It is, therefore, the aim of this chapter to investigate the multi-view analysis by implementing filtering techniques to microarray images to reduce the artifacts' noise and, at the same time, to enhance the position of gene spots.

In this chapter, the focus is that several efficient and effective algorithms have been proposed to automatically clarify the microarray image's features by using up-to-date techniques such as multi-view analysis, filtering and CNN and the best filter is applied with the image transformation engine by quantitative comparisons among different filters in terms of PSNR. The multi-view paradigm combined with Median, Top-Hat and CLD filters is investigated. The proposed image processing methods are applied to a set of real-world cDNA images. It is shown that the CLD filter is the best one to be applied with the image transformation engine. In particular, a fully automated segmentation algorithm, based on the cellular neural networks [Chua & Yang, 1988b,d], is implemented in this chapter as an integrated part of the proposed framework. The proposed algorithms, which can be applied on CNN-UM [Chua & Roska, 1992b], offer a view of parallel computation, which remains reasonably efficient even when applied on Graphics Processing Unit (GPU) simulators [Dolan & DeSouza, 2009; Ho *et al.*, 2008; Soos *et al.*, 2008].

All the issues addressed previously, in addition to the large amount of time that has to be spent on manually processing the microarrays, have stirred a great deal of research interests in using a fully automated procedure to accomplish the task [Bajcsy, 2004; Jain *et al.*, 2002; Katzer *et al.*, 2003]. The main purpose of this chapter follows the same trend by investigating the multi-view analysis and implementing filtering techniques to microarray images to reduce the artifacts' noise and, at the same time, to enhance the position of gene spots.

5.2 Cellular Neural Networks

The basic representation of the CNN cell electronically consists of a single capacitor with nonlinear controlled sources that is coupled to neighboring cells, see Section 4.1.2. The equation that describes the network is Eqn. 4.1, we will state here again:

$$\begin{aligned} C\frac{d}{dt}x_{ij}(t) &= -R^{-1}x_{ij}(t) + \sum_{k,l\in N_r} A_{i,j;k,l}y_{kl}(t) \\ &\quad + \sum_{k,l\in N_r} B_{i,j;k,l}u_{kl} + z_{ij} \\ y_{ij}(t) &= f(x_{ij}(t))) = 0.5(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \end{aligned} \tag{5.1}$$

The main characteristic of cellular neural network is the local interaction between each cell and all neighboring cells within a prescribed sphere of influence on a regular grid. The importance of this formation comes when the task at hand can be solved on a regular grid. This configuration can be implemented efficiently on VLSI. One manifestation of this possibility is the cellular neural network universal machine. CNN-UM is based on the dynamics of CNN and has direct photosensor connections on single VLSI. Therefore, it offers an asynchronous parallel processing, continuous-time dynamics and indirect global interaction of network cells, for more details see Section 4.1.

5.3 The Algorithm

As the algorithm to be developed is concerned with rough gridding information, the segmentation method has to be robust. To get the preliminary coordinates of the spots, the robust gridding algorithm proposed by Morris [2008] has been used. When the girdding information is ready, each region of interest is put to the two-stage process. First, the multi-view analysis is carried out by applying the multi-view function [Fraser *et al.*, 2010]. Ideally, the output of this stage would be the best candidate as input for the segmentation stage. However, analyzing the performance of different multi-view functions in terms of the combinations of the parameters α and β has led to some conclusion and future recommendations to be discussed later. Second, the intermediate output is fed into the segmentation stage. A novel CNN algorithm is then proposed and performed using MatCNN matlab toolbox from AnaLogic Computers Kft. The algorithm could be applied on CNN-UM or GPU.

5.3.1 Multi-View Analysis of cDNA Microarray

Figure 5.1 illustrates the multi-view analysis process, which is called Image Transformation Engine (ITE) in [Fraser *et al.*, 2010] and applied to a dual channel microarray image. After various testing stages are carried out to determine relative performance and speed of execution, a good compromise for the multi-view function is found to be the elements, the square root and inverse transforms, see (5.2). Ideally, such a hybrid function needs to harness the gene spot intensity ranges as calculated by the square root function while, at the same time, taking a higher percentage of the gene spot with similar background intensities from the inverse function.

$$\begin{aligned} S(x) &= \sqrt{x} \\ I(x) &= 1 - \left(\frac{1}{\frac{x}{2^8}+1}\right) 2^8 \\ MV(x) &= \alpha I(x) + \beta S(x) \end{aligned} \tag{5.2}$$

where x is the 16-bit intensity value that is converted into 8-bits.

Applying a smoothing operator before the actual re-scaling process takes place will be of additional benefit as this smoothing process can remove small region high intensity pixels that are of no value in a gene spot identification context. In the basic multi-view process, the two channels of the input image are first smoothed by two different median filters before the actual multi-view filter itself is applied. The role of the first median filter is to render every channel so that the spikes signal will be removed. Yet, this filter will cause the least blurring since the window size is smaller than the spot's size. A 3×3 window will be used to determine the central pixel.

The second median filter is essentially the same as the first in that the second filter uses a larger window region and sampling ratio. In this case, the window measures 57×57 pixels with the sampling or centered pixel set to every forth pixel in turn. This second filter sampling process results in a simple estimation of the image's background features (if the median value is subtracted from every pixel in the image). The background trend estimation will help in attenuating the low-level noises as well as minimizing the effects of

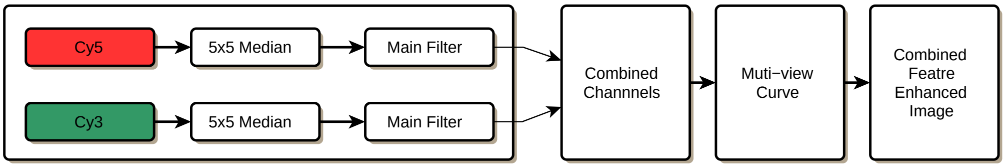


Figure 5.1: Pipeline of multi-view feature response curve generation

large-scale noise.

5.3.1.1 Median Filter

For microarray image processing tools to work independently from the operator, a fundamental step should be considered, that is, the filtering component should be used to clean the image. On the other hand, the effective filter will reduce the noise while reserving the major data required from the image, namely, spots' areas.

Let us emphasize again the importance of the first median filter and its role in removing some sort of contaminations which cannot be removed by the diffusion filter. As has been mentioned before, the median filter will calculate the median of the pixel falling within a predefined window. In this chapter, the window's size is 5×5. Therefore, it has no effect on the spots which have a bigger size.

5.3.1.2 Top-Hat Filter

The top-hat filter can be used to remove the background trend as proposed in [Yang *et al.*, 2002]. The background trend is estimated by using morphological opening, which is obtained by replacing each pixel with the minimum local intensity and then performing a similar operation on the resulting image via the local maximum.

For a region, we use a square of size $(2m+1) \times (2m+1)$ centered on each pixel, where m is a non-negative integer used to specify the size of the top-hat filter. Mathematically, the pixels o_i in the opened image is given by [González & Woods, 2008]

$$o_i = \max_j p_{i+j}, \tag{5.3}$$

where $p_k = \min_j I_{k+j}$ (for $|j_1|, |j_2| \leq m$) with I again denoting the original pixel values. If m is set to a very large value (i.e. $m = \infty$), then $o_i \equiv I_i$ and the filter has no effect. If the top-hat filter is applied, then by using a structuring element obtained through autocorrelation on horizontal and vertical axes mean value vectors, only the pixels in the spots will be substantially changed from I_i to o_i. In this case, by subtracting o from I, these spots will be made more distinct.

5.3.1.3 Complex Diffusion Filter

Linear complex diffusion is another filter that has been used within this chapter. It is one manifestation of the multi-scale approaches, for full discussion see Section 4.2. Complex diffusion follows Perona & Malik [1990] work, and extends it to cover the complex numbers' realm. The generalization step has been achieved by introducing the free Schr´odinger equation to the anisotropic diffusion. In this chapter, we incorporate the complex diffusion process (CLD: complex valued - ramp preserving (5.4)) and the multi-view approach in order to enhance the quality of the image. The real part can be considered as filtered image and the imaginary part can be regarded as a smoothed second derivative by time, when the complex diffusion coefficient approaches the real axes (Gaussian and Laplacian pyramids of the real part).

$$c(Im(I)) = \frac{\exp(i * \theta)}{1 + (\frac{Im(I)}{K * \theta})^2} \tag{5.4}$$

where K is a threshold parameter, Im is the imaginary part of a complex number, I is the image (at any given time). The phase angle θ should be small ($\theta \ll 1$).

5.3.2 Segmentation

In order to qualify the different filtering combinations, we use a modified version of the method proposed by Rekeczky *et al.* [1998b], where Rekeczky used both local threshold estimation and locally adaptive segmentation.

5.3.2.1 Local Threshold Estimation

In the case of using Median Filter or Top-Hat Filter (see Figure 5.2), the threshold estimation is carried out by scaling the mean and standard deviation in the local neighborhood (window 3 × 3 in our case) and adding them up to create the bias map of the adaptive segmentation. The result, therefore, defines a space variant threshold level as a linear combination of the first and second order local statistics (see [Sezgin & Sankur, 2004; Venkateswarlu & Boyle, 1995]).

Since the imaginary part of the complex diffusion filter's output is a smoothed second derivative, we use the imaginary part to create the bias map of the adaptive segmentation, see Figure 5.3.

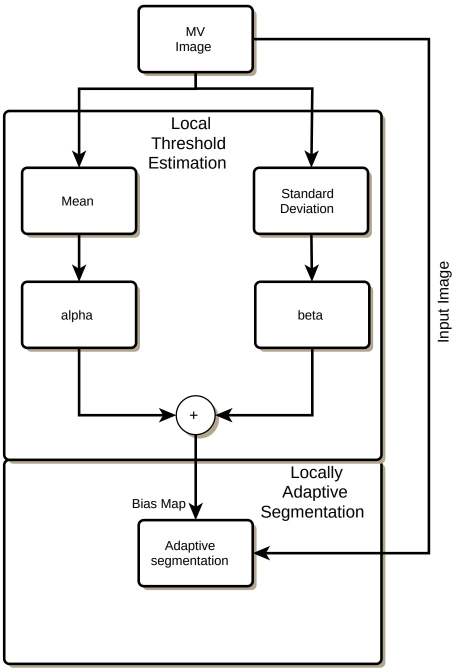


Figure 5.2: Locally adaptive segmentation with median and top-hat filters

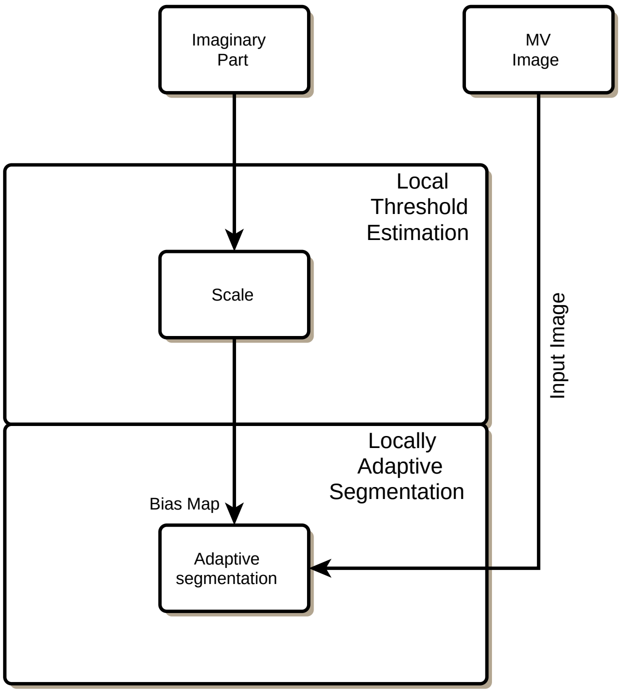


Figure 5.3: Locally adaptive segmentation with complex diffusion filter

5.3.2.2 Locally Adaptive Segmentation

The same method that has been used in Section 4.4.2.6 is applied here. However, the CNN used to carry out the adaptive segmentation with a single template operation **ADTHRES** in (5.5) gives the binary segmentation output [Rekeczky, 2002].

$$\begin{aligned} ADTHRES_A &= \begin{bmatrix} 0.2 & 0.1 & 0.2 \\ 0.1 & 2.0 & 0.1 \\ 0.2 & 0.1 & 0.2 \end{bmatrix} \\ ADTHRES_B &= 0; \\ ADTHRES_I &= -0.5; \end{aligned} \tag{5.5}$$

5.4 Main Results

Although the multi-view is experimentally demonstrated to be efficient in [Fraser *et al.*, 2010], there is still much room for further improvements. Fraser [2006] showed that the median average operators help to reduce small artifacts. However, there is a negative aspect associated with these operators. For instance, by applying the second level median associated with sampling of the background elements slightly more than the sampling of the foreground, there could be a negative effect of reducing the internal gene spot regions intensities.

In this chapter, the second median average operator has been replaced by either morphological Top-Hat filter or Complex Diffusion filter. First, Top-Hat filter is a good tool to estimate the background of the image, and therefore, it is important to test how much it preserves the edge. Remember that enhancing the spot location would have a positive impact when applying hybrid equation (5.2). Second, complex diffusion is suitable to reduce the noise and to enhance the spots' edge, which would be advantageous in integrating with the multi-view process as a whole.

To test the methods proposed, the images in the dataset are divided into many region-of-interest and then processed using the selected methods with the whole range of α and β values, see equation (5.2). Figure 5.4 shows the percentage of each method getting the best output. Note that the complex diffusion filter gives the best performance for most cases (37%).

The best values for α and β specified for each method are listed in Table 5.1. We note that the best output for median and top-hat filters are with the maximum value of α.

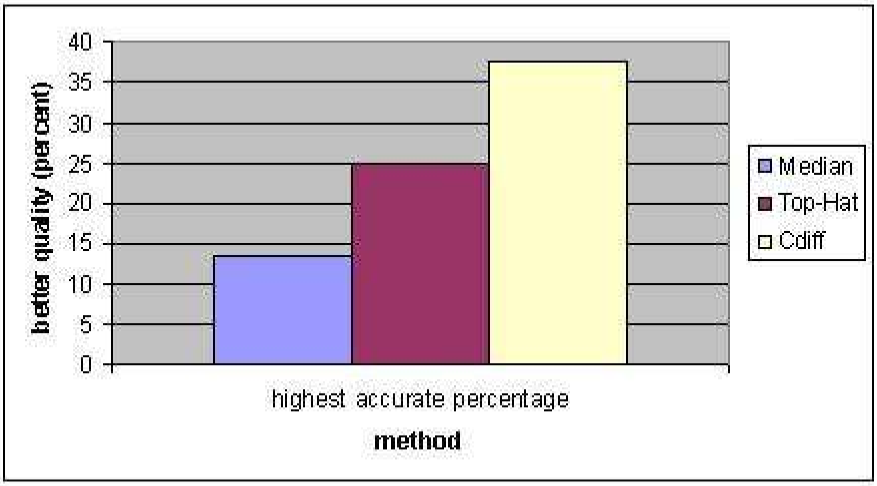


Figure 5.4: The percentage of accuracy for different filters

Although it gives a good filtering performance, the high value of α means removing a lot of information (e.g. spots' intensities) which might be important. On the contrary, complex diffusion gets the best output on the middle range of α and β and, therefore, keeps much more information which might be useful in the later processing.

Filter	$.5 - .5$	$.6 - .4$	$.7 - .3$	$.8 - .2$	$.9 - .1$
Median	1.04	1.04	4.16	9.37	84.37
Top-Hat	0.0	10.41	15.62	10.41	63.54
CDiff	28.12	29.16	16.66	10.41	13.54

Table 5.1: The best α, β vlaues (percent) (see (5.2))

In order to quantify the performance of different filtering methods, a quality measure is required that allows the judgment of how well the calculated template fits the genes' spot position. Note that all the methods produce a mask that classifies the pixels as belonging to either signal (the gene spots) or noise (the local background). For this purpose, an image quality measurement, known as PSNR, see Section 3.7, is used and the rational is justified as follows.

Based on the above discussion, all three methods are applied on a set of raw images to produce masks, which are then scored by using the criterion of PSNR. Figure 5.5, as an example of the scoring outputs, shows how promising the complex filter performs when combined with the multi-view process. From Figure 5.5, we directly compare PSNR values determined by the commercial software GenePix and CLD for the individual images. CLD has shown a marked 2 to 12 dB improvement. Essentially, the CLD process has consistently outperformed the human expert using GenePix in terms of gene spot identification.

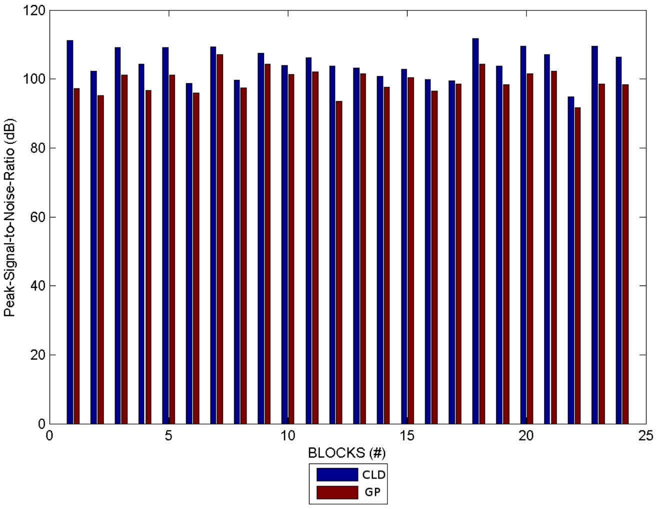


Figure 5.5: PSNR for the Dataset

5.5 Conclusions

This chapter has dealt with the problem of how to improve the processes involved in the analysis of microarray image data. The main focus is to clarify an image's feature space in an unsupervised manner. Rather than using the raw microarray image, it has been suggested that producing multiple views of the image data such that emphasis is placed on certain frequencies or regions of interest would be advantageous and more effective in terms of the overall goal. Different combinations of filtering methods incorporated as a component of multi-view analysis process have been investigated by applying them on a set of real-world cDNA microarray images. Both Median and Top-Hat filters have shown good performance over dataset images. Although the best optimization parameters (α and β), when one set applied over the whole image, could have negative effects on the segmentation process by reducing gene spot regions intensities, using the complex diffusion filter is experimentally demonstrated to be the best among the tested filters.

Fraser [2006] has proven the potential of Multi-view framework. In this chapter, however, the investigation has yielded a more successful design by introducing diffusion based smoothing operator. One of the main conclusions of this work is that the smoothing stage in the framework has a crucial impact on the final performance. Furthermore, the results of the applied diffusion depend on the characteristics of coefficient function; This function prioritizes interregional over intraregional smoothing or the other way around.

Finally, the next step would be designing a dynamic version of this framework. Dynamic version offers a better representation for the later stages of the analysis. However, any possible proposal to this end should consider the local information, since the global-wise approaches will increase the computation complexity significantly with no sound improvement to the overall outcome. In addition, the smoothing operator affects the criteria of any dynamic proposed framework.

Chapter 6

A Novel Cellular Neural Network Approach for Microarray Image Segmentation

The discussion in this chapter based on:

Zineddin, B., Wang, Z. & Liu, X., 2010. A Microarray Image Multiview Analysis Based on Cellular Neural Network. IEEE Transactions on Neural Networks, Submitted.

> *While CNN paradigm is an example of REDUCTIONISM par excellence, the true origin of emergence and complexity is traced back to a much deeper new concept called "local activity"*
>
> Leon O. Chua

A non-standard neural network algorithm for the segmentation of microarray image has been presented in Chapter 3. The main parts of the algorithm have been based on back-propagation and Kohonan networks. Despite the drawback of this algorithm, the significant result of applying a local strategy in order to get the most of the microarray information for later analysis has been highlighted.

In Chapter 4, our approach went on one step further. Since the signal varies remarkably across the image surface, the global information is considerably less important than the local information. In contrast, our algorithm utilized a locally based method that allows some global data from the area that covers a group of neighboring spots. Therefore, the approach has the advantages of relatively global information, which is of vital importance and affects the interpretation of the pixels' intensities. Furthermore, the selection of diffusion based filter has been particularly successful. All diffusion filters that have been used showed a high degree of robustness. In addition, the ability to apply the filter in a local strategy algorithm as well as the ability to solve the PDEs equations using CNN paradigm will be beneficial, as we will see in this chapter.

The multi-view analysis, in Chapter 5, is another approach to improving the quality of the microarray image for the segmentation purpose. In this approach, many views of the image have been produced in order to investigate the intensity of the pixels from different perspectives. Combining median and diffusion based filters in the framework has been efficient to get a fast yet robust algorithm. However, our current concerns are the applicability of this technique on CNN based algorithm and the adaptation of the tuning parameters dynamically. Since the applied method involves pixel-wise operations, which cannot be applied directly in a CNN based application, an algorithm will have to be proposed in order to overcome this limitation. In Chapter 5, the static tuning parameters increase the limitation of this approach. Therefore, an algorithm that selects the parameter dynamically would be highly advantageous.

Therefore, our aim in this chapter is to harness the key conclusions of the previous chapters in one framework. The frequency response transformation, which has been investigated in Chapter 5, is based on simple mathematical mapping functions. The application of this transformation has been experimentally demonstrated to be beneficial for the analysis of the microarray image [Fraser *et al.*, 2010]. However, our research showed that using static parameters (α, β) in (5.2) does not produce the optimal results. Our conclusions emphasized that both the embedded filter and the image itself have effects over the optimum values of the parameters. Therefore, it is highly desirable to investigate the possibility of determining the tuning parameters (α, β) dynamically. On the other hand, considering the design of the algorithm is essential in order to be applied on a CNN-UM. That is, the range of the pixels' values should be within $[-1, +1]$, the mapping functions should give the same response on this range as they do on the original one, and finally the ability of applying a mapping function using CNN should be investigated. All these points will be the topic of the next section.

6.1 Nonlinear Mapping Using Cellular Neural Networks

6.1.1 Introduction

The cellular neural networks are an important type of neural networks, see Chapter 4. Their main advantage is their application in DIP integrated with efficient and robust hardware implementation. The behavior of the CNN is mainly characterized by a local interaction between its nonlinear dynamical cells. The connections between these cells guarantee the direct interaction between local neighbors within a predefined sphere, with radius r, and indirect interaction with the cells outside this sphere.

However, the network's configuration leads to the main drawback in this paradigm. This problem is the limitation in filtering capabilities to a predefined window's templates and the restrictions of the fixed piecewise-linear (PWL) output function. Therefore, in order to be able to apply any mapping function, such as those in Chapter 5, an algorithm for the approximation of any arbitrary function on the CNN-UM framework should be introduced. In this chapter, we will approach this problem by using Chebyshev piece-wise linear approximations [Dunham, 1973, 1974; Fernández-Muñoz *et al.*, 2006].

6.1.2 Chebyshev Approximation

Generally, the approximation techniques deal with two broad types of problems. One sort of problems comes from the need for a simpler type of function to approximate a given function. The other kind of problems arises when we need to derive a best fitting function to a given data among the functions of a specific class.

Chebeshev's methodology is an appropriate technique for approximating any nonlinear function $g(t)$ by piecewise-linear function. The underlaying principle is replacing the parts of the nonlinear function by connected linear segments such as the middle part of the CNN output function. Applying this method will introduce some error e. By using infinite norm as a distance measure, the error function would be:

$$e = \max |g(t) - f(t)| \tag{6.1}$$

where $f(t)$ is the piecewise-linear function in (4.1) that approximates $g(t)$.

The function $g(t)$ can be approximated by a combination of first order polynomials with coefficients $(p_i,\ c_i,\ i = 1, 2, \ldots, N)$, where N is the linear segments which approximate the function $g(t)$ with acceptable error (6.1) in the range $[t_0, t_N]$. The PWL function can be defined as the following:

$$f(t) = p_i t + c_i \qquad t_{i-1} \leq t \leq t_i, \qquad i = 1, 2, \ldots, N \tag{6.2}$$

This polynomial, when minimizing the error e in the range $[t_0, t_N]$, is called the Chebyshev polynomial. Chebyshev polynomials in general are used to minimize the approximation error by reducing the degree of an applied polynomial without suffering from high accuracy loss [Burden & Faires, 2005].

Therefore, the problem here is to design a PWL function $f(t)$ which minimizes the error e between $g(t)$ and $f(t)$. To deal with this problem, Fernández-Muñoz *et al.* [2006] have proposed an algorithm which imitates a PWL approximation for any nonlinear output function on the CNN-UM, which is based on Chebyshev approximation. Their algorithm utilized a Chebyshev polynomial to minimize the approximation error that can be established using Minimax theorem.

Theorem 1. ***(Minimax)*** *[Preciado, 2002]. Let $g(t)$ be a function defined in the open subset (t_i, t_{i+1}), $t_i, t_{i+1} \in \mathbb{R}$ and $P_n(t)$ a polynomial of order n. Then P_n minimizes $\|g(t) - P_n(T)\|_\infty$ if and only if $g(t) - P_n(T) = e = \max(|g(t) - P_n(t)|)$ at least at $n + 2$ points in the range (t_i, t_{i+1}) with alternating sign.*

Fernández-Muñoz *et al.* [2006] considered the case of linear approximation of a nonlinear function. In their approach, three points in each interval t_a, t_b and $t_i \in [t_a, t_b]$ have been specified to meet the maximum error e, with the second derivative of $g(t)$ maintaining its sign over the full range. Therefore, starting from t_a, t_b, the slop of the line p and the constant c will be defined as follows:

$$\begin{aligned} p &= \frac{g(t_b) - g(t_a)}{t_a - t_b} && (6.3) \\ c &= g(t_a) - pt_a && (6.4) \\ P_1(t) &= pt + c && (6.5) \end{aligned}$$

where P_1 is the polynomial that defines the straight line segment. Then, t_i will be defined by $g'(t_i) = p$ meeting the following condition:

$$|P_1(t_i) - g(t_i)| = 2e \tag{6.6}$$

Therefore, based on the sign of the second derivative $g''(t)$, the Chebyshev polynomial that approximates the $g(t)$ in this specific range will be:

$$Q_1(t) = \begin{cases} P_1(t) - e & g''(t) > 0; \forall t \in [t_a, t_b] \\ P_1(t) + e & g''(t) < 0; \forall t \in [t_a, t_b] \end{cases} \tag{6.7}$$

The polynomial $Q_1(t)$ guarantees the maximum approximation error e for all three points t_a, t_b, t_i.

By using this technique, the polynomials $Q_i(t)$ with coefficients $\{p_i, c_i\}$, ($i = 1, 2, \ldots, N$), can be derived. However, two issues should be considered regarding the efficiency of this technique. The first issue is how to specify the optimal number of segments in order to guarantee the desired degree of accuracy by taking into account that the number of segments reflects the number of templates which are needed to perform the algorithm on CNN-UM. The second issue is to choose a more suitable data fitting algorithm to determine the most fitting linear polynomial for every segment of the target function $g(t)$.

6.1.2.1 Chebyshev Nodes

The idea here is that if the maximum error ϵ at t_i between the nonlinear function $g(t)$ and the linear polynomial P_{ab}, which is a line segment within the range $\{t_a,\ t_b\}$, is greater than a preset value ϵ^*, then t_i should be considered as a node and, therefore, the segment $[ab]$ becomes two segments $[ai]$ and $[ib]$. Those steps should be repeated until no more nodes are required.

Algorithm 3 illustrates the steps that are required to produce the list of nodes. Although the algorithm is based on the desired error ϵ^*, the algorithm can be altered to accommodate a maximum number of node number as a condition to stop the algorithm. The error ϵ^*

should be specified based on the requirement of the underlying nature of the application. The procedure “CHECK_ERROR” (Algorithm 3, Line 8) can be approached either by searching for the maximum error within the current range or by specifying the points, where the first derivative of ψ and the slop of linear polynomial are equal; one of these points should represent the maximum error.

Algorithm 3 Chebyshev Nodes Setting

```
Require: $N$ {Initial nodes' number}
Require: $A$ {Lower limit of the range}
Require: $B$ {Upper limit of the range}
Require: $\psi$ {Nonlinear function, to be approximate}
Require: $\epsilon^*$ {Threshold error}
Ensure: $L_o$ {List of nodes}
    {Assuming N = 2}
 1: PUSH ( $L_o$, $A$ ) {Add lower limit to the list}
 2: PUSH ( $L_o$, $B$ )
 3: $TL = L_o$ {Temporary list}
 4: $e$ = TRUE
 5: while $e$ == TRUE do
 6:    $e$ = FALSE
 7:    for Every segment $i \in L_o$ do
 8:       if CHECK_ERROR( $R_i$, $\psi$, $\rho$, $\epsilon^*$ ) == TRUE then
 9:          {$\rho$ is The point of Max error, $R_i$ is the current range}
10:          $e$ = TRUE
11:          PUSH ( $TL$, $\rho$ )
12:       end if
13:    end for
14:    $L_o = TL$
15: end while
```

6.1.2.2 Data modeling

The other important consideration is the way we approach data's modelling. In other words, how the approximation of a non-linear function $g(t)$ by a linear polynomial within a specific range may affect the results of the whole system.

The starting point would be producing a set of M data points that represent the nonlinear function. Therefore, the problem is fitting a set of M data elements (x_i, g_i) to a straight line model.

$$y(x) = px + c \tag{6.8}$$

We want to find the best values for p and c. This problem is often called Linear

Regression. The linear least squares fitting technique [Burden & Faires, 2005] is the simplest and most commonly way to approach this problem, which tends to minimize the square of deviation of each point from this line.

$$\chi^2 = \sum_{i=1}^{N} (g_i - y(x_i))^2 \tag{6.9}$$

6.1.3 Cellular Neural Network Implementation

The next step after obtaining Chebyshev polynomial approximation is to modify the dynamic ranges of input and output of the CNN cells; i.e., the PWL output function should work on input range $[m-d,\ m+d]$ and output range $[a,\ b]$.

The PWL function should be of the form [Fernández-Muñoz *et al.*, 2006]

$$y(u_{ij}) = \begin{cases} a & -\infty < u_{ij} \leq m-d \\ \frac{b-a}{2d}(u_{ij}-m) + \frac{b+a}{2} & m-d < u_{ij} \leq m+d \\ b & m+d < u_{ij} \leq \infty \end{cases} \tag{6.10}$$

where $|b-a| \leq 2$ and $d \leq 1$. The network that can perform this mapping is defined by the following template ($LINMAP$: $X(0) = 0$, $BC = ZF$):

$$LINMAP_A = 0; \quad LINMAP_B = \frac{1}{d}; \quad LINMAP_z = \frac{m}{d} \tag{6.11}$$

This template achieves the linear mapping between $[m-d, |m+d]$ and $[a, |b]$. The output of this stage will be the input for the next network with the template ($DYNRNG$: $X(0) = 0$, $BC = ZF$)

$$DYNRNG_A = 0; \quad DYNRNG_B = \frac{b-a}{2}; \quad DYNRNG_z = \frac{b+a}{2} \tag{6.12}$$

The output of this stage is the desired output for the mapping function (6.10). These two templates will be used to employ Chebyshev PWL linear CNN.

To make these two templates (6.11)-(6.12) sufficient to perform the target mapping, the template should be presented in the following manner [Fernández-Muñoz *et al.*, 2006]:

$$LINMAP_A^{[i]} = 0; \quad LINMAP_B^{[i]} = \frac{1}{d_i}; \quad LINMAP_z^{[i]} = \frac{m_i}{d_i} \tag{6.13}$$

$$\begin{aligned} DYNRNG_A^{[i]} &= 0; \\ DYNRNG_B^{[i]} &= \frac{b_i - a_i}{2}; \\ DYNRNG_z^{[i]} &= \frac{b_i + a_i}{2} - I_i \end{aligned} \tag{6.14}$$

for every segment i, $[t_1^i,\ t_2^i]$, where $a_i = y^i(t_1^i)$, $b_i = y^i(t_2^i)$, $m_i = \frac{t_2^i + t_1^i}{2}$, $d_i = \frac{t_2^i - t_1^i}{2}$. The CNN, which produces the mapping for the any segment, will introduce the saturation function addition. Therefore, in order to get the bias shift required for the correction of accumulative addition, the following is required:

$$I_i = \begin{cases} 0 & i = 1 \\ y^i(t_1^i) & 1 < i < N \end{cases}$$

The last step will be the addition of all the N segment functions applied on every pixel in an image. Therefore, the algorithm applies any arbitrary nonlinear mapping over intensity images.

6.2 Multi-view Analysis: Dynamical Approach

Chapter 5 was dedicated entirely for the investigation of multi-view approach. Many alternatives were considered in order to get the best of such methodology. Furthermore, Multi-view microarray analysis was experimentally demonstrated to be robust, efficient and outperform the Genepix results. However, one of the main conclusions was the drawback of static selection for the parameters α and β, see (5.2).

The application of different filtering ingredients showed that fixed values for (α and β), even when inferred from studying the surface of the image in the dataset, do not guarantee the same performance over different imagery set or different sub filters, such as defining ($\alpha = 0.6$ and $\beta = 0.4$). Therefore, it is beneficial to establish a methodology that can estimate these values based on the information of image itself, for every pixel in our method there is a specified ($\alpha \& \beta$). In such setup, the local spatial information will be of ultimate importance over the global information, although the effects of global properties will present indirectly.

The underlying principle of multi-view approach is that emphasis has to be placed on certain frequency through the image surface. Basically, the mapping functions included in the framework tend to strengthen signals within specific ranges and to cancel some others.

Apart from the two specific functions that were used to achieve the task in (5.2), the main concern here is the underlying argument; i.e., the square-root function is used to emphasize the middle value signals while the inverse function is used to strengthen the low value pixels over the others in order to be canceled later. However, since the inverse function does not have the same characteristics over the CNN-UM input range other function will be used in this chapter to perform the required process.

The main idea in multi-view dynamic approach is that the mean and standard deviation of a sliding 3×3 window are sufficient to estimate the value of Ω, where Ω is the parameter that will be used to calculate α and β for every pixel by the following algorithm:

Algorithm 4 Obtaining the parameters of multi-view methodology, α & β

```
Require: $U$ {Microarray image}
Ensure: $A$ {Array of $\alpha$ for every pixel}
Ensure: $B$ {Array of $\beta$ for every pixel}
    {$r$ defines the window's size; $r = 1 \rightarrow 3 \times 3$}
 1: $M \leftarrow$ MEAN( $U$, $r$ )
 2: $M \leftarrow$ SCALE($M$) {scale the $M$ to the range $[0,\ 1]$}
 3: $S \leftarrow$ STD( $U$, $r$ )
 4: $S \leftarrow$ SCALE($S$) {scale the $S$ to the range $[0,\ 1]$}
 5: $\Omega \leftarrow a * M + b * S$
 6: $A \leftarrow \Omega$
 7: $B \leftarrow 1 - \Omega$
```

The results of applying the algorithm on the dataset suggested optimum values close to $a = 0.8$ and $b = 0.2$ (Algorithm 4, Line 5).

6.3 Microarray Image Analysis

Figure 6.1 illustrates the general structure of a microarray image analysis framework. The vigorousness and potency of the algorithm can be evaluated using the testing dataset, see Section 2.6. This dataset highlights many features that challenge the standard segmentation approaches. These features include week spots' signals, spots' shape problems and different types of noises in spots and background.

In the proposed algorithm, every channel will be processed separately, and then the produced masks will be merged by OR operator. In Figure 6.1, the stages of the algorithm include the smoothing stage, background trend cancellation, multi-view curve generation

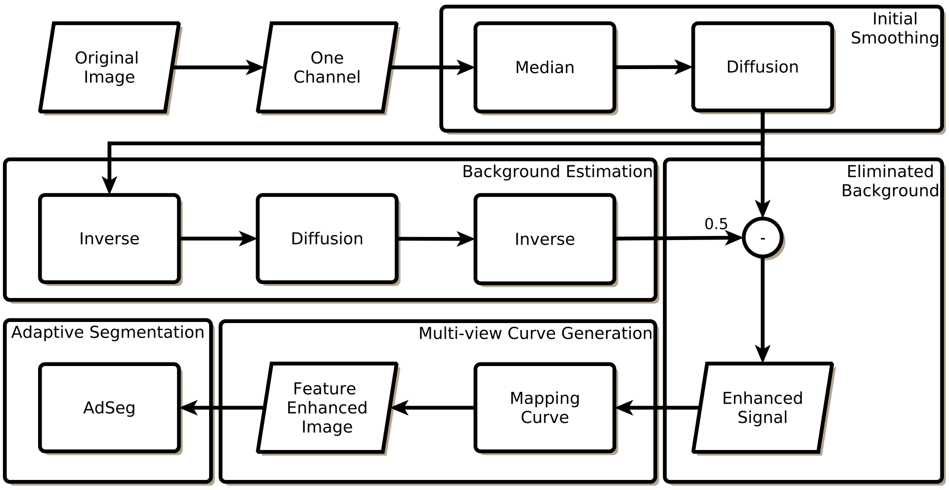


Figure 6.1: Adaptive Mulit-View Analysis AMVA algorithm

and, finally, adaptive segmentation and binary analysis, see Figure 6.2. The novel CNN algorithm is then proposed and tested using MatCNN Matlab toolbox. The algorithm can be applied completely on CNN-UM.

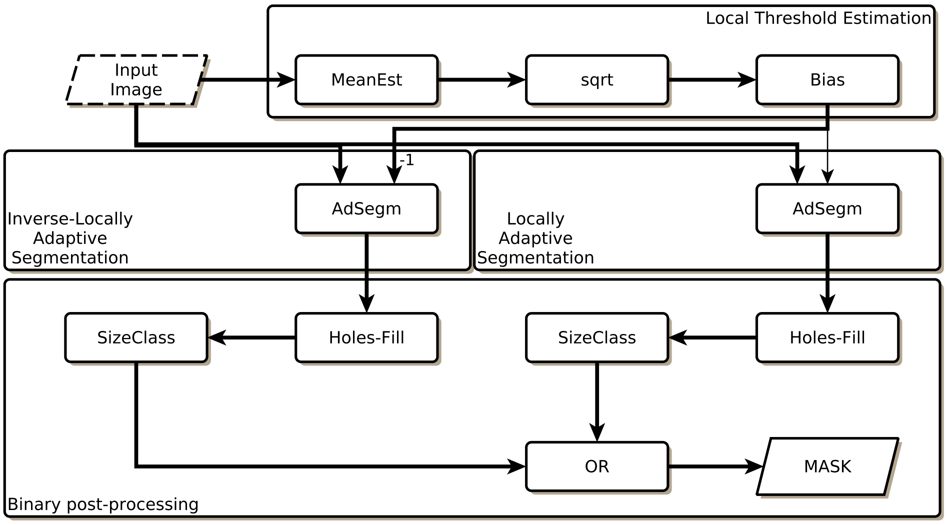


Figure 6.2: Adaptive segmentation algorithm

We have experimentally demonstrated the potential of using the spatial information in producing a robust and reliable segmentation algorithm in microarray image processing application. This local strategy has been kept in mind during the development of our algorithm in this chapter.

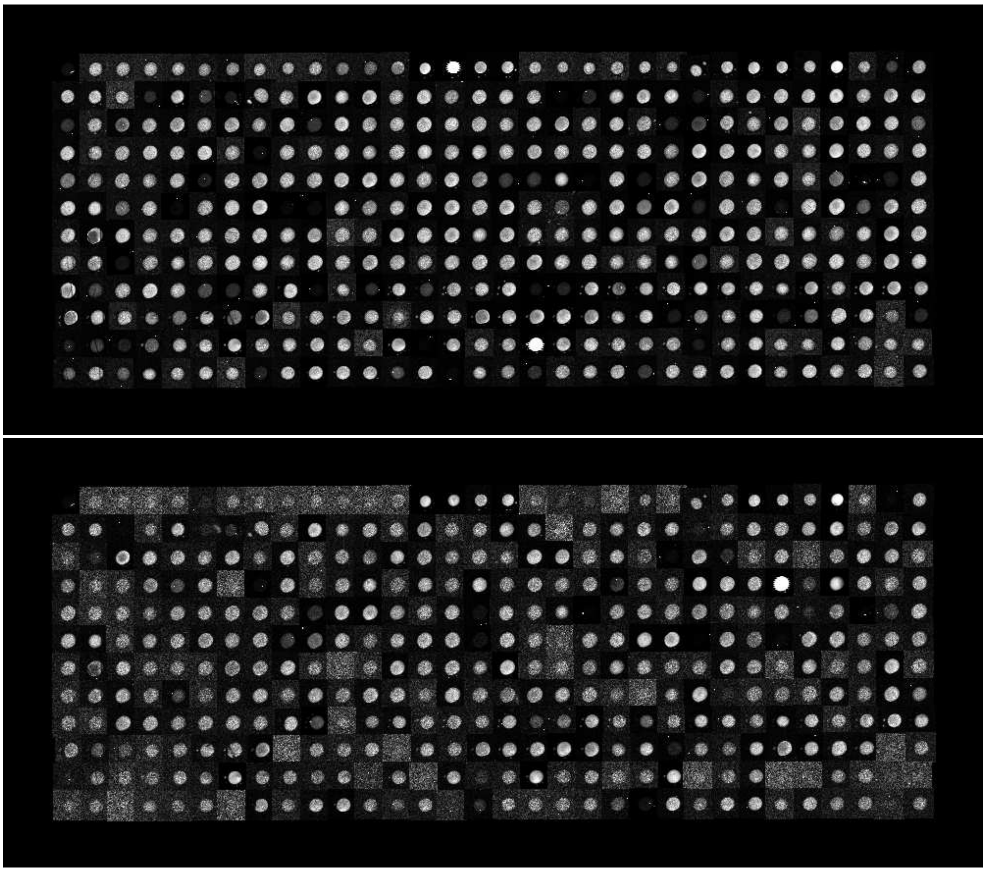


Figure 6.3: An example of microarray image block (green and red channels) will be used to demonstrate the output of each stage in the algorithm

6.3.1 Initial Smoothing

The first step in the algorithm is the median filter, see Section 4.4.2.1. The median filter is applied on a (3×3) window. Due to the difference between the size of the window and the size of the spot, the filter will have a minimal effect of the boundary of the spot and, at the same time, attenuate the impulsive noise. The Median filter can be achieved using the template [Kék *et al.*, 2007] ($MEDIAN$: $X(0) = 0$, $BC = 0$)

$$MEDIAN_A = 1; \quad MEDIAN_D = \begin{bmatrix} d & d & d \\ d & 0 & d \\ d & d & d \end{bmatrix}; \quad MEDIAN_z = 0 \tag{6.15}$$

where $d = -SIGN(x_{ij} - u_{kl})$ and D is the non-linear template in (4.1).

After the median filter, diffusion filer is applied, see Section 4.2. The simplest form of diffusion is the linear version [Chua & Yang, 1988d] ($DIFFUS$: $X(0) = ORIGINAL\, IMAGE$, $BC = ZF$):

$$\begin{aligned} DIFFUS_A &= \begin{bmatrix} 0 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix} \\ DIFFUS_B &= 0; \\ DIFFUS_z &= 0; \end{aligned} \tag{6.16}$$

Another possible alternative is the anisotropic diffusion [Rekeczky *et al.*, 1998b] with the template ($ANIDIF$: $X(0) = ORIGINAL\, IMAGE$, $BC = ZF$):

$$\begin{aligned} ANIDIF_A &= 1; ANIDIF_D \begin{bmatrix} 0 & d & 0 \\ d & 0 & d \\ 0 & d & 0 \end{bmatrix} \\ ANIDIF_z &= 0; \end{aligned} \tag{6.17}$$

where $d = g\Delta v_{xx}$ and

$$g = \begin{cases} 1 - |\Delta v_{xx}|/2k & |\Delta v_{xx}| < 2k \\ 0 & otherwise \end{cases} \tag{6.18}$$

where $ANIDIF_D$ is the non-linear template and k is the noise level estimate.

The main advantage of diffusion type filter is that it achieves an acceptable degree of smoothness yet preserves the edges efficiently.

6.3.2 Background Trend Estimation

Having the image smoothed, one problem has to be dealt with is the non-uniform background intensity values. The approach that has been followed in Section 4.4.2.3 is the median filter over a window of a size more than the size of the spot. This method produces an image with spots that are efficiently distinguished from the background. The size of the window is essential for removing the spots from the image and then get the trend of the background. However, the mean filter with a suitable window's size is suitable as well to achieve this goal [Akbari & Albregtsen, 2003]. Furthermore, a good method to get the average is by using the diffusion filter (6.16) with a long transient time which will lead to a good background estimate. In the Algorithm 6.1, a proportion of the resulted image has been canceled from the microarray image to get an enhanced features' image.

6.3.3 Multi-view Curve Generation

The next stage is the multi-view curve production. The process follows the discussion in Sections 6.2 and 6.1. The first step here is to define efficient functions which can achieve the desired outputs. However, as it has been discussed earlier, the inverse function that has been employed in Chapter 5 is inapplicable in this context. Therefore, a new function has been proposed to replace it. The multi-view process will follow these two functions:

$$S(x) = \sqrt{x} \tag{6.19}$$

$$I(x) = 1 - e^{-0.5x} \tag{6.20}$$

For every function, the methodology steps will be the following. First, the number of chebyshev nodes is specified with a maximum acceptable error ϵ^*. Second, for every segment that has been determined, the relevant CNN template is derived. At this point, the derived template will be part of the multi-view stage in Algorithm 6.1. Therefore, the whole algorithm is applied to get the estimation of the mapping function.

The matrices A and B can be set using the method in Section 6.2. These two matrices are adaptive and are based entirely on the microarray's surface properties. At this stage, the multi-view adaptive curve can be produced directly by applying the following equation:

$$Y_{ij} = \sum_{ij} A_{ij} S_{ij} \; - \; \sum_{ij} B_{ij} I_{ij} \tag{6.21}$$

where Y is the feature enhanced image for every channel.

The mean can be estimated by the template [Rekeczky, 2002] ($MEANEST$:$X(0) =$

$ORIGINAL\ IMAGE$, $BC = ZF$):

$$\begin{aligned} MEANEST_A &= 0; MEANEST_B = \begin{bmatrix} 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \end{bmatrix} \\ MEANEST_z &= 0; \end{aligned} \tag{6.22}$$

and the Standard deviation can be calculated be the template [Rekeczky, 2002] ($STDEST$: $X(0) = ORIGINAL\ IMAGE$, $BC = ZF$):

$$\begin{aligned} STDEST_A &= 0; STDEST_B = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \\ STDEST_z &= 0; \end{aligned} \tag{6.23}$$

Figure **??** is the output of filtering stage for the block example Figure 6.4.

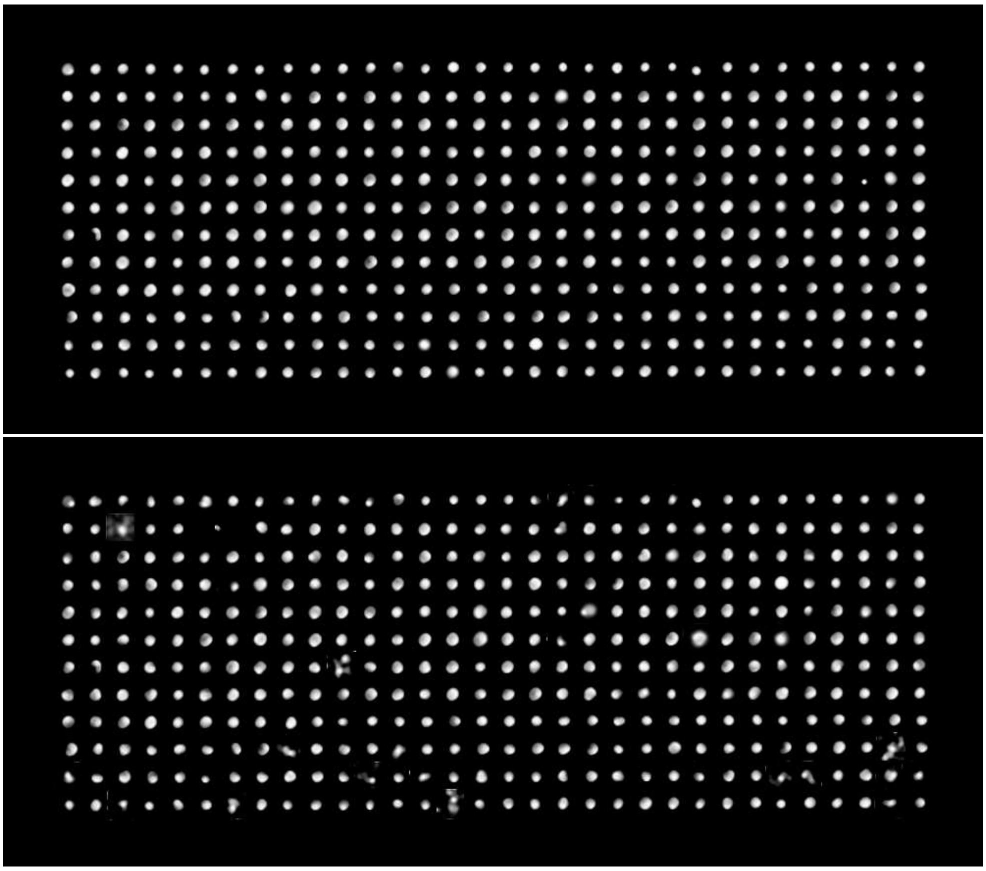


Figure 6.4: The output of filtering stage for the block example presented in Figure 6.4

6.3.4 Adaptive Segmentation

The algorithm in Figure 6.2 highlights the main steps of the adaptive segmentation algorithm. Thus, the first step is to establish the adaptive threshold. The threshold estimation can be described by the following equation:

$$\Theta = \sqrt{\mu_r(U)} \tag{6.24}$$

where $\mu_r(U)$ is the image composed of the mean in the local neighborhood of radius r (see Template (6.22)), U is the input image, Θ is the threshold estimate which is used as a bias map of the CNN in the next step. Equation (6.24) defines the noise level estimation across the image surface based on proportions of mean. The $(\sqrt{.})$ function has been used to emphasize the middle rage signals, which is most probably part of the spot's signal.

The output of this last step will be the bias for the *AdSegm* template in Section 4.4.2.6. However, the adaptive segmentation template applied twice. The first part is the "Locally Adaptive Segmentation", where the bias is applied directly. The second part is one way to capture the spot signal even when the pixels' intensities are less than the background's. In the "Inverse-Locally Adaptive Segmentation", the inverse of the bias is applied with thresholding template.

The outputs of the two thresholding templates will then go through the *HoleFil* & *SizeClass* templates before mixed together by *OR* operation to form the last output mask for each channel. The output of this last step would be a black and white image. The next steps in the algorithm are first removing the spots with holes and then removing irregular size spots.

6.3.4.1 HoleClass

In Chapter 4, the spots with holes were removed from the output image. However, investigating the outputs of applying that algorithm on the whole dataset showed that keeping those spots and trying to fill in the holes might come with benefits to the whole performance of the segmentation algorithm. Therefore, in this chapter the following simple template [Fajfar *et al.*, 1998] will be used to fill the holes, ($HOLEFIL$: $X(0) = 1$, $BC = ZF$):

$$\begin{aligned} HOLEFIL_A &= \begin{bmatrix} 0 & 2.5 & 0 \\ 2.5 & 4 & 2.5 \\ 0 & 2.5 & 0 \end{bmatrix}, \; HOLEFIL_B = 10 \\ HOLEFIL_z &= 0; \end{aligned} \tag{6.25}$$

6.3.4.2 SizeClass

The main strategy for dealing with the size problem is similar to the methodology in Section 4.4.2.7. However, removing the small clusters will follow exactly the same method in Figure 4.8, while removing the big clusters needs an extra step. This last step is XOR template that will produce the XOR function between the input mask and the output of RECALL template. Note that n for the first case will be a small number in order to specify the smallest spot, and n in the second will be big enough to keep only clusters, which are of a size that cannot be considered as a spot. Figure 6.5 represent the output mask for green and red channels while Figure 6.6 is the final output of the system.

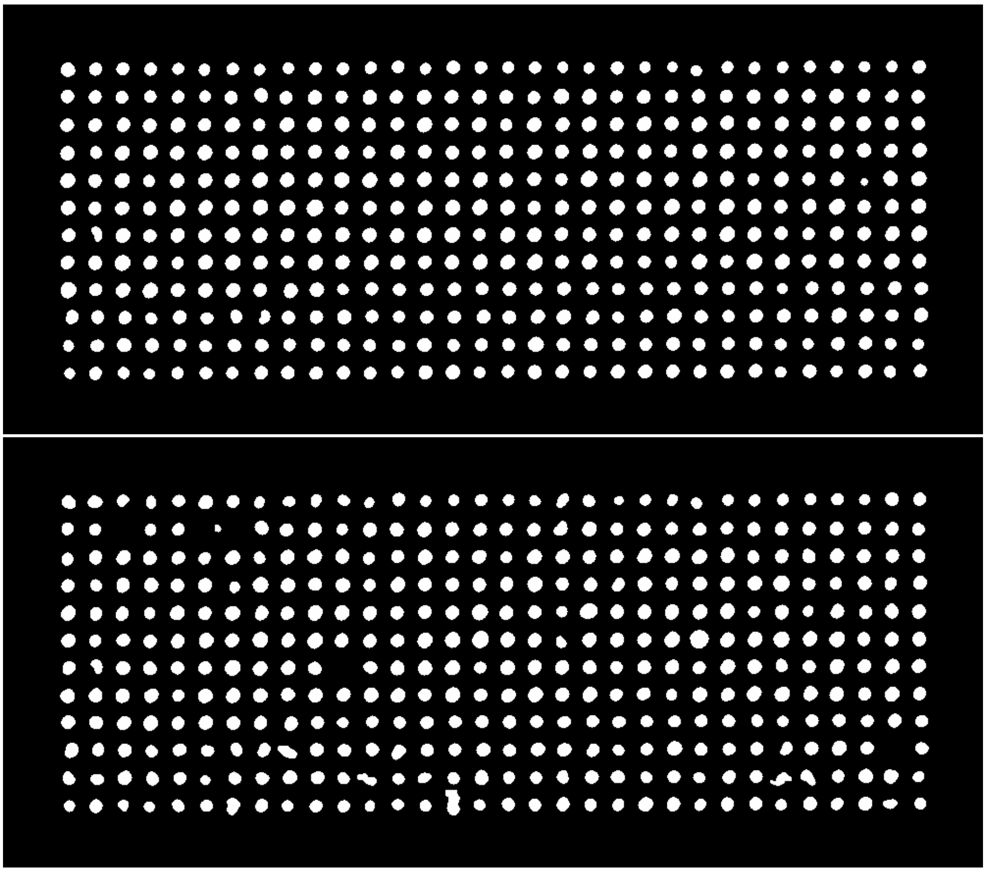


Figure 6.5: Segmentation results for the green and red channels in Figure 6.4

6.4 Discussion

Before discussing and evaluating the results of the proposed algorithm, specifying the method for defining the background of every spot should be considered. As in Chapter

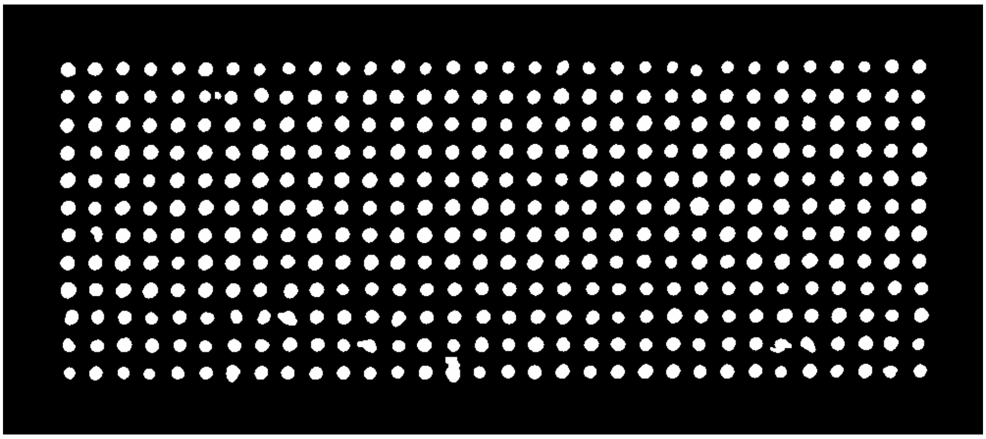


Figure 6.6: Block 6.4's final output

4, pixels of the squared area around every spot, except the spot's pixels, will be regarded as a background. This area will represent more rigorously the distribution of the background and, therefore, this presentation will be used to calculate the required statistic measurements of every spot and its background.

6.4.1 Testing Dataset

The dataset that has been used in this chapter is the same set that has been used for the previous chapters. Every slide in these set consists of 24 blocks, each contains 32×12 gene spots. The first row of every block contains 32 pre-defined genes from Lucidea ScoreCard [Samartzidou *et al.*, 2001] that can be used to test various experiment properties. The rest of the 11 rows contain the human genes under study, each gene is repeated twice; odd and even side blocks.

6.4.2 Evaluation

Finally, the performance capabilities of the proposed algorithm should be evaluated. Quantifying the performance consists of two points; first, the operating of the mapping algorithm will be assessed, in particular, the chebyshev approximation methodology against normal Chebyshev approximation setting will be considered. Second, the investigated technique produces a template that classifies the pixels into either gene spot or background. For this purpose, a descriptive statistic Interclass Correlation Coefficient (ICC) measures are used in order to compare the results produced by the proposed algorithm and GenePix output. Besides, the two subjective comparison methods utilized in Chapter 4 are used as well.

The rational is justified as follows.

6.4.2.1 Non-linear Mapping Using CNN

A methodology for mapping a non-linear function using a PWL output function of the CNN has been discussed in Section 6.1. The best way to evaluate the algorithm is by comparing the outputs with the approximated function. Therefore, we assume the function $y = \sqrt{(x)}$ is the function to be approximated. Two methods were used, the first, the proposed method was applied with a maximum error $\epsilon* = 0.01$, and the second, the original algorithm was applied without the proposed improvements.

The proposed algorithm has shown that the maximum number of segments is 9, thus the CNN templates will be 18 templates. Each couple of these templates will be applied successively, then the sum of all the nine couples will be the final output of the algorithm. Table 6.1 shows the parameters of these nine segments.

Range	p	c	a_i	b_i	m_i	d_i
0-0.0004	35.8076	0.006	35.8076	0.006	0.0002	0.0002
0.0004-0.0019	14.7001	0.0164	14.7001	0.0164	0.0012	0.0007
0.0019-0.0078	7.35	0.0328	7.35	0.0328	0.0048	0.0029
0.0078-0.0312	3.6750	0.0657	3.675	0.0657	0.01953	0.0117
0.0312-0.07062	2.2369	0.1103	2.2369	0.1103	0.0509	0.0196
0.0706-0.125	1.6059	0.1546	1.6059	0.1546	0.0978	0.0271
0.125-0.2825	1.1184	0.2206	1.1184	0.2206	0.2037	0.0787
0.2825-0.5	0.8029	0.3092	0.8029	0.3092	0.3912	0.10875
0.5-1	0.5813	0.4259	0.5813	0.4259	0.75	0.25

Table 6.1: Chebyshev approximation for the $y = \sqrt{x}$ function

On the other hand, nine equal segments were used in order to approximate the same function $y = \sqrt{(x)}$ without any dynamic data modelling. The results of both methodologies are illustrated in Fig. 6.7.

The results in the Figure 6.7 shows that the original algorithm gives an output better than the improved one over some part of the function. The overall performance, however, suggests that the improved algorithm outperforms the original one. Furthermore, in order for the original algorithm to give a similar performance with a relatively similar maximum error, the number of templates needed to be applied will be very high, particularly if there is a large variation in the first derivative of the approximated function.

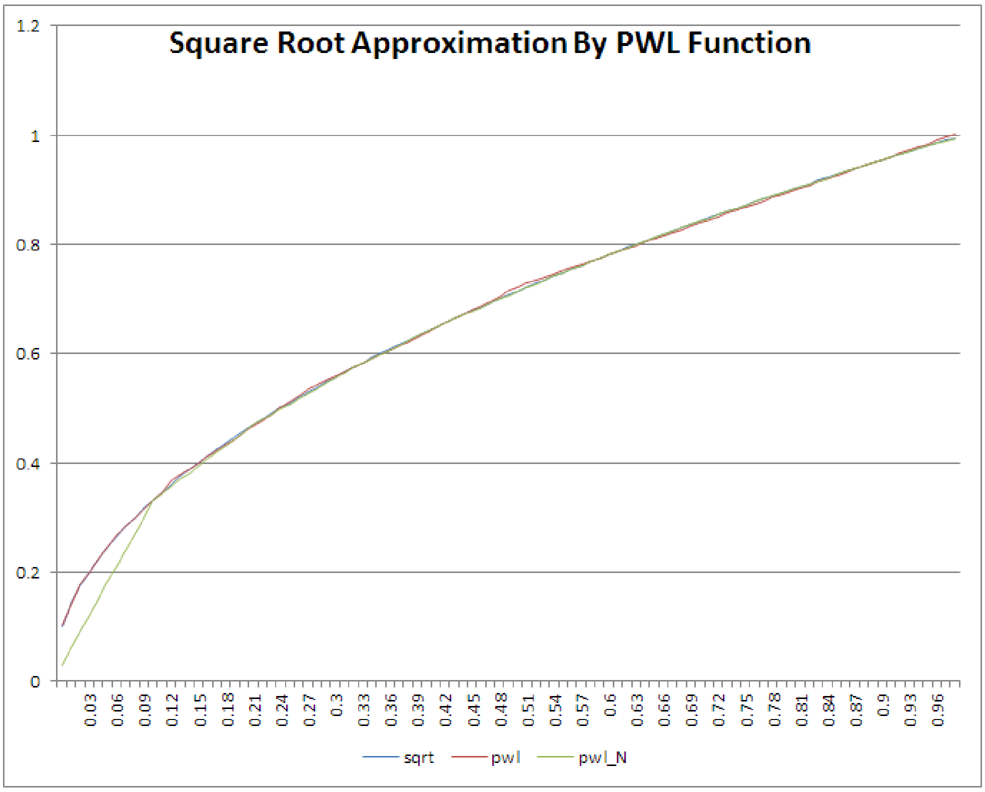


Figure 6.7: (a) sqrt represents the approximated function $y = \sqrt{x}$; (b) pwl represents the results of the improved algorithm; and (c) pwl_N is the results of the original algorithm using the same segment number that is suggested in (b)

6.4.3 The Subjective Comparisons

Figure 6.8 shows a direct comparison between PSNR values (see Section 4.5.1), determined by GenePix and the proposed CNN technique for the dataset images and on average. Our algorithm shows a remarkable 0.5 to 6 dB improvement. Although GenePix outperforms the CNN algorithm in one block measure but, apart form that, the performance of the proposed algorithm is remarkably better than GenePix's. Note that the proposed algorithm is fully automatic and totaly blind.

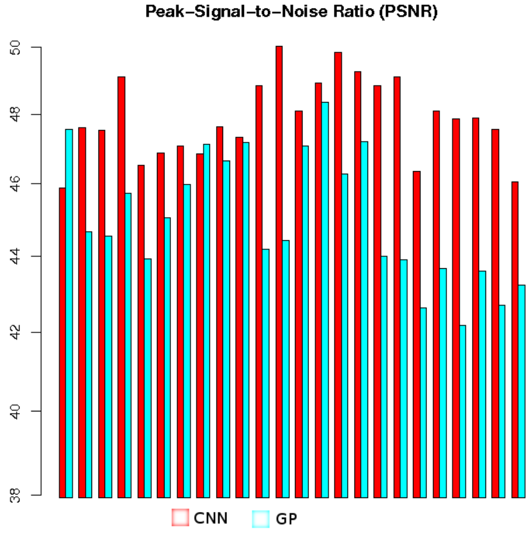


Figure 6.8: psnr

k-means objective function is another measurement that was used to compare the two methods, see Section 4.5.2. Based on Figure 6.9, we can see that the proposed algorithm shows an improvement. Essentially, the superiority of the algorithmic process over the human expert using GenePix, in terms of gene spot identification, is experimentally demonstrated.

6.4.3.1 The Objective Comparison

An objective comparison has been carried out using the Interclass Correlation Coefficient (ICC) measures, see Chapter 4. The underlying principle is that by comparing the results both between methods within arrays and within methods between replicate genes on the same array, and by assessing the observed variations relative to the variations between genes, we can objectively compare image processing methods [Korn *et al.*, 2004].

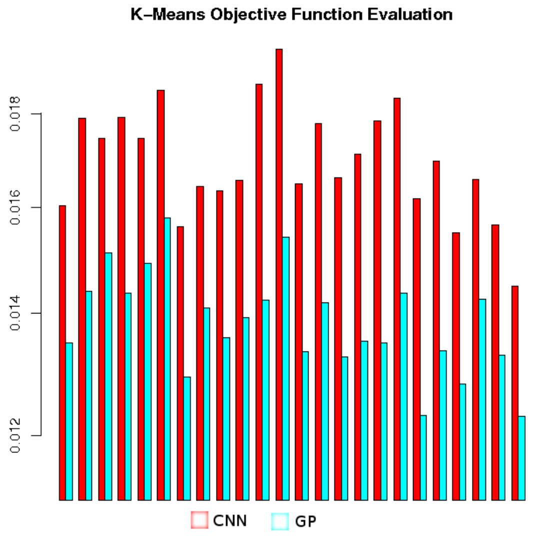


Figure 6.9: k-means

Figures 6.10-6.12 present the estimated variance components and ICC for the dataset images and on average. The reliabilities of both methods are high with our method showing results that are perhaps slightly more reliable than GenePix. Note that even the within-spot variability $\hat{\sigma}_e^2$ (the noise) is smaller for our method, and the between-spot variability (the signal) is bigger likewise.

6.5 Conclusion

In this chapter, a novel CNN algorithm for segmentation of microarray image has been proposed in order to improve the microarray image analysis. The framework consists of several components that can be applied sequentially on CNN-UM, and produces a mask that can be used to calculate the final log ratios and related statistics without human intervention. The proposed algorithm offers a simple yet a robust way to process the microarray image.

This chapter concludes the findings of all the previous chapters in one framework that highlights and offers the following points:

1. All the stages in the proposed framework utilize the local information with indirect access to the global properties of the image. Therefore, the local strategy filtering is experimentally demonstrated to be efficient to deliver the desired outcomes in

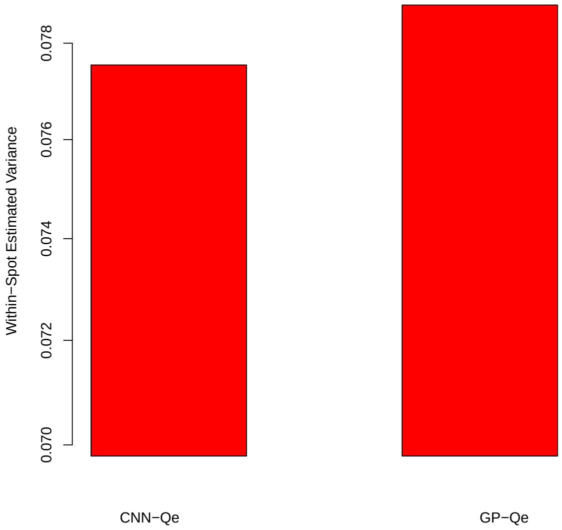


Figure 6.10: Within-spot estimated variance σ_e^2

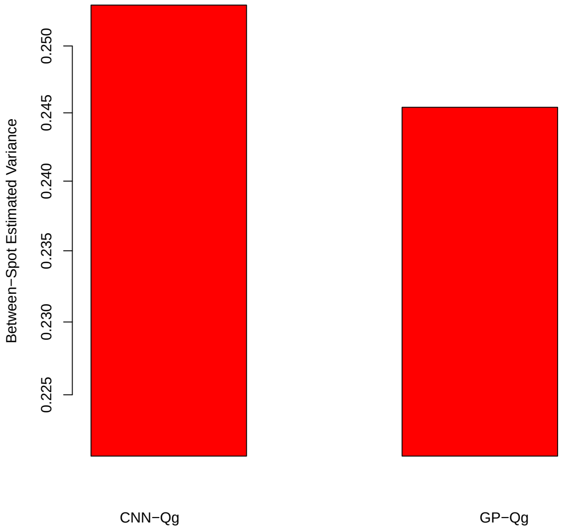


Figure 6.11: Between-spots estimated variance σ_g^2

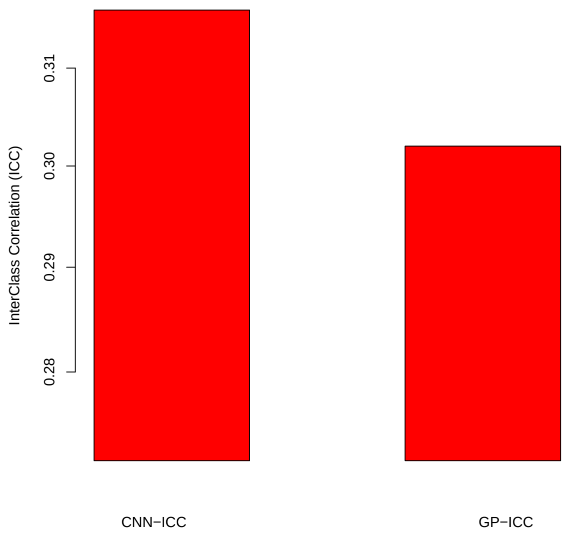


Figure 6.12: Interclass correlation ICC

both enhancing the image quality stages and adaptive segmentation and binary post-processing stage.

2. The blindness of the algorithm should be emphasized since the algorithm takes only the raw data as the input and no assumption is made about the structure of the image or the properties of the signal.

3. The diffusion based filter is experimentally demonstrated to be beneficial and is used within different stages of the algorithm in order to get as much as possible from any microarray image.

4. The potential of the proposed framework is demonstrated by direct comparisons between our proposed approach and semi-automated segmentation results achieved by GenePix (the commercially available software), over a real imagery dataset.

5. In Chapter 5, the multi-view analysis showed a high potential as a part in microarray image analysis. However, in this chapter many methods have been developed to overcome the drawbacks that have been discussed in the previous chapter. Toward this end, an algorithm based on Chebyshev approximation is improved to get a better quality output yet with less cost; i.e., number of the CNN networks that are needed to approximate the multi-view mapping function.

6. The proposed algorithm has no need for prior knowledge about the gridding information. On the contrary, the resulted image can be easily analyzed to specify the blocks' structure of the image with any simple algorithm.

7. Finally, the proposed framework can be a basis for a hardware realization for microarray image processing, since it can be applied directly on a CNN-UM with a raw microarray image as input.

One last comment about the current framework is that since this algorithm has been experimentally demonstrated to be an effective and efficient segmentation algorithm, it can be utilized completely on CNN-UM. Therefore, it would be beneficial if we can deal with the normalization of the image's background within the same framework. This last point will be the topic of the next chapter.

Chapter 7

A Cellular Neural Network for Microarray Image Reconstruction

The discussion in this chapter based on:

Zineddin, B., Wang, Z. & Liu, X., 2010. Cellular Neural Networks, Navier-Stokes Equation and Microarray Image Reconstruction. IEEE Transaction On Image Processing, Revised.

Zineddin, B., Wang, Z. & Liu, X., 2010. Cellular Neural Networks, Navier-Stokes Equation and Microarray Image Reconstruction. In IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW). HK, 234-239.

Zineddin, B., Wang, Z. & Liu, X., 2010. A Cellular Neural Network for Microarray Image Reconstruction. In Proceedings of the 16th International Conference on Automation & Computing. Birmingham, UK, 106-111.

> *Thus the partial differential equation entered theoretical physics as a handmaid, but has gradually become mistress.*
>
> Albert Einstein

Designing a robust algorithm for the microarray segmentation is the main target throughout this thesis. In Chapter 6, a remarkable algorithm based on cellular neural network has been proposed. The CNN algorithm has experimentally demonstrated to be a highly robust and fast way to classify the pixels of the microarray image into a background and foreground groups. However, our algorithm gives only the mask, while the normalization process will be later used to produce acceptable quantification values of the microarray image. Our next step is to obtain the best values that represent the spot and background signals during image processing stage.

Some hardware implementations of microarray image processing have been proposed in the literature and proved to be promising alternatives to the currently available software systems. However, the main drawback of those proposed approaches is the unsuitable addressing of the quantification of the gene spot in a realistic way without any assumption about the image surface.

Therefore, it is our aim in this chapter is to present new Image Reconstruction algorithms using the Cellular Neural Network. These algorithms offer a robust method for estimating the background signal within the gene spot region. The MATCNN Toolbox for Matlab is used to test the proposed method. Quantitative comparisons are carried out in terms of objective criteria between our approach and some available methods such as Bertalmio's method [Bertalmio *et al.*, 2000] and median background estimation, which is widely used in available packages such as GenePix, ScanAlyze and ImaGene. It is shown that the proposed algorithm gives highly accurate and realistic measurements in a fully automated manner, and also, in a remarkably efficient time.

7.1 Introduction

In general, analyzing the microarray image complies with the following main steps (Chapter 2): 1) the filtering stage; 2) the gridding stage; 3) the segmentation stage; and 4) the quantification stage. The available software packages for image analysis (e.g. [Anonymous, 1999; Eisen, 2010]) are largely dependent on manual, semi-automated or fully automated methods, which consume a lot of processing time. For instance, ScanAlyze [Eisen, 2010] software requires the operator to take as many as 14 steps [Bassett *et al.*, 1999] and many of these steps have to be repeated several times. Therefore, in order to manifest the full potential of the parallelism of the microarray technology, implementing hardware microarray image analysis is an auspicious alternative to these tools. Furthermore, the hardware utilization should process the image to obtain highly accurate and realistic data in a fully automated manner within a remarkably efficient time.

Apart from our approach in the previous chapter, there are two hardware implementation proposals. To overcome the aforementioned bottleneck for microarray image processing, Samavi *et al.* [2004] proposed a hardware architecture to analyze the microarray image. In particular, Arena *et al.* [2002b] used the Cellular Neural Networks (CNN) and analogic (analog and logical) signal dedicated CPU called CNN universal machine. Note that the CNN [Chua & Yang, 1988b] framework provides a flexible approach to describing spatiotemporal dynamics in the discrete space. In particular, it allows for efficient VLSI (very-large-scale integration) implementation of analogue, array-computing structures. Such devices possess a huge processing power that can be employed to solve numerically expensive problems. The CNN representation of a PDE (Partial Differential Equation) is a spatially discrete dynamical system which is qualitatively equivalent to the original, spatially continuous system. Both systems operate in continuous time, and values of state's variables, interactions and parameters are all continuous. In [Arena *et al.*, 2002b], the proposed algorithm facilitates the parallel nature of the CNN to achieve the required objectives. Unfortunately, there have been two limitations with the methods developed in [Arena *et al.*, 2002b; Samavi *et al.*, 2004]: 1) the operator has to define, in advance, a specific set of threshold values in order to address the intensity analysis. Consequently, the analysis depends on a particular hypothesis about the underlying question of the experiment; 2) in these methods, it is implicitly assumed that the background noise and the other artifacts are absent in the output of the segmentation stage, which is not necessarily the case. To this end, it is concluded that the image analysis stage should have a specific background determination process that can analyze the inherent variation between the gene and the background signals within any proposed hardware framework dedicated to microarray image analysis, and this constitutes the motivation of our current investigation.

In this chapter, a new methodology for DNA microarray image reconstruction is proposed. The idea is to use a practical CNN approximation to the Navier-Stokes Equation (NSE), which describes the fluid velocity in the incompressible fluid, to obtain an exemplary approximation of the background in the gene spot region. The theoretical basis of this approach can be found in Bertozzi & Bertalmio [2001] where the remarkable similar-

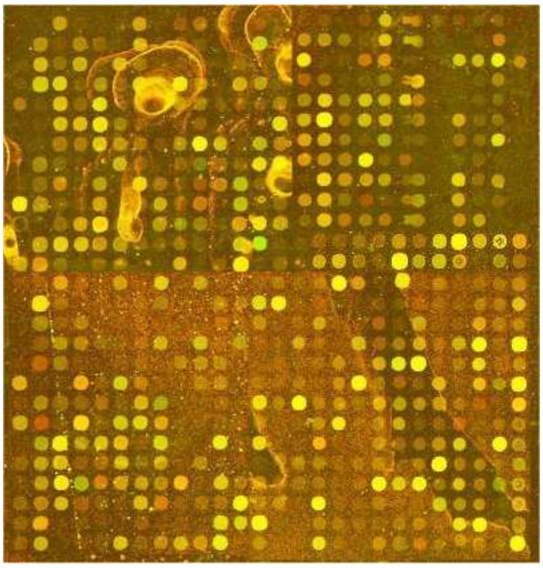


Figure 7.1: Part of a microarray image gives a good example of the variations in the background signal. The target is to reconstruct the areas of the spots, the bright circles signals, assuming that there is no information in these areas (‘$0'$s)

ity has been highlighted between the steam function and the image intensity. It has also been suggested in [Bertozzi & Bertalmio, 2001] that the NSE solution is applicable to the inpainting (reconstruction) purpose. A CNN is an analogic processor array that allows the application of local strategy, with less computational complexity, to meet the task requirements. It is important to note that using local information leads to a robust and reliable algorithm in some applications such as microarray image reconstruction as we will see later. Due to its architecture, the two-dimensional CNN array is widely used to solve image processing and pattern recognition problems. Furthermore, the parallelism of this structure allows one to perform the most computationally expensive image analysis tasks in a faster way than classical CPU-based computer. For our research, subtracting the reconstructed background from the original should give rise to a more accurate quantification of genes’ signals. In this chapter, the gene expression results of the proposed reconstruction method are compared to those as produced by methods utilized in some available systems commonly used by biologists to analyze images, such as GenePix. besides, the comparison is carried out between our method and the results of Bertalmio’s method [Bertalmio *et al.*, 2000].

The main contribution of this chapter lies in two aspects: 1) two new microarray image reconstruction algorithms are proposed. The first one is by using Cellular Neural Networks representation for diffusion equation. Another one is by using the Cellular Neural

Network that solves the Navier-Stokes equation and such an algorithm is experimentally demonstrated to be robust for estimating the background signal within the gene spot region; and 2) the CNN templates (complete set of templates) are developed with specific steps to achieve the microarray image reconstruction. The paper is organized in the following manner. First, we formalize the problem area as it pertains to microarray image data and briefly explains some available approaches in Section 7.2. Section 7.3 discusses the basic idea of our proposed algorithms with appropriate steps involved in the analysis highlighted. We then briefly describe the data used throughout the work and evaluate the algorithm over real-world data in Section 7.4. Section 7.5 summarizes our findings and renders some observations into possible future directions.

7.2 Existing Techniques

The most popular method for the log2 ratio is subtracting the background median from the foreground median. This methodology based on the assumption that there are little variations within the spot area and within the background area.

Unfortunately, this is not always the case. A good example of the low-level signal produced in the image can be seen in Fig. 7.2. The image may have many problems such as the missing or partial gene spots, shape inconsistencies, and background variation, i.e, the scratch and the variation of the background illuminations around the presented genes.

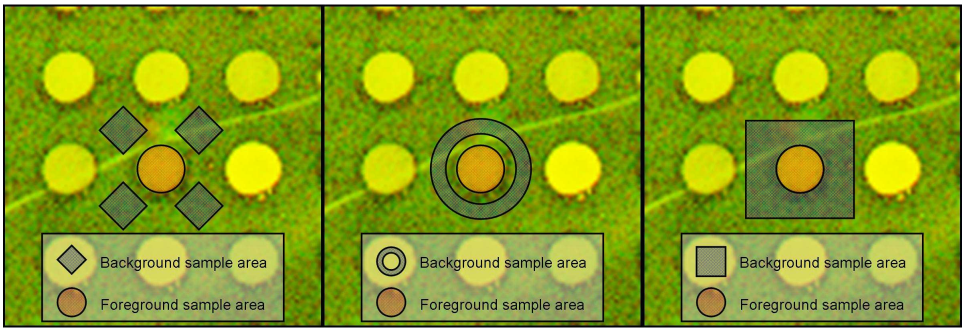


Figure 7.2: Gene spot background region as used by common packages: 1) GenePix; 2) ImaGene; 3) ScanAlyze

In the recent years, there has been a growing need for establishing a more specific background determination process that can account for the inherent variation between the gene and background regions. One of the first techniques applied specifically to reconstruct microarray images is the proposal of O'Neill *et al.* [2003]. In particular, a gene area is replaced by selecting pixels, that are most similar to the known border, from a known background region. The underlying assumption is that the similarity with the given border intensities guarantees the transition of the local background structures through the new region.

7.3 Novel Techniques

7.3.1 Description

In this chapter, we propose to use CNN approximation to the Navier-Stokes equation for Image Reconstruction (CNN-NSIR) that is a novel technique that removes gene spot regions from a microarray image surface. The removal of these regions leads to more accurate background estimation, which can then be used to yield yet more realistic genes' signal. Techniques such as O'Neill's work [O'Neill *et al.*, 2003] in the spatial domain exclusively and essentially utilize the gene border pixels as a reference values to produce appropriate pixel mappings. Although this approach works well, such kind of brute force methods are typically expensive with respect to the execution time. However, if we can utilize the locally spatial information integrated with a hardware implementation, we are able to overcome this limitation. Suppose we have a subset $\Omega \in D$ where we would like to modify the gray-level of I based on the information of I from the surrounding region $D \backslash \Omega$ where Ω is the reconstructed region. The modified region I^*, the solution, will have equal values as I in $D \backslash \Omega$. The process of finding the appropriate I^* is the reconstruction problem.

The approach proposed in Bertalmio *et al.* [2000] attempts to mimic techniques as used by skilled artists to perform inpainting manually. It works on the principle of a PDE isotropic diffusion model. Using a mask to specify the area to be inpainted, the algorithm fills in these areas by propagating the information of the border region along a level line (isophotes). Isophotes are level lines of equal graylevels. Mathematically, the direction of the isophotes can be interpreted as

$$\nabla^{\perp} I \tag{7.1}$$

where ∇ is the gradient (∂_x, ∂_y) and $\nabla^{\perp} = (-\partial_y, \partial_x)$ means the direction of the smallest change. Next, the smoothness could be interpreted as

$$\Delta I \tag{7.2}$$

where Δ is the usual Laplace operator $(\partial_x^2 + \partial_y^2)$.

In general, ΔI will extract edge and noise in an image. Therefore, in order to mimic the idea of artistic inpainting, we should propagate ΔI in the direction of $\nabla^{\perp} I$ from the boundary of the reconstructed area $\partial\Omega$. Consequently, the solution criterion for the inpainting problem I^* satisfies

$$\nabla^{\perp} I^* \cdot \nabla \Delta I^* = 0 \tag{7.3}$$

and it is equal to I on $\delta\Omega$, the boundary of Ω. ΔI is iteratively propagated in the direction of $\nabla^{\perp} I$ until a steady state (7.3) is met.

However, microarray images contain thousands of regions requiring such reconstructions and are, therefore, computationally expensive to examine with the highlighted technique. In an attempt to handle such a time restriction, Oliveira *et al.* [2001] aimed to produce similar results to [Bertalmio *et al.*, 2000] albeit quicker, although the approach may lead to loss of some information in the translation.

7.3.2 CNNIR Algorithm

Again, let Ω be a small area to be reconstructed (inpainted) and let $\partial\Omega$ be its boundary. The small size of the gene spot, Ω, allows the ability to use isotropic diffusion in order to propagates information from one or two layers of pixels from the boundary of the gene spot $\partial\Omega$ into Ω. Therefore, approximating inpainting procedure has been achieved.

The linear isotropic diffusion equation can be directly mapped onto the CNN array resulting in the following simple template [Chua & Yang, 1988d],

$$\begin{aligned} DIFFUS_A &= \begin{bmatrix} 0 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix} \\ DIFFUS_B &= 0; \\ DIFFUS_z &= 0; \end{aligned} \tag{7.4}$$

The simplest version of the algorithm consists of initializing Ω region by clearing Ω and repeatedly convolving the region to be inpainted with a diffusion template. $\partial\Omega$ is a two-pixel thick boundary. The number of iterations is dependent on the size of the mask (Ω). However, This simple algorithm, which is similar to the method in [Oliveira *et al.*, 2001], produces a weak background signal as if it loses something in translation, see Fig. 7.3.

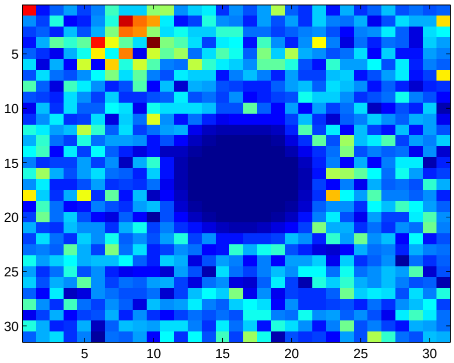


Figure 7.3: Output sample by applying diffusion template on Ω region

7.3.2.1 The Pseudo-Code Of CNNIR (Fast-Inpainting)

In order to overcome this drawback, a CNN algorithm is proposed. A pseudo-code of CNNIR can be found in Algorithm 5.

Algorithm 5 Pseudo-Code of Cellular Neural Network Image Reconstruction Function

```
Require: $\Omega$ {Image: specify gene spot region pixels}
Require: $I$ {Image: the image to be reconstructed}
Ensure: $I_o$ {Image: the reconstructed image(without gene spot signal)}
    $\{\Omega \in \{0,1\}\}$ $\{I \in [-1,1]\}$
1: $\Omega \leftarrow$ CNNDilation($\Omega$), Add two pixel layer to $\Omega$
2: $I \leftarrow$ CNNMask($I$, $\hat{\Omega}$), $\hat{\Omega}$ is The Complementary of $\Omega$
3: while $\Omega$ is not empty, there is '1's pixel or more do
4:     $I \leftarrow$ CNNDiffuion($I$, $\Omega$, $DIFUS$)
5:     $I \leftarrow I +$ CNNMask($I$, $\Omega$)
6:     $\Omega \leftarrow$ CNNErosion($\Omega$), , Peel one pixel layer from $\Omega$
7: end while
8: $I_o \leftarrow$ $I$
```

Essentially, the proposed algorithm takes a mask Ω and the input image I. Note that in this algorithm we deal with each channel separately, thus, we should consider I as either the Cy_5 or Cy_3 image. The mask is an image that marks the spot regions with pixels of value '1'. In the experiments, the mask image is the output of the segmentation algorithm of another CNN algorithm outlined in the previous chapter. The first step in the algorithm is to add a layer of two pixels thick. This layer guarantees the elimination of the effect of the direct spot boundary pixels. The boundary pixels usually contain overlapping information from the spot and the background signal, and thus, it is not a good representative of the background signal in the local area. The lines 4, 5, 6 is the solution for the situation in

Fig. 7.3. By accumulating a proportion of the result of the diffusion process, with peeling one layer from the mask every cycle, we keep the signal's level in the reconstructed area close to the local background.

Fig. 7.4 presents a sample-reconstructed region, which is the same region of Figure. 7.3.

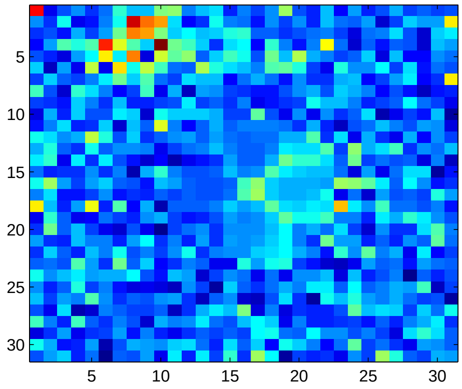


Figure 7.4: Output sample by applying CNNIR algorithm on Ω region

7.3.3 CNN-NSIR Algorithm

In [Bertozzi & Bertalmio, 2001], an inpainting approach has been introduced based on the ideas from classical fluid dynamics to propagate isophote lines continuously from the border into the reconstructed region. The underlying assumption is to think of the image intensity as a 'stream function' for a two-dimensional incompressible flow. The Laplacian of the image intensity plays the role of the vorticity of the fluid, i.e., it is propagated into the inpainted area by a vector field defined by the stream function. The method is directly based on the Navier-Stokes equations for fluid dynamics, which has the immediate advantage of well-developed theoretical and numerical results.

The basic equation for Incompressible Newtonian flow is as follows:

$$\frac{\partial \mathsf{v}}{\partial t} + \mathsf{v}\nabla\mathsf{v} = -\nabla p + \mu\nabla^2\mathsf{v} \tag{7.5}$$

where v is the velocity vector, p is the pressure and μ is the viscosity. For two-dimensional flows, we introduce a stream function Ψ where

$$\nabla^{\perp}\Psi = \mathsf{v} \tag{7.6}$$

Fluid dynamics	**Image processing**
stream function Ψ	Image intensity I
fluid velocity $\mathsf{v} = \nabla^{\perp}\Psi$	isophote direction $\nabla^{\perp}I$
vorticity $\omega = \Delta\Psi$	smoothness $\omega = \Delta I$
viscosity μ	anisotropic diffusion μ

Table 7.1: The counterpart between 2D Navier-Stokes equation and image inpainting

eliminates p, and identically satisfies the divergence free condition, in (7.5). Letting $\omega = \Delta \times \mathsf{v}$, the vorticity, we obtain the vorticity-stream function formulation for the Navier-Stokes equations:

$$\omega_t + \mathsf{v}.\nabla\omega = \mu\Delta\omega \tag{7.7}$$

In the case of near absence of viscosity, i.e. $\mu \approx 0$, we have the steady state solution of (7.7) approaching,

$$\mathsf{v}.\nabla\omega = \nabla^{\perp}\Psi.\nabla\Delta\Psi \approx 0 \tag{7.8}$$

Ψ is a continuous function defined on a continuous domain, while I is like an integer function defined on a discrete domain. Since we will discretize Ψ using finite difference techniques, the latter discrepancy is resolved. For the former case, values are rounded to the nearest integer.

Notice the remarkable similarity between (7.8) and the solution criterion (7.3) for the inpainting problem. Exploiting this fact and replacing Ψ with an image matrix I , we summarize the counterparts between $2D$ incompressible fluid flow and image inpainting in the Table 7.1 below Bertozzi & Bertalmio [2001].

In image processing terms, we have the counterpart to the vorticity-stream function

$$\omega_t + \mathsf{v}\nabla\omega = \mu\nabla.(g(|\nabla\omega|)\nabla\omega) \tag{7.9}$$

where $\Delta I = \omega$ is the vorticity, $\nabla^{\perp}I = \mathsf{v}$ is the direction of the isophotes and $g(.)$ accounts for anisotropic diffusion (or edge preserving diffusion).

7.3.3.1 Designing The Templates

Again, let Ω be a small area to be reconstructed (inpainted) and let $\partial\Omega$ be its boundary. The small size of the gene spot, Ω, allows the ability to use isotropic diffusion in order to propagate information from one or two layers of pixels from the boundary of the gene spot $\partial\Omega$ into Ω. Therefore, approximating inpainting procedure has been achieved.

We define the standard central finite difference operators applied to the grid function u_{ij} below

$$D_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} \tag{7.10}$$

$$-D_y u_{i,j} = -\frac{u_{i,j+1} - u_{i,j-1}}{2h} \tag{7.11}$$

$$\begin{aligned} \Delta_h u_{i,j} &= (D_x^2 + D_y^2) u_{i,j} \\ &= +\frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \\ &\quad +\frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \end{aligned} \tag{7.12}$$

In all the following templates, h is the uniform grid size and R is the value of the state resistor in a CNN cell. In addition, provided that the transient remains bounded (i.e. the cells do not saturate), it is assumed that a CNN array is stable when it starts from a specified initial condition.

The first derivative (I_x) can be directly mapped onto the CNN array resulting in the following simple template ($DERx$: $X(0) =$ ORIGINAL IMAGE, $BC = ZF$):

$$\begin{aligned} DERx_A &= \begin{bmatrix} 0 & 0 & 0 \\ \frac{-1}{2h} & 0 & \frac{1}{2h} \\ 0 & 0 & 0 \end{bmatrix}; \\ DERx_B &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{R} & 0 \\ 0 & 0 & 0 \end{bmatrix}; \end{aligned} \tag{7.13}$$

Similarly, the first derivative ($-I_y$) can be directly mapped onto the CNN array resulting in the following simple template ($DERy_-$: $X(0) =$ ORIGINAL IMAGE, $BC = ZF$):

$$\begin{aligned} DERy_{-A} &= \begin{bmatrix} 0 & \frac{1}{2h} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{-1}{2h} & 0 \end{bmatrix}; \\ DERy_{-B} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{R} & 0 \\ 0 & 0 & 0 \end{bmatrix}; \end{aligned} \tag{7.14}$$

The linear isotropic diffusion equation can be directly mapped onto the CNN array resulting in the following simple template [Chua & Yang, 1988b] ($DIFFUS$: $X(0) =$ ORIGINAL IMAGE, $BC = ZF$)

$$DIFFUS_A = \begin{bmatrix} 0 & \frac{1}{h^2} & 0 \\ \frac{1}{h^2} & \frac{-4}{h^2} + \frac{1}{R} & \frac{1}{h^2} \\ 0 & \frac{1}{h^2} & 0 \end{bmatrix} \tag{7.15}$$

There is a considerable number of methods for numerical integration. One of the best known techniques is the Newton-Côtes method that is based on a polynomial interpolation on equally-spaced points. This method can be transformed into integration rules using polynomials of any order giving an error that decreases faster and faster with the number of points being used as higher-order polynomials are chosen [Luchini, 1984]. In the light of Luchini's work [Luchini, 1984] and by applying the closed Newton-Côtes formula (Simpson's rule)[Stoer & Bulirsch, 2002],

$$\begin{aligned}\int_{-\delta_1 h}^{(N+\delta_2)h} f(x)\mathsf{d}x \approx\ & +h\{w_{ext}(\delta_1)f(-h) + w_{int}(\delta_1)f(0) \\ & + \sum_{i=1}^{N-1} f(ih) \\ & + w_{int}(\delta_2)f(Nh) + w_{ext}(\delta_2)f((N+1)h)\} \end{aligned} \tag{7.16}$$

where

$$w_{int}(\delta) = 7/12 + \delta - \delta^2/2 \tag{7.17}$$

$$w_{ext}(\delta) = -1/12 + \delta^2/2 \tag{7.18}$$

Equation (7.16) is an integration formula [Luchini, 1984], with equally-spaced points for an interval not ending exactly in any of the points where $f(x)$ is known. Therefore, we can numerically compute the integration over x (or y) axis based on the spatial information and the following template ($INTx$: $X(0) =$ ORIGINAL IMAGE, $BC = ZF$)

$$INTx_A = \begin{bmatrix} 0 & 0 & 0 \\ 0.4h & 0 & 0.4h \\ 0 & 0 & 0 \end{bmatrix}; \tag{7.19}$$

$$INTx_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2h + \frac{1}{R} & 0 \\ 0 & 0 & 0 \end{bmatrix};$$

Finally, a CNN approximation of the NSE (7.9) can be created using a two-layer CNN, see the solution of NSE for incompressible fluids in [Kozek & Roska, 1996]. With the continuity condition in rectangular co-ordinates, the equation can be written as

$$\frac{\partial u}{\partial t} = \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) - \frac{\partial u^2}{\partial x} - \frac{\partial uv}{\partial y} \tag{7.20}$$

$$\frac{\partial v}{\partial t} = \mu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) - \frac{\partial uv}{\partial x} - \frac{\partial v^2}{\partial y} \tag{7.21}$$

Similar to the previous solutions, to obtain a spatially discrete system, spatial derivatives are replaced with difference terms to yield the approximate expression for Equation (7.20) (similarly for (7.21))

$$\begin{aligned} \frac{\partial u}{\partial t} = & + \mu(\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2}) \\ & + \mu(\frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2}) \\ & - (\frac{-(u_{i-1,j})^2 + (u_{i+1,j})^2}{2h}) \\ & - (\frac{-(u_{i,j-1}v_{i,j-1}) + (u_{i,j+1}v_{i,j+1})}{2h}) \end{aligned} \tag{7.22}$$

With the developed rational in [Kozek & Roska, 1996], we can evaluate v in the Equation (7.5) using the CNN template (7.23) for u component (those for the v component can be generated analogously) (NSE: $X(0) =$ ORIGINAL IMAGE, $BC = ZF$)

$$\begin{aligned} NSEA_{uu}(ij,kl) &= \begin{bmatrix} 0 & \frac{\mu}{h^2} & 0 \\ \frac{\mu}{h^2} & \frac{-4\mu}{h^2} + \frac{1}{R} & \frac{\mu}{h^2} \\ 0 & \frac{\mu}{h^2} & 0 \end{bmatrix}; \\ NSE\hat{A}_{uu}(ij,kl) &= \begin{bmatrix} 0 & 0 & 0 \\ \frac{-1}{2h} & 0 & \frac{1}{2h} \\ 0 & 0 & 0 \end{bmatrix} [u_i.u_k]; \\ NSE\hat{A}_{uv}(ij,kl) &= \begin{bmatrix} 0 & \frac{-1}{2h} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2h} & 0 \end{bmatrix} [u_j.v_l]; \end{aligned} \tag{7.23}$$

the lower right indexes of the templates indicate from which layer and to which layer they connect.

Remark 6. *All the derived templates have been tuned to be stable in the gray-scale area* *(the linear area in the output function, see [Chua & Roska, 2002, Chapter 6]). Therefore, after a specific transient time elapses, the output of any specific operation would be either the state value or the output value.*

7.3.3.2 The Pseudo-Code of CNN-NSIR

As discussed previously, a common drawback with the existing microarray image processing methods is that they cannot properly address the quantification of the gene spot in a realistic way without any assumption about the image surface. In order to overcome this drawback, a CNN algorithm, named as CNN-NSIR, is proposed in this paper with the pseudo-code given in Algorithm 6.

Algorithm 6 CNN-NSIR Algorithm

```
**Require:** $\Omega$ {Image: specify gene spot region pixels}
**Require:** $I$ {Image: the image to be reconstructed}
**Ensure:** $I_o$ {Image: the output image}
  $\{\Omega \in \{0,1\}\}$ $\{I \in [-1,1]\}$
$\Omega \leftarrow$ CNNDilation($\Omega$, 2) {Add two pixel layer to $\Omega$}
$I \leftarrow$CNNMask($I$, $\hat{\Omega}$) {$\hat{\Omega}$ is The Complementary of $\Omega$}
$I \leftarrow$ CNNDiffusion($I$)
Set $\alpha$ {$\alpha$: # of NSE evaluations}
Set $\beta$ {$\beta$: transient time of NSE evaluation}
$\grave{\Omega} \leftarrow$ CNNDilation($\Omega$, 1)
$u \leftarrow$ CNNget_u($I$, $\partial\grave{\Omega}$, $DERy_-$)
$v \leftarrow$ CNNget_v($I$, $\partial\grave{\Omega}$, $DERx$)
**for** $i = 1$ to $\alpha$ **do**
    $[U,\ V] \leftarrow$ CNNnse($u$, $v$, $\grave{\Omega}$, $\beta$) {Propagate $\partial\grave{\Omega}$ into $\grave{\Omega}$}
    $I_1 \leftarrow$ -CNNIntegration($U$, $INTy$)
    $I_2 \leftarrow$ CNNIntegration($V$, $INTx$)
    $I \leftarrow I_2 - I_1$
    $\grave{\Omega} \leftarrow$ threshold($I$) {returns 1 where cells' values equal 0}
    $\grave{\Omega} \leftarrow$ CNNDilation($\grave{\Omega}$, 1)
    $u \leftarrow$ CNNget_u($I$, $\partial\grave{\Omega}$, $DERy_-$)
    $v \leftarrow$ CNNget_v($I$, $\partial\grave{\Omega}$, $DERx$)
**end for**
$I_o \leftarrow$ $I$
```

Essentially, the proposed algorithm takes a mask Ω and the input image I. Note that, in this algorithm, we deal with each channel separately and, thus, we should consider I as either the Cy_5 or Cy_3 image. The mask is an image that marks the spot regions with pixels of value '1'. The first step in the algorithm is to add a layer of two-pixels thick.

This layer guarantees the elimination of the effect of the direct spot boundary pixels. In the segmentation result, the boundary pixels usually contain overlapping information from the spot and the background signal. Therefore, it is not a good representative of the background signal in the local area. Figure 7.5 presents a sample-reconstructed region.

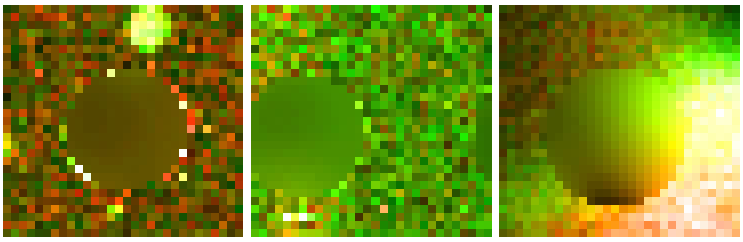


Figure 7.5: Samples of three different background signals. The reconstructed images shows an acceptable background trend estimation. The third example represents the effect of the noise artefacts on the spot's area.

It should be mentioned that the spot data of the microarray are different from the natural pattern, or usual image that would be the ordinal target for the reconstruction technique. The boundary of the spot in the microarray data would usually show no-correlation while, in the natural image, the pattern or arrangement is important to express the meaning. However, in our algorithm, a smoothing operator has been applied in Algorithm 1 [line 3]. This operator can achieve the smoothness to a degree that is enough for this application. As can be seen from Figure 7.5, even the random pattern in the surrounding area of the spot has been approximated very well in the reconstructed area.

7.4 Discussion

7.4.1 Notes About The Algorithms

Due to the high signal variability that exists across the microarray surface, when working directly with the raw microarray information, propagating the information of the border region $\partial\Omega$ along a level line (isophotes) would be impotent. Therefore, rather than using the raw image, it is suggested that producing a smoothed version of the image data would not only be advantageous, but more effective in terms of the overall goal. In our algorithm, the $DIFFUS$ template has been used as smoothing operator. Although this isotropic diffusion template causes blurring effects, it can achieve the required refining result by calculating the regions (local) average intensities.

Anisotropic diffusion would be a better alternative for the diffusion operator, not only as smoothing operator (see line 3 in Algorithm 1) but in evaluating NSE as well. However, the anisotropic diffusion models require a noise-level estimate K that determines the magnitude of the edges to be conserved during the smoothing process. K could be set according to the priori knowledge about the noise statistics or could be estimated from the absolute gradient histogram [Canny, 1986; Perona & Malik, 1990]. However, in a locally connected parallel processing architecture, it would be very difficult to calculate these values. Thus, other approaches should be sought to achieve this target. Rekeczky *et al.* [1998b] proposed a possible method to estimate the noise-level roughly as the minimum of the maximal local variations at nodes of the coarse-grid model.

To evaluate the outcome of the proposed approach, the results have been compared with the output image by another algorithm that is dedicated specifically for reconstructing (inpainting) image. These algorithm is Bertalmio's method [Bertalmio *et al.*, 2000]. Although the isotropic diffusion operator has been applied as a first step in Bertalmio's algorithm (and anisotropic diffusion thereafter), yet, the microarray characteristics cause a very long settling time. CNNIR method uses the information of one bit thickness layer to reconstruct the spot's area. Even though CNNIR is remarkably faster than CNN-NSIR, it causes more spots to be considered as a bad region and therefore omitted in later analyzes.

7.4.2 Dataset Characteristics

The images used in this paper are derived from the human gen1 clone set data. These experiments were designed to contrast the effects of two cancer inhibiting drugs (PolyIC and LPS) over two different cell lines. One cell line represents the control (untreated) and the other the treatment (HeLa) line over a series of several time points. In total, there are 47 distinct slides with the corresponding GenePix results presented. Each slide consists of 24 gene blocks with each block containing 32 columns and 12 rows of gene spots. The gene spots in the first row of each odd-numbered block are known as the Lucidea ScoreCard [Samartzidou *et al.*, 2001] and consist of a set of 32 pre-defined genes that can be used to test various experiment characteristics.

7.4.3 Evaluation

In order to quantify the performance capabilities of our technique, a quality measure is required to allow the judgment of how the estimated background affects the quantification of the gene spot. For this purpose, a systematic objective method, which is based on the descriptive statistic Interclass Correlation Coefficient (ICC) measures, see Section 4.5.3, is used to compare the results produced by different techniques. The rational is justified as follows. The set of 32 pre-defined genes is used in the comparison process. Using these controls, we base our analysis on the following assumptions: 1) the better the reconstruction, the higher the correlation within the same control should be (minimum σ_e^2); 2) the better the reconstruction, the lower the correlation between the genes within the array should be (maximum σ_g^2); and 3) the better the reconstruction, the higher the ICC value should be.

Fig. 4.12 presents the estimated variance components and ICC for the dataset images and on average. The reliabilities of all methods are high, with CNN-NSIR method appearing on average to be more reliable than the other methods. Note that even the within-spot variability $\hat{\sigma}_e^2$ (the noise) is notably smaller for CNN-NSIR method, though the between-spot variability (the signal) is bigger for CNNIR method.

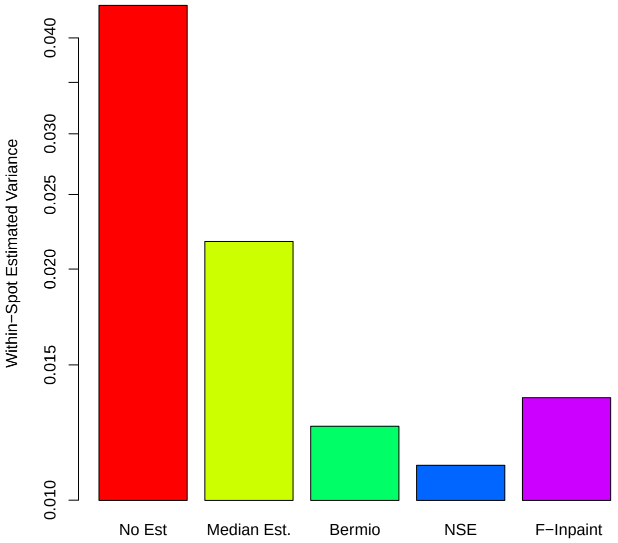


Figure 7.6: Within-Spot estimated variance over the dataset. (The methods are ‘No Background Correction’, ‘Median Background Estimation’, ‘Bertalmio’, ‘CNN-NSIR’ and ‘CNNIR(f-Inpaint)’.

It should be pointed out that the blind algorithm (operator independent algorithm) limits our ability to discriminate between bad spots from the good ones. However, this will give better insight about the robustness of the implemented methodology. In our analysis, every signal less than 100 is considered to be a bad reading and, consequently, omitted from the analysis.

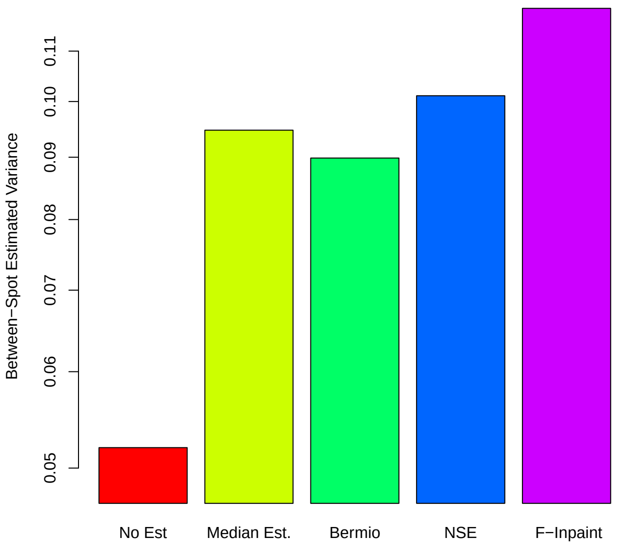


Figure 7.7: Between-Spots estimated variance over the dataset. (The methods are 'No Background Correction', 'Median Background Estimation', 'Bertalmio', 'CNN-NSIR' and 'CNNIR(f-Inpaint)'.

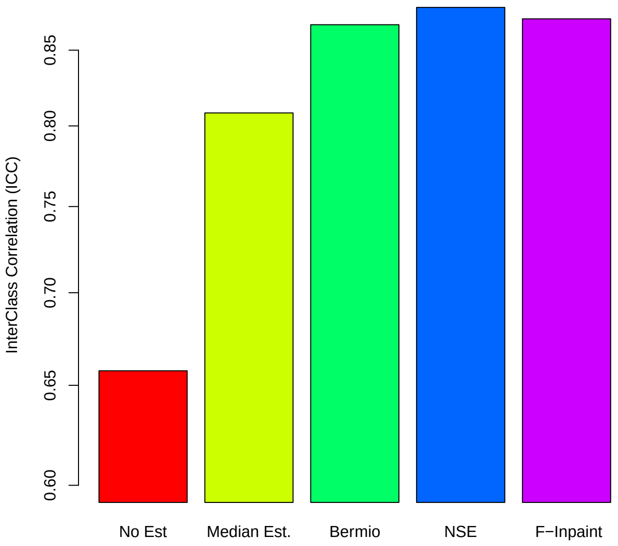


Figure 7.8: Average ICC over the dataset. (The methods are 'No Background Correction', 'Median Background Estimation', 'Bertalmio', 'CNN-NSIR' and 'CNNIR(f-Inpaint)'.

Figure 7.9 gives more detailed plots about the data. Figure 7.9-(a) shows that not only the range of the signal has been changed due to the reconstruction algorithm application, but the average values have also been changed as well. The distribution of the signals, after applying the reconstruction methods (Median Background Estimation, Bertalmio's, CNNIR and CNN-NSIR), have a narrower normal distribution with the data skewed towards the lower side. Note that 50% of the gene' values fall within 0.1 range around 0.5 for our method. Furthermore, while the middle 50% values of ("No Est") fall into 0.4 range width and the whole range is 1.1, the middle 50% values of (CNN-NSIR) fall into 0.7 range width and the whole range is 1.6, see Figure 7.9-(b).

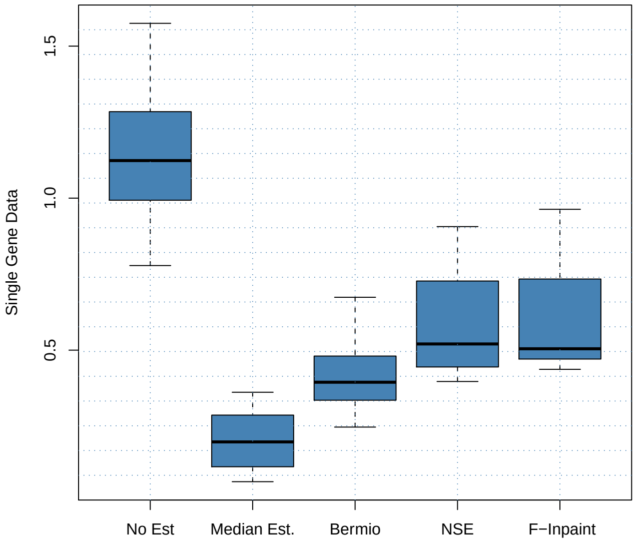


Figure 7.9: Boxplot for one sample gene.

7.5 Conclusion

In this chapter, we have presented a novel image reconstruction framework that attempts to improve the quantification results of the microarray image. Specifically, the framework consists of several components that process a microarray image based on a given mask (could be the output of any automated segmentation process) without human intervention.

The CNN algorithms for Network Image Reconstruction, outlined in Algorithms 5-6, have been found to have the following advantages over current implementations:

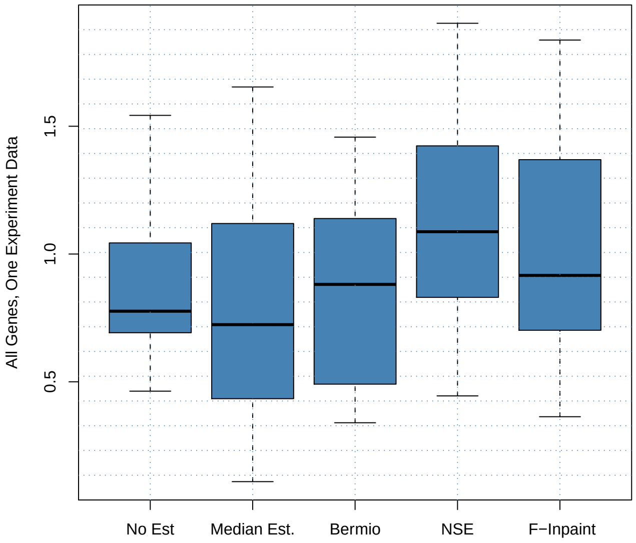


Figure 7.10: Boxplot for one experiment, image, data (All Genes).

- The importance of computational fluid dynamics and Navier-Stokes equations in the application of image inpainting has been highlighted in [Bertozzi & Bertalmio, 2001]. On the other hand, cellular neural networks approximation of the Navier-Stokes equation describing the viscous flow of incompressible fluids is proposed in [Kozek & Roska, 1996]. Hoverer, to the best of our knowledge, a cellular neural network algorithm that connects and applies both ideas is novel.

- The cellular neural network algorithm, based on isotropic diffusion model, has been applied. Although it is a simple model, its results are comparable to the other proposed nonlinear algorithm. However, CNNIR algorithm has two advantages over the CNN-NSIR. The first is that CNNIR is much simpler and any CNN-UM simple hardware can be used to execute it. The second is that CNNIR is considerably faster than CNN-NSIR, which is minor advantage.

- Both algorithms exploit characteristics of cellular neural network. This paradigm can be used to approximate partial differential equations by introducing finite differences.

- The proposed algorithm achieves the reconstruction of the microarray in a simple yet robust way. The dataset contains slides from experiments that were conducted in different time points. However, the wide range of spots' areas, regarding the characteristics of the background of each spot, have proved the ability of the proposed algorithms to achieve a high performance under many different conditions.

On the other hand, testing the algorithms on different set of experiments' slides, produced using different protocols for instance, is expected to increase the confidence in the results of the algorithms.

- The algorithms are an operator-free methods that take only the raw data and a mask as the input. The requirement of human intervention would lead to an inefficient process by not only increasing the processing time but by also introducing the human-factor error.
- The algorithm can be applied on CNN-UM [Roska & Chua, 1993] and therefore allows the researchers to process the image itself and get the quantitative data for further analysis not only in efficient time but also with remarkably high accuracy. However, CNN-UM is not the only available alternative as a target for the algorithm. Currently, Graphics Processing Unit (GPU) is a very efficient alternative that can be considered as a good candidate to apply CNN algorithms in a remarkably fast time, without any extra cost.
- The potential of the algorithm is proven based on the direct comparisons between our proposed approaches with other methods such as Bertalmio's method and median background estimation method.

With the inpaiting step, we should have got the best possible information from the microarray image. The next step would be collecting some statistics about the spots and background signals for further statistic analysis.

Chapter 8

Conclusions & Future Work

8.1 Introduction

The development of biomedical research has been led by the increasing knowledge as well as new advances in technology. Traditionally, researchers were able to investigate a small number of genes at a time by using the available techniques back then. During the 1990s, many new technologies emerged with a huge potential for tackling problems, which were unfeasible previously. DNA microarray technology has enabled biologists to study all the genes within an entire organism to obtain a global view of genes' interaction and regulation. This technology has a great potential in obtaining a deep understanding of the functional organization of cells. However, it is still early in its development, and needs improvements in all the main stages of the microarray process. The emergence of this technology allows the researchers to tackle difficult problems and reveal promising solutions in many fields, i.e. pharmaceutical and agricultural industries. To name some of the applications, we might mention tumor classification, prognosis prediction, drug development, therapy development and tracking disease progression. Because they allow researchers to the genes' functions in tissues that are subjected to a medication being tested. Moreover, drug companies are often interested in altering some protein to prevent the faulty gene behavior from causing a disease.

The typical microarray image can contain information about a thousands of genes' spots. Besides the analysis required for huge volume of information produced by such experiments, one of the main concerns is the quality of the image. Although microarray technology has been engineered to a high tolerance, the microarray image suffers a considerable amount of noise, in spite of which a very valuable data still to be extracted from these images. These errors and noises will propagate down through, and can significantly affect, all subsequent processing and analysis. Therefore, to realize the potential of such technology, it is crucial to obtain high quality image data that would indeed reflect the underlying biology in the samples. One of the key steps in extracting information from a microarray image is segmentation: identifying which pixels within an image represent which gene. This area of spotted microarray image analysis has received relatively little attention relative to the advances in proceeding analysis stages. But, the lack of advanced image analysis including the segmentation results in sub-optimal data being used in all downstream analysis methods.

In this chapter, it is our aim to overview the whole research conducted through the chapters of this thesis in order to highlight the contributions as well as the possible improvements for future work. The main motivation for this research is to tackle the challenges involved with the microarray image processing. Furthermore, recent interest in achieving the biological cDNA microarray experiments in a fully automated way, on one hand, and the advances in neural network hardware implementation, on the other, led our ambition to investigate the possibility of analyzing the microarray image with this up-to-date paradigm. The application of the proposed techniques showed not only robust and efficient methods but also also a remarkably faster implementation.

The investigation's main components cover three areas, which are filtering, segmentation and reconstruction stages. For every stage, the initial analysis of currently available techniques sets some rules that are based on the strong and weak points. This analysis sets the road maps for all the presented improvements. The improvements were either novel algorithms, which deal with all and every specific parts, or developments of some available methods and integrating them with the whole proposed system.

In the following, Section 8.2 reviews the achievements and contributions in respect to the main components as discussed throughout the thesis. The final Section 8.3 reflects on possible limitation of the proposed methods and highlights some key topic for future research.

8.2 Achievements and Contributions

The discussion in Chapter 2 reviews the main methodologies which have been applied to process the microarray image. Although lots of researches have been done to achieve an automated methodology, until recently, the human operators played a vital role in the process as a whole. In this respect, our research based heavily on the requirement for a reliable yet time efficient automated method.

The primary contribution of this research based on the conclusion of Chapter 2 can be regarded as a fully automated cDNA microarray image processing system. However, before highlighting the contributions, a summary of each chapter will be represented.

One of the key steps in extracting information from a microarray image is the segmentation whose aim is to identify which pixels within an image represent which gene. This task is greatly complicated by noise within the image and a wide degree of variation in the values of the pixels belonging to a typical spot. In the past, there have been many methods proposed for the segmentation of microarray image. In Chapter 3, a new method utilizing a series of artificial neural networks, which are based on multi-layer perceptron (MLP) and Kohonen networks, is proposed. The proposed method is applied to a set of real-world cDNA images. Quantitative comparisons between the proposed method and the commercial software GenePix are carried out in terms of the peak signal-to-noise ratio (PSNR). This method is shown to not only deliver results comparable and even superior to existing techniques but also have a faster run time.

Chapter 4 is concerned with improving the processes involved in the analysis of microarray image data. The main focus is to clarify an image's feature space in an unsupervised manner. Rather than using the raw microarray image, it is suggested that producing filtered versions of the image data by applying nonlinear anisotropic diffusion, so that the dynamic range of the image could be increased, and hence, a better ability of signal extraction could be achieved. Therefore, a novel segmentation algorithm is proposed. This algorithm is based on Cellular Neural Network computational paradigm integrated with median and anisotropic diffusion filters. The AnaLogic CNN Simulation Toolbox for MATLAB (InstantVision Toolboxes for MATLAB) is used during the segmentation process. Quantitative comparisons among the proposed methods and GenePix are carried out in terms of objective and subjective point of views. It has shown that analogic algorithm integrated with Complex Diffusion filter is the best one to be applied to achieve the segmentation.

The main focus in Chapter 5 is to clarify an image's feature space in a fully automated algorithm. Thus, instead of employing the raw microarray image, it is suggested that producing multiple views of the image data, such that, emphasis is placed on certain frequencies or regions of interest would not only be advantageous, but also more effective in terms of the overall goal. Therefore, a multi-view analysis system combined with Median, Top-Hat and complex diffusion filters is investigated. The proposed image processing methods are applied to a set of real-world cDNA images. The AnaLogic CNN Simulation Toolbox for MATLAB is used during the segmentation process. Quantitative comparisons among different filters are carried out in terms of the peak signal-to-noise ratio (PSNR). It is shown that the CLD filter is the best one to be applied with the image transformation engine.

In Chapter 6, an unsupervised CNN algorithm for the segmentation of microarray image was presented. AN improved approximation method was proposed in order to estimate any non-linear function using a CNN output function PWL. Comparison has shown the remarkable results which outperform the results of the other method. On the other hand, the multi-view filtering technique was integrated with an elastic segmentation algorithm, which based on the linear diffusion and locally adaptive threshold technique. The algorithm was applied on a real microarray image. An objective comparison with GenePix results suggested that this approach produces remarkably good outcomes in terms of the robustness and efficiency.

Apart from our approach, some hardware implementations for microarray image processing have been proposed in the literature. Although they represent a potential alternative for the currently available software systems, they suffer an improper genes' quantification approach. Therefore, Chapter 7 presented novel Image Reconstruction algorithms using the cellular neural network paradigm. The underlying principles of this approach based on linear diffusion and Navier-Stokes equation. This algorithm offers a robust method for estimating the background signal within the gene spot region. The MATCNN Toolbox for Matlab is used to test the proposed method. Quantitative comparisons are carried out, in terms of objective criteria, between our approach and some other available methods. It is shown that the proposed algorithms give highly accurate and realistic measurements in a fully automated manner within a remarkably efficient time.

8.2.1 Contribution 1: Total Automation and a Totally Blind Process

Minimizing the operator roles in the image processing can be sought in two respects shown in Chapter 2. The number and characteristics of the spots complicate the process considerably, on the one hand. On the other hand, the high degree of noise reduces the ability to specify the spots's signal. These issues result in the time-consuming manual, or semi-automated, processing of the microarrays. Therefore, the fully automated algorithm will not only guarantee time efficiency but also will produce more accurate results.

Many methods have been proposed to accomplish different functions in microarray image processing. However, many of those methods suffer either the computation complexity or the need for some manual input from the operator, sometime both are required. The operator's inputs are highly unfavorable since they affect the throughput of the process and introduce user bias across multiple images.

Based on these conclusions, one principal contribution in this thesis is the production of novel fully automated algorithms for the key areas highlighted above. Furthermore, the developed algorithms require no prior assumption about the image or the spots. Thus, another key contribution is that the algorithms are flexible and can be easily adapted for any developments in the cDNA microarray image technology.

8.2.2 Contribution 2: Adopting Local Strategy Algorithms

Investigating the surface of the images, which are available in our dataset, showed a huge diverse in the intensity's values. Therefore, the global information will be of a minimum benefit if there is a benefit at all. In addition, utilizing the global data will result in a costly computation algorithm. On the other hand, relatively local information is enough to achieve various tasks required for the microarray images since it is sufficient for a highly effective and robust yet time efficient algorithms. Therefore, one of the contributions in this thesis is that the algorithms heavily depend on the local data to perform all the covered tasks, i.e., filtering, segmentation and reconstruction. Note that even with the multi-view analysis, where pixel-wise mapping functions have been used, the local information was of utter importance for the algorithm.

8.2.3 Contribution 3: Image Filtering and Noise Reduction

The existence of the flaws in the microarray images has an impact on our ability to process these images. Besides, it is highly important to keep the throughput feature of the microarray technology. Therefore, any successful system should employ an effective automated filtering algorithm. This algorithm enhances the image's quality without any significant data loss. That is, it is much more rational to produce filtered versions of the image in order to increase the dynamic range and then to enhance the signal extraction process.

Regarding the filtering stage, the main contribution was the introduction of a fully dynamic and highly efficient algorithm, Chapter 6. This algorithm integrates many individual filters in one system so that it does not only improve the spots' positions, but also it does that without any suppositions about the image. Therefore, The algorithm not only offers a good method for filtering current microarray images, but also it can cope with any new development in the cDNA microarray production process.

8.2.4 Contribution 4: Image Segmentation

The final goal of the microarray processing is to get a set of statistical values that characterize the gene expression. However, in order to get sufficiently good results, a good methodology for separating the spot signal and the background signal is prominent. Although the segmentation algorithm was regarded rather irrelevant when some background correction algorithm is applied, it is, conversely, proved that the applied segmentation procedure has a huge implication on the final log ratios of spots.

Part of the contributions in this thesis was the introduction of two now segmentation algorithms. Chapter 3 presented a neural network approach developed with a novel topology structure that allows the manifestation of our conclusions about locality and robustness. Chapters 4-6 showed a developed algorithm, where a diffusion based filter was integrated with a locally adaptive thresholding technique and an advanced meta analysis procedure.

8.2.5 Contribution 5: Image Reconstruction

In general, the main procedure after the quantification stage is the normalization process, which is introduced to get the statistics of the spots as correct as possible. Currently available normalization algorithms lie within two categories, either local procedures or global procedures. However, most of these methods assume some propositions in order to justify their results with no access to the real data of the image; i.e., they are built solely on the statistical summaries.

Image reconstruction might be one of the best techniques that can be applied for a more specific background determination process. In such method, some local pixels that are most similar to the known border replace selected pixels within the spot. The basic rational behind it is that the similarities with the border of the spot guarantee the evolution of the local background into the spot region. Therefore, Chapter 7 presented a major contribution by the introduction of novel CNN algorithms for local strategy microarray image reconstruction.

8.2.6 Contribution 6: Cellular Neural Networks and PDEs Application

There has been recently much interest in a fully automated microarray experiment. In such automation, the whole experiment is curried out in one devise such that the human operator has no effects of the final slide produced by such devises. In addition, the need for fully automated and resilient algorithm for the image processing has remarkably increased. The need for automated process is led by two important factor which are the cost which is yet a relatively high as well as the tendency to reserve the throughput of this technology.

On the other hand, neural network paradigm has become a potential approach to achieve a hardware implementation for any designed algorithm. Cellular neural network, in particular, offers a remarkable tool for image processing applications.

Therefore, as a main goal in this thesis, the development of CNN algorithm for various stages in the microarray image processing is a major contribution. Over many chapters, from 4 to 7, the design and application of a robust and effective CNN algorithm were in mind. The filtering proposed methodologies showed the ability of utilizing some findings from PDEs based filters, which can be applied easily on the CNN-UM. The segmentation stage was developed with a good consideration about the diffusion PDE equation. Finally, Navier-Stokes PDE equations were used for the development of the reconstruction algorithm.

The main advantage of the CNN is the parallel computing ability, on one hand, and the implementation potential on either CNN-UM or GPU (such as CUDA), on the other.

8.3 Future Work

All the proposed algorithms have performed extremely well over the available dataset. However, the algorithms presented in this thesis are not perfect and further investigation may yield much more improvements in design and implementation. The following highlights certain points that can be the topics for future research.

Although the training algorithm BP in Chapter 3 is able to build some rules which reign the classification procedure, it is highly recommended that a thorough investigation should be done to cover the generalization properly. One possible improvement is the application of Maximum Covariance Technique. Another possible route is the introduction of multiple hidden layer. Furthermore, the one-dimension Kohonan network can be replaced with 2-dimensional network. Such formation is more suitable for inferring some intrinsic statistical patterns within the input set.

The algorithms, presented in Chapter 4, are remarkably successful. However, a further improvement can be achieved by employing two or more thresholding templates with different bias value. The outputs of this stage should be fed into a more complicated meta level algorithm. The sophistication of the binary analysis algorithm can lead to more robustness as a tool for detecting the spot signal within high noisy areas.

In Chapter 6, a fully automated CNN algorithm is proposed. However, the main diffusion equation is the linear diffusion. The application of multiple layer CNN algorithm for conducting anisotropic diffusion might be beneficial. On the other hand, more research should be devoted the development of CNN algorithm that approximate the linear complex diffusion equations. Although such algorithm would be much more complicated, computation-wise, and will require much more complicated hardware requirements, linear complex diffusion showed a brilliant performance over the other applied filters.

Finally, one remarkable observation has been developed last two years while conducting the research presented in this thesis. Significantly, there is a possibility to approximate the grayscale morphological operators using a series of PDEs. In addition, the structure of CNN was proven to be highly successful in approximating and solving PDEs. On the other hand, the research community does not have a systematic methodology which is suitable for developing CNN algorithm effectively, rather more heuristic approaches are usually followed. One starting point for further research might be exploring the ability to develop a simple CNN programming block based on morphological PDEs. This simple block might be used later for developing a sophisticated CNN algorithm for challenging applications such as microarray image processing. Another possible route is to establish a better understanding of the effects of these methodology on the speed of process by conducting a quantitative research using the CNN-UM and the available GPU hardware.

References

Abrishambaf, R., Demirel, H. & Kale, I. (2008). A fully CNN based fingerprint recognition system. In *Cellular Neural Networks and Their Applications, 2008. CNNA 2008. 11th International Workshop on*, 146-149, IEEE, Santiago de Compostela. 77

Acton, S. & Crawford, M. (1992). A mean eld solution to anisotropic edge detection of remotely sensed data. In *Proceedings 12th International Geoscience and Remote Sensing Symposium*, 845-847, Houston, USA. 82

Acton, S., Bovik, A. & Crawford, M. (1994). Anisotropic diffusion pyramids for image segmentation. In *Proceedings of 1st International Conference on Image Processing*, 478-482, IEEE Comput. Soc. Press, Austin, TX , USA. 82

Acton, S.T. (1998). Multigrid anisotropic diffusion. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, **7**, 280-91. 83, 84

Adams, M.D., Soares, M.B., Kerlavage, A.R., Fields, C. & Venter, J.C. (1993). Rapid cDNA sequencing (expressed sequence tags) from a directionally cloned human infant brain cDNA library. *Nature Genetics*, **4**, 373-380. 10

Adams, R. & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 641-647. 33

Ahuja, N., Rosenfeld, A. & Haralick, R. (1980). Neighbor gray levels as features in pixel classification. *Pattern Recognition*, **12**, 251-260. 34

Aizenberg, I., Aizenberga, N., J.Hiltnerb, Moragab, C. & Bexten, E.M.Z. (2001). Cellular neural networks and computational intelligence in medical image processing. *Image and Vision Computing*, **19**, 177-183. 77

Akbari, A. & Albregtsen, F. (2003). Normalizing The Background And Removing The Trend In One-Dimensional Dna Fingerprint Images. *Journal of Chromatography A*, **1014**, 11-19. 129

Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X. & al., E. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503-511. 11, 19

Allain, L., Askari, M., Stokes, D. & Vo-Dinh, T. (2001). Microarray sampling-platform fabrication using bubble-jet technology for a biochip system. *Fresenius' Journal Of Analytical Chemistry*, **2**, 146-150. 24

Allen, G.E. (1978). *Thomas Hunt Morgan: the man and his science*. Princeton University Press, Princeton, USA. 13

Alon, U. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, **96**, 6745-6750. 19

Alparone, L., Barni, M., Bartolini, F. & Cappellini, V. (1996). Adaptively weighted vector-median filters for motion-fields smoothing. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 2267-2270, IEEE. 91

Alvarez, L., Lions, P.L. & Morel, J.M. (1992b). Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. II. *SIAM Journal on Numerical Analysis*, **29**, 845-866. 84, 85

Alvarez, L., Guichard, F., Lions, P.L. & Morel, J.M. (1993). Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, **123**, 199-257. 30, 91

Angulo, J. & Serra, J. (2003). Automatic analysis of DNA microarray images using mathematical morphology. *Bioinformatics*, **19**, 553-562. 4, 34

Anonymous (1999). *GenePix 4000 A User's Guide*. 32, 142

Anonymous (2001). Lucidea Microarray ScoreCard: User's Guide v1.1. 99

Anonymous (2008). *ImaGene 6.1 User Manual*. 33

Arce, G. (1998). A general weighted median filter structure admitting negative weights. *IEEE Transactions on Signal Processing*, **46**, 3195-3205. 91

Arce, G.R., Gallagher, J.N.C. & Nodes, T.A. (1986). *Median filters: theory and aplications*, vol. 2, 378. Elsevier Science & Technology, Oxford, UK. 91

Arena, P., Bucolo, M., Fortuna, L. & Occhipinti, L. (2002a). Cellular neural networks for real-time DNA microarray analysis. *IEEE Engineering in Medicine and Biology Magazine*, **21**, 17-25. 3, 4, 36

Arena, P., Fortuna, L. & Occhipinti, L. (2002b). A CNN algorithm for real time analysis of DNA microarrays. *IEEE Transactions onCircuits and Systems I: Fundamental Theory and Applications*, **49**, 335-340. 143

Athanasiadis, E., Cavouras, D. & Glotsos, D. (2009). Segmentation of complementary DNA microarray images by wavelet-based Markov random field model. *IEEE Transactions on Information Technology in Biomedicine*, **13**, 1068-1074.

Baggerly, K., Mitra, R., Grier, R. & Medhane, D. (2004). Comparison of sample-labeling techniques in DNA microarray experiments. *Analytica Chimica* , **506**, 117-125. 22, 26

Bajcsy, P. (2004). Gridline: automatic grid alignment DNA microarray scans. *IEEE Transactions on Image Processing*, **13**, 15-25. 5, 32, 108

Bajla, I., Marušiak, M. & Šrámek, M. (1993). *Anisotropic filtering of MRI data based upon image gradient histogram* -, vol. 719, 90-97. Springer-Verlag, Berlin / Heidelberg. 82

Bassett, D., Eisen, M. & Boguski, M. (1999). Gene expression informatics - it's all in your mine. *Nature genetics*, **21**, 51-55. 143

Bateson, W. (2007). *Mendel's Principles of Heredity*. Cosimo, Inc. 12

Bengtsson, A. & Bengtsson, H. (2006). Microarray image analysis: background estimation using quantile and morphological filters. *BMC bioinformatics*, **7**, 96. 38

Bergholm, F. (1987). Edge Focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-9**, 726-741. 84

Bertalmio, M., Sapiro, G., Caselles, V. & Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 417-424, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA. 142, 144, 146, 147, 156

Bertozzi, A. & Bertalmio, M. (2001). Navier-Stokes fluid dynamics and image and video inpainting. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, **1**, 355-362. 143, 144, 149, 150, 160

Bertucco, L., Nunnari, G., Randieri, C., Rizza, V. & Sacco, A. (1998). A cellular neural network based system for cell counting in culture of biological cells. In *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No.98CH36104)*, 341-345, IEEE, Trieste , Italy. 77

Beucher, S. (1982). Watersheds of functions and picture segmentation. In *ICASSP 82, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1928-1931, Paris. 4

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press. 42, 48, 49, 61

Biswas, S. (1996). Smoothing of digital images using the concept of diffusion process. *Pattern Recognition*, **29**, 497-510. 83

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A. & Others (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536-540. 19

Black, M., Sapiro, G., Marimont, D. & Heeger, D. (1998). Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, **7**, 421-432. 84, 86

Blanchard, A., Kaiser, R. & Hood, L. (1996). High-density oligonucleotide arrays. *Biosensors and bioelectronics*, **11**, 687-690. 24

Blekas, K., Galatsanos, N.P. & Georgiou, I. (2003). An unsupervised artifact correction approach for the analysis of DNA microarray images. In N.P. Galatsanos, ed., *IEEE International Conference on Image Processing (ICIP 2003)*, vol. 2, 165-168, Barcelona. 35

Blekas, K., Galatsanos, N.P., Likas, A. & Lagaris, I.E. (2005). Mixture model analysis of DNA microarray images. *IEEE Transactions on Medical Imaging*, **24**, 901-909. 35

Botros, N. & Abdul-Aziz, M. (2002). Hardware implementation of an artificial neural network using field programmable gate arrays (FPGA's). *IEEE Transactions on Industrial Electronics*, **41**, 665-667. 73

Bovik, A. (2000). *Handbook of Image and Video Processing*. Electronics & Electrical, Elsevier Academic Press. 30, 92

Bower, J.M. & Beeman, D. (1998). *The book of GENESIS: exploring realistic neural models with the GEneral ...*. TELOS, New York. 43

Bozinov, D. (2003). Autonomous system for web-based microarray image analysis. *IEEE Transactions on Nanobioscience*, **2**, 215-220. 35

Bozinov, D. & Rahnenfuhrer, J. (2002). Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering. *Bioinformatics*, **18**, 747-756. 35

Brazma, A., Robinson, A., Cameron, G. & Ashburner, M. (2000). One-stop shop for microarray data. *Nature*, **403**, 699-700. 26

Brown, P. & Botstein, D. (1999). Exploring the new world of the genome with DNA microarrays. *nature genetics*, **21**, 33-37. 19, 24

Buckley, M.J. (2000). *The Spot User's Guide*. 30

Budgen, D. (1993). *Software Design (International Computer Science Series)*. Addison Wesley.

Buhler, J., Ideker, T. & Haynor, D. (2000). Dapple: Improved Techniques for Finding Spots on DNA Microarrays. 32

Burden, R.L. & Faires, J.D. (2005). *Numerical analysis*. Cengage Learning, CA, USA. 120, 123

Burt, P. & Adelson, E. (1983). The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, **31**, 532-540. 64

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-8**, 679-698. 156

Cantor, C.R. & Smith, C.L. (1999). *Genomics: The Science and Technology Behind the Human Genome Project*. Wiley-Interscience, USA. 19

Catte, F., Lions, P.L., Morel, J.M. & Coll, T. (1992). Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. *SIAM Journal on Numerical Analysis*, **29**, 182-193. 85

Causton, H., Ren, B., Koh, S., Harbison, C., Kanin, E., Jennings, E., Lee, T., True, H., Lander, E. & Young, R. (2001). Remodeling of yeast genome expression in response to environmental changes. *Molecular biology of the cell*, **12**, 323. 19

Charbonnier, P., Blanc-Feraud, L., Aubert, G. & Barlaud, M. (1994). Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of IEEE International Conference on Image Processing*, 168-172, IEEE Comput. Soc. Press, Austin, USA. 84

Chedjou, J., Fasih, A., Grausberg, P. & Kyamakya, K. (2009). Use of CNN processors for ultra-fast solution ODE's and PDE's: A renaissance of the analog computer. In *2nd International Workshop on Nonlinear Dynamics and Synchronization, 2009. INDS '09.*, 1-1. 77

Chen, T. (2001). Adaptive impulse detection using center-weighted median filters. *IEEE Signal Processing Letters*, **8**, 1-3. 91

Cheriet, M., Said, J.N. & Suen, C.Y. (1998). A recursive thresholding technique for image segmentation. *IEEE Transactions on Image Processing*, **7**, 918-921. 34

Cheung, V., Morley, M., Aguilar, F., Massimi, A., Kucherlapati, R., Childs, G. & Others (1999). Making and reading microarrays. *Nature genetics*, **21**, 15-19. 23

Chou, J.W., Zhou, T., Kaufmann, W.K., Paules, R.S. & Bushel, P.R. (2007). Extracting gene expression patterns and identifying co-expressed genes from microarray data reveals biologically responsive processes. *BMC bioinformatics*, **8**, 427. 19

Chua, L.O. (1997). CNN: A Vision of Complexity. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, **7**, 1425-2219. 36, 77

Chua, L.O. & Roska, T. (1992b). The CNN universal machine. I. The architecture. In *Second International Workshop on Cellular Neural Networks and their Applications*, 1-10. 73, 76, 107

Chua, L.O. & Roska, T. (1993b). The CNN paradigm. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, **40**, 147-156. 3, 76, 87

Chua, L.O. & Roska, T. (2002). *Cellular neural networks and visual computing: foundation and applications*. Cambridge University Press. 154

Chua, L.O. & Yang, L. (1988b). Cellular Neural Networks: Applications. *IEEE Transactions on Circuits and Systems*, **35**, 1273-1290. 3, 76, 107, 143, 152

Chua, L.O. & Yang, L. (1988d). Cellular Neural Networks: Theory. *IEEE Transactions on Circuits and Systems*, **35**, 1257-1272. 3, 76, 107, 128, 147

Chua, L.O., Yang, L. & Krieg, K.R. (1991). Signal processing using cellular neural networks. *The Journal of VLSI Signal Processing*, **3**, 25-51. 76

Cohen, E., Riesenfeld, R.F. & Elber, G. (2001). *Geometric modeling with splines: an introduction*. A K Peters, Ltd., MA, USA. 4

Cohen, J. (2005). Computer science and bioinformatics. *Communications of the ACM*, **48**, 72-78. 11, 19

Crank, J. (1975). *The mathematics of diffusion*. larendon Press, Michigan, USA. 83

Crick, F. (1955). On degenerate templates and the adaptor hypothesis: A note for the RNA tie club. 14

Crounse, K.R. (1997). *Image processing techniques for cellular neural network hardware*. Dissertation, University of California, Berkeley. 76

Crounse, K.R. & Chua, L.O. (1995). Methods for image processing and pattern formation in Cellular Neural Networks: a tutorial. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, **42**, 583-601. 76

CRUZ, J.M. & CHUA, L.O. (1998). A 16 x 16 cellular neural network universal chip: The first complete single-chip dynamic computer array with distributed memory and with gray-scale input-output. *Analog Integrated Circuits and Signal Processing*, **15**, 227-238. 76

CSAPODI, M. & ROSKA, T. (1996). Dynamic Analogic CNN AAlgorithms for A Complex Recognition Task - A First Step Towards A Bionic Eyeglass. *International Journal of Circuit Theory and Applications*, **24**, 127-144. 76

CYBENKO, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, **2**, 303-314. 53

DASKALAKIS, A., GLOTSOS, D. & KOSTOPOULOS, S. (2009). A comparative study of individual and ensemble majority vote cDNA microarray image segmentation schemes, originating from a spot-adjustable based restoration framework. *Computer Methods and Programs in Biomedicine*, **95**, 72-88.

DEBOUCK, C. & GOODFELLOW, P. (1999). DNA microarrays in drug discovery and development. *nature genetics*, **21**, 48-50. 22

DEMIRKAYA, O., ASYALI, M.H. & SHOUKRI, M.M. (2005). Segmentation of cDNA Microarray Spots Using Markov Random Field Modeling. *Bioinformatics*, **21**, 2994-3000. 35

DEMUTH, H., BEALE, M. & HAGAN, M. (2007). *Neural Network Toolbox5: User's Guide*. 64

DERISI, J.L., IYER, V.R. & BROWN, P.O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680-686. 18

DIAMANDIS, E.P. (2000). Sequencing with Microarray Technology-A Powerful New Tool For Molecular Diagnostics. *Clinical Chemistry*, **46**, 1523-1525. 22

DOLAN, R. & DESOUZA, G. (2009). GPU-based simulation of cellular neural networks for image processing. In *Proc. International Joint Conference on Neural Networks IJCNN 2009*, 730-735. 87, 107

DOMINGUEZ-CASTRO, R., ESPEJO, S., RODRIGUEZ-VAZQUEZ, A. & CARMONA, R. (1994). A CNN universal chip in CMOS technology. In *The Third IEEE International Workshop on Cellular Neural Networks and their Applications*, 91-96. 76

DOMINGUEZ-CASTRO, R., ESPEJO, S., RODRIGUEZ-VAZQUEZ, A., CARMONA, R.A., FOLDESY, P., ZARANDY, A., SZOLGAY, P., SZIRANYI, T. & ROSKA, T. (1997). A 0.8 micrometer CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. *IEEE Journal of Solid-State Circuits*, **32**, 1013-1026. 76

DRMANAC, S., KITA, D., LABAT, I., HAUSER, B., SCHMIDT, C., BURCZAK, J.D. & DRMANAC, R. (1998). Accurate sequencing by hybridization for DNA diagnostics and individual genomics. *Nature biotechnology*, **16**, 54-8. 22

DUDA, R.O. & HART, P.E. (1973). *Pattern classification and scene analysis*. Wiley-Interscience, New York, USA. 48

Dumontier, C., Luthon, F. & Charras, J.P. (1999). Real-time DSP implementation for MRF-based video motion detection. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, **8**, 1341-7. 77

Dunham, C.B. (1973). The limit of non-linear Chebyshev approximation on subsets. *Aequationes Mathematicae*, **9**, 60-63. 119

Dunham, C.B. (1974). Linear Chebyshev approximation. *Aequationes Mathematicae*, **10**, 40-45. 119

Dunn, J. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Cybernetics and Systems*, **3**, 32-57. 35

Eisen, M.B. (2010). ScanAlayse. Online. 32, 142

Eisen, M.B. & Brown, P.O. (1999). DNA arrays for analysis of gene expression. *Methods in Enzymology*, **303**, 179-205. 5, 19

Eisen, M.B., Spellman, P.T., Brown, P.O. & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, **95**, 14863-14868. 19

Epstein, J., Biran, I. & Walt, D. (2002). Fluorescence-based nucleic acid detection and microarrays. *Analytica Chimica Acta*, **469**, 3-36. 24

Erler, K. & Jernigan, E. (1994). Adaptive image restoration using recursive image filters. *IEEE Transactions on Signal Processing*, **42**, 1877-1881. 82

Evans, W.E. (1999). Pharmacogenomics: Translating Functional Genomics into Rational Therapeutics. *Science*, **286**, 487-491. 17

Fajfar, I., Bratkovič, F., Tuma, T. & Puhan, J. (1998). A rigorous design method for binary cellular neural networks. *International Journal of Circuit Theory and Applications*, **26**, 365-373. 131

Fasih, A., Chedjou, J.C. & Kyamakya, K. (2008). Cellular Neural Network Trainer and Template Optimization for Advanced Robot Locomotion, Based on Genetic Algorithm. In *Proc. 15th International Conference on Mechatronics and Machine Vision in Practice M2VIP 2008*, 317-322, IEEE, Auckland. 77

Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall, US. 50, 64, 66

Feng, Q., Yu, S. & Wang, H. (2006). An New Automatic Nucleated Cell Counting Method With Improved Cellular Neural Networks (ICNN). In *Proc. 10th International Workshop on Cellular Neural Networks and Their Applications CNNA '06*, 1-4, IEEE, Istanbul, Turkey. 77

Fernández-Muñoz, J., Preciado-D\'\iaz, V. & Jaramillo-Morán, M. (2006). *Nonlinear Mappings with Cellular Neural Networks*, vol. 4177, 350-359. Springer-Verlag, Berlin, Germany. 77, 119, 120, 123, 124

Fischl, B. & Schwartz, E. (1997). Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 342-352. 83

FISHER, R.A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, **7**, 179-188. 51

FLEISS, J.L. (1986). *The Design and Analysis of Clinical Experiments*. John Wiley & Sons, 1st edn. 98

FORD, G., ESTES, R. & CHEN, H. (1992). Space scale analysis for image sampling and interpolation. In *Proceedings ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 165-168, IEEE. 82

FRANKLIN, R.E. & GOSLING, R.G. (1953). The structure of sodium thymonucleate fibres. I. The influence of water content. *Acta Crystallographica*, **6**, 673-677. 14

FRASER, K. (2006). *cDNA Microarray Image Analysis - A Fully Automatic framework*. Ph.D. thesis, Brunel University. 4, 5, 113, 115

FRASER, K., O'NEILL, P., WANG, Z. & LIU, X. (2004). Copasetic analysis: a framework for the blind analysis of microarray imagery. *IEE Proceedings Systems Biology*, **1**, 190-196. 35

FRASER, K., WANG, Z., LI, Y., KELLAM, P. & LIU, X. (2007). Noise Filtering and Microarray Image Reconstruction Via Chained Fouriers. In *Proceedings of the 7th international conference on Intelligent data analysis*, 308-319. 38

FRASER, K., WANG, Z., LI, Y., KELLAM, P. & LIU, X. (2008). Can graph-cutting improve microarray gene expression reconstructions? *Pattern Recognition Letters*, **29**, 2129-2136. 38

FRASER, K., WANG, Z. & LIU, X. (2010). *Microarray Image Analysis*. Taylor and Francis. 34, 62, 69, 71, 107, 108, 109, 113, 119

FROEHLICH, J. & WEICKERT, J. (1994). Image processing using a wavelet algorithm for nonlinear diffusion. 85

FUNAHASHI, K. (1989). On The Approximate Realization Of Continuous Mappings By Neural Networks. *Neural Networks*, **2**, 183-192. 53

GADEA, R., CERDA, J., BALLESTER, F. & MACHOLI, A. (2000). Artificial neural network implementation on a single FPGA of a pipelined on-line backpropagation. In *13th International Symposium on System Synthesis (ISSS'00)*, 225-230, The IEEE Computer Society, Madrid, Spain. 73

GENE, M. (1999). Whole-Genome DNA Sequencing. *Computing in Science and Eng.*, **1**, 33-43. 11

GERHOLD, D. & CASKEY, C.T. (1996). It's the genes! EST access to human genome content. *BioEssays*, **18**, 973-981. 10

GERIG, G., KUBLER, O., KIKINIS, R. & JOLESZ, F.A. (1992). Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, **11**, 221-232. 82, 83

GERSHON, D. (2002). Microarray technology: an array of opportunities. *Nature*, **416**, 885-91. 24

Gilboa, G., Zeevi, Y. & Sochen, N. (2001). Complex Difusion Processes for Image Filtering. In *Scale-Space and Morphology in Computer Vision*, 299-307, Springer, Berlin / Heidelberg. 86, 92

Gilboa, G., Sochen, N. & Zeevi, Y.Y. (2004). Image enhancement and denoising by complex diffusion processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1020-1036. 86

Gillespie, D. & Spiegelman, S. (1965). A quantitative assay for DNA-RNA hybrids with DNA immobilized on a membrane. *Journal of Molecular Biology*, **12**, 829-842. 18

Glasbey, C.A. (2001). Image Analysis Of A Genotyping Microarray Experiment. 30

Glökler, J. & Angenendt, P. (2003). Protein and antibody microarray technology. *Journal of Chromatography B*, **797**, 229-240. 22

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M. & Others (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, **286**, 531. 19

González, R.C. & Woods, R.E. (2008). *Digital image processing*. Prentice Hall, Upper Saddle River, New Jersey. 82, 110

Gray, N., Wodicka, L., Thunnissen, A., Norman, T., Kwon, S., Espinoza, F., Morgan, D., Barnes, G., LeClerc, S., Meijer, L. & Others (1998). Exploiting chemical libraries, structure, and genomics in the search for kinase inhibitors. *Science*, **281**, 533. 19

Guillermo Sapiro (2001). *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge, UK. 84

Guo, Z., Guilfoyle, R., Thiel, A. & Wang, R. (1994). Direct fluorescence analysis of genetic polymorphisms by hybridization with oligonucleotide arrays on glass supports. *Nucleic Acids Research*, **22**, 5456-5465. 22, 23, 24

Gygi, S.P., Rochon, Y., Franza, B.R. & Aebersold, R. (1999). Correlation between Protein and mRNA Abundance in Yeast. *Molecule Cellular Biology*, **19**, 1720-1730. 12, 16

Ham, F. & Kostanic, I. (2000). *Principles of Neurocomputing for Science and Engineering*. McGraw-Hill. 64

Ham, F.M. (1994). Detection and Classification of Biological Substances Using Infrared Absorption Spectroscopy and Hybrid Artiffcial Network. *Journal of Atrficial Neural Networks*, **1**, 100-104. 54

Hartman, E.J., Keeler, J.D. & Kowalski, J.M. (1990). Layered Neural Networks with Gaussian Hidden Units as Universal Approximations. *Neural Computation*, **2**, 210-215. 53

Haung, C.H., Leow, W.K. & Racoceanu, D. (2009). A Cellular Neural Network as a Principal Component Analyzer. In *Proc. International Joint Conference on Neural Networks IJCNN 2009*, 1163-1170, IEEE, Atlanta, GA, USA. 77

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice-Hall International, London, UK, 2nd edn. 42, 54, 64, 72

Haykin, S. (2001). *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, New Jersey, 4th edn. 72, 82

Hebb, D.O. (2002). *The Organization of Behavior: A Neuropsychological Theory*. Lawrence Erlbaum Associates Inc, New York,, new editio edn. 51, 62

Hegde, P., Qi, R., Abernathy, K., Gay, C., Dharap, S., Gaspard, R., Hughes, J.E., Snesrud, E., Lee, N. & Quackenbush, J. (2000). A concise guide to cDNA microarray analysis. *BioTechniques*, **29**, 548-556. 11

Hirata, R., Barrera, J., Hashimoto, R.F., Dantas, D.O. & Esteves, G.H. (2002). Segmentation of Microarray Images by Mathematical Morphology. *Real-Time Imaging*, **8**, 491-505. 4

Ho, J. & Hwang, W.L. (2007). Segmenting Microarray Image Spots using an Active Contour Approach. In W.L. Hwang, ed., *IEEE International Conference on Image Processing*, vol. 6, 273-276. 35

Ho, T.Y., Lam, P.M. & Leung, C.S. (2008). Parallelization of cellular neural networks on GPU. *Pattern Recognition*, **41**, 2684-2692. 87, 107

Holloway, A.J., van Laar, R.K., Tothill, R.W. & Bowtell, D.D.L. (2002). Options available ,from start to finish, for obtaining data from DNA microarrays II. *Nature genetics supplement*, **32**, 481-489. 18

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, **79**, 2554-2558. 50

Hornik, K., Stinchcombe, M. & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, **2**, 359-366. 53

Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H., He, Y. & Others (2000). Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109-126. 19

Hughes, T.R., Mao, M., Jones, A.R., Burchard, J., Marton, M.J., Shannon, K.W. & al., E. (2001). Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. *Nat Biotech*, **19**, 342-347. 18

Iijima, T. (1962). Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrical Laboratory*, **26**, 368-388. 83

Inoue, T. & Nishio, Y. (2009). Applications of color image processing using three-layer cellular neural network considering HSB model. In *Proc. International Joint Conference on Neural Networks IJCNN 2009*, 1335-1342, IEEE, Atlanta, GA, USA. 77

Jain, A.N., Tokuyasu, T.A., Snijders, A.M., Segraves, R., Albertson, D.G. & Pinkel, D. (2002). Fully Automatic Quantification of Microarray Image Data. *Genome Res.*, **12**, 325-332. 5, 32, 108

Katzer, M., Kummert, F. & Sagerer, G. (2003). Methods for automatic microarray image segmentation. *IEEE Transactions on Nanobioscience*, **2**, 202-214. 5, 32, 35, 108

Kaveh, M. (1996). Anisotropic blind image restoration. In *Proceedings of 3rd IEEE International Conference on Image Processing*, 461-464, IEEE. 82

Kék, L., Karacs, K. & Roska, T. (2007). *cellular wave computing library*. Cellular Sensory Wave Computers Laboratory, Ungarian Academy Of Sciences, Budapest , Hungary, 2nd edn. 128

Kim, H. (2006). Gradient Histogram-Based Anisotropic Diffusion. *Personal Communication*. 85

Kim, S.S. & Jung, S. (2004). Hardware implementation of a real time neural network controller with a DSP and an FPGA. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04.*, vol. 5, 4639-4644, IEEE. 73

Kimia, B. & Siddiqi, K. (1994). Geometric heat equation and nonlinear diffusion of shapes and images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 113-120, IEEE Comput. Soc. Press. 83

King, M. & Glick, S. (1993). Local geometry variable conductance diffusion for post-reconstruction filtering. In *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*, 1667-1671, IEEE. 82

Kleppe, K. (1971). Studies on polynucleotides: XCVI. Repair replication of short synthetic DNA's as catalyzed by DNA polymerases. *Journal of Molecular Biology*, **56**, 341-361. 10

Klug, W.S. & Cummings, M.R. (2003). *Concepts of genetics*. Prentice Hall. 12

Ko, S.J. & Lee, Y. (1991). Center weighted median filters and their applications to image enhancement. *IEEE Transactions on Circuits and Systems*, **38**, 984-993. 91

Kohonen, T. (1977). *Associative memory: a system-theoretical approach*. Springer, New York, USA. 50

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, **43**, 59-69. 58

Kohonen, T. (1989). *Self-organization and associative memory*. Springer-Verlag, Inc., New York, 3rd edn. 58

Korn, E.L., Habermann, J.K., Upender, M.B., Ried, T. & McShane, L.M. (2004). Objective method of comparing DNA microarray image analysis systems. *Biotechniques*, **36**, 960-967. 99, 136

Kozek, T. & Roska, T. (1996). A Double Time-Scale CNN For Solving Two-Dimensional Navier - Stokes Equation. *International Journal of Circuit Theory and Applications*, **24**, 49-55. 77, 153, 160

Kozek, T., Chua, L.O., Roska, T., Wolf, D., Tetzlaff, R., Puffer, F. & Lotz, K. (1995). Simulating nonlinear waves and partial differential equations via CNN II. Typical examples. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **42**, 816-820. 77

Krieg, K., Chua, L. & Yang, L. (1990). Analog signal processing using cellular neural networks. In *IEEE International Symposium on Circuits and Systems*, 958-961, IEEE, New Orleans, LA , USA. 76

Lamberti, C., Sitta, M. & Sgallari, F. (2002). Improvements to the Anisotropic Diffusion Model for 2-D Echo ImageProcessing. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 5, 1872-1873, IEEE. 82

Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J. & al., E. (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**, 860-921. 11

Larese, M. & Gómez, J. (2009). Quantitative Improvements in cDNA Microarray Spot Segmentation. In *Proceedings of the 4th Brazilian Symposium on Bioinformatics: Advances in Bioinformatics and Computational Biology*, vol. 5676, 60-72.

Lashkari, D.A., DeRisi, J.L., McCusker, J.H., Namath, A.F., Gentile, C., Hwang, S.Y., Brown, P.O. & Davis, R.W. (1997). Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proceedings of the National Academy of Sciences of the United States of America*, **94**, 13057-13062. 19

Lawrence, N.D., Milo, M., Niranjan, M., Rashbass, P. & Soullier, S. (2003). Bayesian processing of microarray images. *Neural Networks for Signal Processing*, 71-80. 35

Ledford, H. (2007). All about Craig: the first 'full' genome sequence. *Nature*, **449**, 6-7. 11

Lehmussola, A., Ruusuvuori, P. & Yli-Harja, O. (2006). Evaluating the performance of microarray segmentation algorithms. *Bioinformatics*, **22**, 2910-2917. 5, 28, 32

Lehtokangas, M., Korpisaari, P. & Kaski, K. (1996). Maximum Covariance Method for Weight Initialisation of Multilayer Perceptron Networks. In *European Symposium on Artificial Neural Networks*. 72

Lemieux, B., Aharoni, A. & Schena, M. (1998). Overview of DNA chip technology. *Molecular Breeding*, **4**, 277-289. 23

Leung, Y. & Cavalieri, D. (2003). Fundamentals of cDNA microarray data analysis. *TRENDS in Genetics*, **19**, 649-659. 18, 22

Levene, P.A. (1919). The Structure of Yeast Nucleic Acid. *The Journal Of Biological Chemistry*, **40**, 415-424. 13

Ley, T.J., Mardis, E.R., Ding, L., Fulton, B., McLellan, M.D., Chen, K. & al., E. (2008). DNA sequencing of a cytogenetically normal acute myeloid leukaemia genome. *Nature*, **456**, 66-72. 11

Li, S. (1994). Markov random field models in computer vision. In *Computer Vision*, vol. 801, 361370, Springer, Berlin / Heidelberg. 35

Li, Y., Li, T., Liu, S., Qiu, M., Han, Z., Jiang, Z. & Li, R. (2004). Systematic comparison of the fidelity of aRNA, mRNA and T-RNA on gene expression profiling using cDNA microarray. *Journal of Biotechnology*, **107**, 19-28. 22

Liao, P.S., Chen, T.S. & Chung, P.C. (2001). A Fast Algorithm for Multilevel Thresholding. *Journal of Information Science and Engineering*, **17**, 713-727.

Linan, G., Rodriguez-Vazquez, A., Espejo, S. & Dominguez-Castro, R. (2003). ACE16k: a 128x128 focal plane analog processor with digital I/O. *International Journal of Neural Systems*, **13**, 427-434. 76

Lindeberg, T. (1993). *Scale-space theory in computer vision*. Kluwer Academic Publishers, Dordrecht, The Netherlands. 83

Lipshutz, R.J., Fodor, S.P.A., Gingeras, T.R. & Lockhart, D.J. (1999). High density synthetic oligonucleotide arrays. *Nature Genetics*, **21**, 20-24. 18

Loew, M., Rosenman, J. & Chen, J. (1994). Clinical tool for enhancement of portal images. *Medical Imaging 1994: Image Processing*, **2167**, 543-550. 82

Lonardi, S. & Yu, L. (2004). Gridding and Compression of Microarray Images. In *IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004*, 122 - 130, {IEEE} Computer Society. 31

Lorenz, M.G. & Wackernagel, W. (1994). Bacterial gene transfer by natural genetic transformation in the environment. *Microbiological reviews*, **58**, 563-602. 13

Luchini, P. (1984). Two-Dimensional Numerical Integration Using A Square Mesh. *Computer Physics Communications*, **31**, 303-310. 152

Lueking, A., Horn, M., Eickhoff, H., Bussow, K., Lehrach, H. & Walter, G. (1999). Protein Microarrays For Gene Expression And Antibody Screening. *Analytical Biochemistry*, **270**, 103-111. 22

Lukac, R., Plataniotis, K.N., Smolka, B. & Venetsanopoulos, A.N. (2004). A Multichannel Order-Statistic Technique For Cdna Microarray Image Processing. *IEEE Transactions on Nanobioscience*, **3**, 272-285. 5, 32, 36

Lukac, R., Plataniotis, K.N., Smolka, B. & Venetsanopoulos, A.N. (2005). Cdna Microarray Image Processing Using Fuzzy Vector Filtering Framework. *Fuzzy Sets and Systems*, **152**, 17-35. 5

Lumonics, G. (1999). Quantarray Analysis Software. *Operator's manual*. 32

MacQueen, J.B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In L.M.L. Cam & J. Neyman, eds., *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297, University of California Press. 35, 96

Maeda, J., Iizawa, T., Ishizaka, T., Ishikawa, C. & Suzuki, Y. (1998). Segmentation Of Natural Images Using Anisotropic Diffusion And Linking Of Boundary Edges. *Pattern Recognition*, **31**, 1993-1999. 83

Mahner, M. & Kary, M. (1997). What Exactly Are Genomes, Genotypes And Phenotypes? And What About Phenomes? *Journal of Theoretical Biology*, **186**, 55-63.

Marr, D. & Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, **207**, 187-217. 34

Marra, M.A., Hillier, L. & Waterston, R.H. (1998). Expressed sequence tags-ESTablishing bridges between genomes. *Trends in Genetics*, **14**, 4-7. 10

Marton, M., DeRisi, J., Bennett, H., Iyer, V., Meyer, M., Roberts, C., Stoughton, R., Burchard, J., Slade, D., Dai, H. & Others (1998). Drug target validation and identification of secondary drug target effects using DNA microarrays. *Nature medicine*, **4**, 1293-1301. 19

MathWorks (2007). *Image Processing Toolbox5: User's Guide*. 67

Matsumoto, T., Chua, L. & Yokohama, T. (1990a). Image thinning with a cellular neural network. *IEEE Transactions on Circuits and Systems*, **37**, 638-640. 77

Matsumoto, T., Chua, L.O. & Suzuki, H. (1990b). CNN cloning template: connected component detector. *IEEE Transactions on Circuits and Systems*, **37**, 633-635. 95

McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, **5**, 115-133. 42, 46, 51, 62

MCDONNELL, M. (1981). Box-filtering techniques. *Computer Graphics and Image Processing*, **17**, 65-70. 82

McGraw, K.O. & Wong, S.P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological methods*, **1**, 30-46. 98, 100

McLachlan, G.J., Do, K.A. & Ambroise, C. (2004). *Analyzing Microarray Gene Expression Data*. John Wiley & Sons, Inc., Hoboken, NJ, USA. 10

Mehnert, A. & Jackway, P. (1997). An improved seeded region growing algorithm. *Pattern Recognition Letters*, **18**, 1065-1071. 35

Meselson, M. & Stahl, F.W. (1958). The Replication Of Dna In Escherichia Coli. *Proceedings of the National Academy of Sciences of the United States of America*, **44**, 671-82. 14

Minsky, M.L. & Papert, S. (1972). *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, USA. 52

Mitra, S.K. & Sicuranza, G.L. (2000). *Nonlinear Image Processing*. Elsevier Science & Technology, Oxford, UK. 81

Moore, S., Payton, P. & Giovannoni, J. (2002). DNA Micro-arrays for Gene Expression Analysis. *Molecular plant biology*, **2**, 65-76. 23

Moore, S.K. (2001). Making chips to probe genes. *IEEE Spectrum*, **38**, 54-60. 11, 12

Morris, D. (2008). Blind Microarray Gridding: A New Framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **38**, 33-41. 62, 88, 108

Nagasawa, M. (1993). *Schrödinger equations and diffusion theory*. Birkhauser Verlag AG, Basel. 86

NAMBA, M. (2008). Estimating learner's comprehension with Cellular Neural Network for associative memory. In *Cellular Neural Networks and Their Applications, 2008. CNNA 2008. 11th International Workshop on*, 150-153, IEEE, Santiago de Compostela. 77

NITZBERG, M. & SHIOTA, T. (1992). Nonlinear image smoothing with edge and corner enhancement. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **14**. 85

NODA, H., SHIRAZI, M.N. & KAWAGUCHI, E. (2002). MRF-based texture segmentation using wavelet decomposed images. *Pattern Recognition*, **35**, 771-782. 35

NUWAYSIR, E., HUANG, W., ALBERT, T. & SINGH, J. (2002). Gene expression analysis using oligonucleotide arrays produced by maskless photolithography. *Genome Research*, **12**, 1749-1755. 22, 24

OLIVEIRA, M.M., MCKENNA, R., BOWEN, B. & CHANG, Y.S. (2001). Fast Digital Image Inpainting. In *Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001),*, 261-266, Marbella, Spain. 147

OMONDI, A.R. & RAJAPAKSE, J.C. (2006). *FPGA implementations of neural networks*. Springer Verlag, Dordrecht, The Netherlands. 73

O'NEILL, P., MAGOULAS, G. & LIU, X. (2003). Improved processing of microarray data using image reconstruction techniques. *IEEE Transactions On Nanobioscience*, **2**, 176-183. 38, 146

ORENGO, C.A., JONES, D.T. & THORNTON, J.M. (2003). *Bioinformatics: Genes, Proteins and Computers*. BIOS Scientific Publishers, Oxford, UK. 12

OTSU, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, **11**, 285-296. 63

OTSU, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, **9**, 62-66. 34

OZKAYA, B., DEMIR, A. & BILGILI, M. (2007). Neural network prediction model for the methane fraction in biogas from field-scale landfill bioreactors. *Environmental Modelling & Software*, **22**, 815-822. 43

PARK, H. & NISHIMURA, T. (2007). Reduced speckle noise on medical ultrasound images using cellular neural network. In *Annual International Conference of the IEEE: Engineering in Medicine and Biology Society*, vol. 2007, 2138-41, IEEE, Lyon. 77

PARKER, D.B. (1985). Learning-logic. 52

PARVIN, B.A. & BHANU, B. (1983). Segmentation of images using a relaxation technique. *Proceedings CVPR'83*, 151. 34

PEARLMUTTER, B. (1990). Dynamic recurrent neural networks. 50

PEASE, A.C., SOLAS, D., SULLIVAN, E.J., CRONIN, M.T., HOLMES, C.P. & FODOR, S.P. (1994). Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proceedings of the National Academy of Sciences of the United States of America*, **91**, 5022-5028. 22

PERKINS, W. (1980). Area segmentation of images using edge points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 8-15. 34

PERONA, P. & MALIK, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629-639. 30, 34, 82, 84, 85, 92, 111, 156

PEROU, C., JEFFREY, S., VAN DE RIJN, M., REES, C., EISEN, M., ROSS, D., PERGAMENSCHIKOV, A., WILLIAMS, C., ZHU, S., LEE, J. & OTHERS (1999). Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proceedings of the National Academy of Sciences of the United States of America*, **96**, 9212. 19

PITAS, I. (1993). *Digital Image Processing Algorithms*. Prentice-Hall, Harlow, UK. 81

PITAS, I. (2000). *Digital image processing algorithms and applications*. Wiley & Sons, USA. 81

PITAS, I. & VENETSANOPOULOS, A.N. (1990). *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands. 81

POLLARA, G., KWAN, A., NEWTON, P.J., HANDLEY, M.E., CHAIN, B.M. & KATZ, D.R. (2005). Dendritic cells in viral pathogenesis: protective or defective? *International journal of experimental pathology*, **86**, 187-204.

POLYCARPOU, M.M. & IOANNOU, P.A. (1992). Modelling, Identification and Stable Adaptive Control of Continuous-Time Nonlinear Dynamical Systems Using Neural Networks. In *American Control Conference, 1992*, 36-40, Chicago, IL, USA. 43

PON, R.T. & YU, S. (2005). Tandem oligonucleotide synthesis using linker phosphoramidites. *Nucleic acids research*, **33**, 1940-1948. 22

PRECIADO, V. (2002). Piecewise-Linear Approximation of Any Smooth Output Function on the Cellular Neural Network. In *Artificial Neural Networks-ICANN 2002*, 462-467, Springer. 120

QUACKENBUSH, J. (2001). Computational analysis of microarray data. *Nature Reviews Genetics*, **2**, 418-427. 10

RABINOW, P. (1997). *Making PCR: A Story of Biotechnology*. University Of Chicago Press. 11

RANADE, S. & ROSENFELD, A. (1980). Point pattern matching by relaxation. *Pattern recognition*, **12**, 269-275. 34

REKECZKY, C. (2002). CNN architectures for constrained diffusion based locally adaptive image processing. *International Journal of Circuit Theory and Applications*, **30**, 313-348. 113, 129, 130

REKECZKY, C. & CHUA, L.O. (1999). Computing with Front Propagation: Active Contour And Skeleton Models In Continuous-Time CNN. *The Journal of VLSI Signal Processing*, **23**, 373-402. 76

REKECZKY, C., USHIDA, A. & ROSKA, T. (1995). Rotation Invariant Detection of Moving and Standing Objects Using Analogic Cellular Neural Network Algorithms Based on Ring-Codes. *IEICE Transactions on Fundamentals and Information Sciences*, **78**, 1316-1330. 76

Rekeczky, C., Roska, T. & Ushida, A. (1998b). CNN-based difference-controlled adaptive non-linear image filters. *International Journal of Circuit Theory and Applications*, **26**, 375-423. 111, 128, 156

Rekeczky, C., Tahy, A., Vegh, Z. & Roska, T. (1999). CNN-based spatio-temporal nonlinear filtering and endocardial boundary detection in echocardiography. *International Journal of Circuit Theory and Applications*, **27**, 171-207. 76

Ripley, B.D. (2008). *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK. 43

Roberts, R.B., Abelson, P.H., Cowi, D.B., Bolton, E.B. & Britten, J.R. (1955). *Studies of biosynthesis in Escherichia coli*. Carnegie Institution of Washington, Washington, DC. 10

Rodriguez-Fernandez, D., Vilarino, D.L. & Pardo, X.M. (2008). CNN implementation of a moving object segmentation approach for real-time video surveillance. In *Cellular Neural Networks and Their Applications, 2008. CNNA 2008. 11th International Workshop on*, 129-134, IEEE, Santiago de Compostela. 77

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386-408. 51

Rosenblatt, F. (1962). *Principles of neurodynamics: perceptions and the theory of brain mechanisms*. Cornell Aeronautical Laboratory, Buffalo, USA. 51

Roska, T. & Chua, L.O. (1990). Cellular neural networks with nonlinear and delay-type template elements. In *Cellular Neural Networks and their Applications, 1990. CNNA-90 Proceedings., 1990 IEEE International Workshop on*, 12-25. 77

Roska, T. & Chua, L.O. (1993). The CNN universal machine: an analogic array computer. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, **40**, 163-173. 161

Roska, T., Boros, T., Thiran, P. & Chua, L. (1990). Detecting simple motion using cellular neural networks. In *IEEE International Workshop on Cellular Neural Networks and their Applications*, 127-138, IEEE, Budapest , Hungary. 77

Roska, T., Boros, T., Radványi, A., Thiran, P. & Chua, L.O. (1992). Detecting moving and standing objects using cellular neural networks. *International Journal of Circuit Theory and Applications*, **20**, 613-628. 77

Roska, T., Chua, L.O., Wolf, D., Kozek, T., Tetzlaff, R. & Puffer, F. (1995). Simulating nonlinear waves and partial differential equations via CNN I. Basic techniques. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **42**, 807-815. 77

Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S.S., de Rijn, M.V. & Waltham, M. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, **24**, 227-234. 19

Rudin, L., Osher, S. & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, **60**, 259-268. 84

RUEDA, L. & ROJAS, J. (2009). A Pattern Classification Approach to DNA Microarray Image Segmentation. *Pattern Recognition in Bioinformatics*.

RUMELHART, D.E., HINTON, G.E. & WILLIAMS, R.J. (1986). *Learning internal representation by error propagation, Parallel Distributed*, 318-362. MIT Press, Cambridge, Mass. 42, 52

SAINT-MARC, P., CHEN, J. & MEDIONI, G. (1989). Adaptive smoothing: a general tool for early vision. In *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 618-624, IEEE Comput. Soc. Press. 82

SAINT-MARC, P., CHEN, J.S. & MEDIONI, G. (1991). Adaptive smoothing: a general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 514-529. 82

SAMARTZIDOU, H., TURNER, L., HOUTS, T., FRORNE, M., WORLEY, J. & ALBERTSEN, H. (2001). Lucidea Microarray ScoreCard: An integrated analysis tool for microarray experiments. *Life Science News*, **7**, 1-10. 18, 39, 133, 156

SAMAVI, S., SHIRANI, S., KARIMI, N. & DEEN, M. (2004). A Pipeline Architecture for Processing of DNA Microarrays Images. *The Journal of VLSI Signal Processing*, **38**, 287-297. 4, 143

SCHARR, H., BLACK, M.J. & HAUSSECKER, H.W. (2003). Image statistics and anisotropic diffusion. In *IEEE International Conference on Computer Vision 2003*, vol. 2, 840-847. 86

SCHENA, M. (2000). *Microarray Biochip Technology*. Eaton Publishing Company. 21

SCHENA, M., SHALON, D., DAVIS, R.W. & BROWN, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467-470. 11, 12, 18, 28

SCHENA, M., HELLER, R., THENAULT, T., KONRAD, K., LACHENMEIER, E. & DAVIS, R. (1998). Microarrays: biotechnology's discovery platform for functional genomics. *Trends in Biotechnology*, **16**, 301-306. 18

SCHERF, U., ROSS, D., WALTHAM, M., SMITH, L., LEE, J., TANABE, L., KOHN, K., REINHOLD, W., MYERS, T., ANDREWS, D. & OTHERS (2000). A gene expression database for the molecular pharmacology of cancer. *nature genetics*, **24**, 236-244. 19

SCHERMER, M.J. (1999). *Confocal scanning in microscopy in microarray detection*. Oxford University Press, New York. 27

SEZGIN, M. & SANKUR, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, **13**, 146-168. 94, 111

SHAH, J. (1996). A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 136-142, IEEE Comput. Soc. Press. 83

SHALON, D., SMITH, S.J. & BROWN, P.O. (1996). A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, **6**, 639-645. 11

SHENGHUA, N., WANG, P., PAUN, M. & WEIZHONG, D. (2009). Spotted cDNA microarray image segmentation using ACWE. *Science And Technology*, **12**, 249263.

SHERLOCK, G. & HERNANDEZ-BOUSSARD, T. (2001). The Stanford Microarray Database. *Nucleic Acids Research*, **29**, 152-155. 26

SHI, B.E., ROSKA, T. & CHUA, L.O. (1993). Design of linear cellular neural networks for motion sensitive filtering. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, **40**, 320-331. 77

SHIPP, M., ROSS, K., TAMAYO, P., WENG, A., KUTOK, J., AGUIAR, R., GAASENBEEK, M., ANGELO, M., REICH, M., PINKUS, G. & OTHERS (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, **8**, 68-74. 19

SHITONG, W. & MIN, W. (2006). A New Detection Algorithm (NDA) Based on Fuzzy Cellular Neural Networks for White Blood Cell Detection. *IEEE Transactions on Information Technology in Biomedicine*, **10**, 5-10. 77

SIDDIQUI, K., HERO, A. & SIDDIQUI, M. (2002). Mathematical morphology applied to spot segmentation and quantification of gene microarray images. In *Asilomar Conference on Signals and Systems*, Pacific Grove, CA. 4, 34

SIJBERS, J., SCHEUNDERS, P., VERHOYE, M., LINDEN, A.V.D., DYCK, D.V. & RAMAN, E. (1997). Watershed-based segmentation of 3D MR data for volume quantization. *Magnetic resonance imaging*, **15**, 679-688. 82

SINGH-GASSON, S., GREEN, R.D., YUE, Y., NELSON, C., BLATTNER, F., SUSSMAN, M.R. & CERRINA, F. (1999). Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array. *Nat Biotech*, **17**, 974-978. 24

SMYTH, G.K., YANG, Y.H. & SPEED, T. (2003). *Statistical Issues in cDNA Microarray Data Analysis*, vol. 224, chap. 9, 111-136. 37

SOILLE, P. & VINCENT, L. (1990). Determining watersheds in digital pictures via flooding simulations. In *Visual Communications and Image Processing*, vol. 1360, 1240-250, Bellingham, USA. 4

SOOS, B., RAK, A., VERES, J. & CSEREY, G. (2008). GPU powered CNN simulator (SIMCNN) with graphical flow based programmability. In *11th International Workshop on Cellular Neural Networks and Their Applications CNNA2008.*, 163-168. 87, 107

SOUTHERN, E. & MIR, K. (1999). Molecular interactions on microarrays. *nature genetics*, **21**, 5-9. 24

SOUTHERN, E.M. (1975). Detection of specific sequences among DNA fragments separated by gel electrophoresis. *Journal of Molecular Biology*, **98**, 503-517. 18

SPRANKLE, M. & HUBBARD, J. (2008). *Problem Solving and Programming Concepts: International Version*. Pearson Education.

SRIDHAR, B., PHATAK, A. & CHATTERJI, G.B. (1993). Scene segmentation of natural images using texture features and back-propagation. 72

SRINARK, T. & KAMBHAMETTU, C. (2004). A microarray image analysis system based on multiple snakes. *Journal of Biological systems*, **12**, 127-157. 35

STAUNTON, J., SLONIM, D., COLLER, H., TAMAYO, P., ANGELO, M., PARK, J., SCHERF, U., LEE, J., REINHOLD, W., WEINSTEIN, J. & OTHERS (2001). Chemosensitivity prediction by transcriptional profiling. *Proceedings of the National Academy of Sciences*, **98**, 10787. 19

STEEN, E.N. & OLSTAD, B. (1994). *Scale-space and boundary detection in ultrasonic imaging using nonlinear signal-adaptive anisotropic diffusion*, vol. 2167, 116127. 82

STEKEL, D. (2003). *Microarray bioinformatics*. Cambridge University Press. 11, 31

STILLER, E. & LEBLANC, C. (2002). *Project-based software engineering: an object-oriented approach*. Addison Wesley.

STOER, J. & BULIRSCH, R. (2002). *Introduction to Numerical Analysis*. Springer. 152

SU, T.J. & JHANG, J.W. (2006). Medical Image Noise Reduction Using Cellular Neural Networks. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 228-231, IEEE, Pasadena, CA, USA. 77

SUTTON, W. (1903). The chromosomes in heredity. *The Biological Bulletin*, **4**, 231. 12

SZATMARI, I., SCHULTZ, A., REKECZKY, C., KOZEK, T., ROSKA, T. & CHUA, L.O. (2000). Morphology and autowave metric on CNN applied to bubble-debris classification. *IEEE Transactions on Neural Networks*, **11**, 1385-1393. 85

SZIRANYI, T. & CSAPODI, M. (1998). Texture classification and segmentation by cellular neural networks using genetic learning. *Computer vision and image understanding*. 77

TANAKA, M., CROUNSE, K.R. & ROSKA, T. (1992). Template synthesis of cellular neural networks for information coding and decoding. In *Cellular Neural Networks and their Applications, 1992. CNNA-92 Proceedings., Second International Workshop on*, 29-35. 76

TER HAAR ROMENY, B.M. (1994). *Geometry-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, Dordrecht, The Netherlands. 84

TOKES, S., SZABO, V., ORZO, L., DIVOS, P. & KRIVOSIJA, Z. (2008). Digital holographic microscopy and CNN based image processing for biohazard detection. In *Cellular Neural Networks and Their Applications, 2008. CNNA 2008. 11th International Workshop on*, 8, IEEE, Santiago de Compostela. 77

TORKAMANI-AZAR, F. & TAIT, K.E. (1996). Image recovery using the anisotropic diffusion equation. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, **5**, 1573-8. 83

TRAN, D., WAGNER, M., LAU, Y.W. & GEN, M. (2004). Fuzzy Methods for Voice-Based Person Authentication. *Transactions of the Institute of Electrical Engineers of Japan*, **124**, 1958-1963. 33

VENKATESWARLU, N.B. & BOYLE, R.D. (1995). New segmentation techniques for document image analysis. *Image and Vision Computing*, **13**, 573-583. 94, 111

Vincent, L. (1993). Grayscale area openings and closings: their applications and efficient implementation. In *Proc. EURASIP Workshop on Mathematical Morphology and its Applications to Signal Processing, Barcelona, Spain*, 22-27. 34

Vincent, L. & Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE transactions on pattern analysis and machine intelligence*, **13**, 583-598. 4

Voci, F., Eiho, S., Sugimoto, N. & Sekiguchi, H. (2004). Estimating the gradient threshold in the perona-malik equation. *IEEE Signal Processing Magazine*, **23**, 39-46. 85

Vogl, T., Mangis, J., Rigler, A. & Zink, W. (1988). Accelerating the convergence of the back-propagation method. *Biological Cybernetics*, **59**, 257-263. 54

Wang, X.H., Istepanian, R.S.H. & Hua, S.Y. (2003a). Application of wavelet modulus maxima in microarray spots recognition. *IEEE Transactions on Nanobioscience*, **2**, 190-192. 5, 35

Wang, X.H., Istepanian, R.S.H. & Song, Y.H. (2003b). Microarray image enhancement by denoising using stationary wavelet transform. *IEEE Transactions on Nanobioscience*, **2**, 184-189. 5

Watson, J.D. & Crick, F.H. (1953). A structure for deoxyribose nucleic acid. *Nature*, **171**, 737-738. 10, 13

Weickert, J. (1995). *Multiscale texture enhancement*, vol. 970, 230-237. Springer Verlag, Berlin / Heidelberg. 82

Weickert, J. (1996). Theoretical foundations of anisotropic diffusion in image processing. *Computing Supplementum*, **11**, 221-236. 82

Weickert, J. (1998). *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner, Stuttgart, Germany. 31, 83, 85, 92

Weickert, J. & Benhamouda, B. (1997). *A semidiscrete nonlinear scalespace theory and its relation to the PeronaMalik paradox*, 1-10. Springer Verlag GmbH, Vienna. 85

Weickert, J., Ishikawa, S. & Imiya, A. (1997). *On the history of Gaussian scale-space axiomatics*, 45-59. Kluwer Academic Publishers, Dordrecht, The Netherlands. 83

Weinstein, J., Myers, T., O'Connor, P., Friend, S., Fornace Jr, A., Kohn, K., Fojo, T., Bates, S., Rubinstein, L., Anderson, N. & Others (1997). An information-intensive approach to the molecular pharmacology of cancer. *Science*, **275**, 343. 19

Weng, G. (2009). cDNA Microarray Image Processing Using Morphological Operator and Edge-Enhancing Diffusion. In *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, 1-4, IEEE.

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. thesis, Harvard University, Cambridge, MA. 52

Wheeler, D.A., Srinivasan, M., Egholm, M., Shen, Y., Chen, L., McGuire, A. & al., E. (2008). The complete genome of an individual by massively parallel DNA sequencing. *Nature*, **452**, 872-6. 11

Whitaker, R. & Gerig, G. (1994). *Vector-valued di usion*, 93134. Kluwer Academic Publishers, Dordrecht, The Netherlands. 82

Whitaker, R.T. (1993). Geometry-limited diffusion in the characterization of geometric patches in images. *CVGIP: Image Understanding*, **57**, 111-120. 82

Wit, E. & McClure, J. (2004). *Statistics for Microarrays: Design, Analysis and Inference*. Wiley. 28

Witkin, A. (1983). Scale-Space Filtering. In *International Joint Conference Artificial Intelligence*, 1019-1021, Karlsruhe, West Germany. 82, 83, 84

Witkin, A. (1984). Scale-space filtering: A new approach to multi-scale description. In *IEEE International Conference on ICASSP '84. Acoustics, Speech, and Signal Processing*, vol. 9, 150-153. 34

Xiang, C. & Chen, Y. (2000). cDNA microarray technology and its applications. *Biotechnology Advances*, **18**, 35-46. 23

Yang, Y.H., Buckley, M.J. & Speed, T.P. (2001). Analysis of cDNA microarray images. *Briefings in Bioinformatics*, **2**, 341-349. 30, 32

Yang, Y.H., Buckley, M.J., Dudoit, S. & Speed, T.P. (2002). Comparison of Methods for Image Analysis on cDNA Microarray Data. *Journal of Computational & Graphical Statistics*, **11**, 108-136. 5, 21, 22, 23, 28, 30, 32, 110

Yaroslavsky, L. & Eden, M. (1996). *Fundamentals of digital optics*. Birkhauser Boston, Berlin, Germany. 81

Ye, P. & Weng, G. (2009). Microarray Image Segmentation Using Region Growing Algorithm and Mathematical Morphology. *2009 Fifth International Conference on Information Information Assurance and Security*, l373-376.

Yi, W., Liangpei, Z. & Pingxiang, L. (2005). Nonlinear multispectral anisotropic diffusion filters for remote sensed images based on MDL and morphology. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium, IGARSS '05*, 4327-4330. 85

You, Y.L., Xu, W., Tannenbaum, A. & Kaveh, M. (1996). Behavioral analysis of anisotropic diffusion in image processing. *IEEE transactions on image processing*, **5**, 1539-1553. 85, 86

Young, R. & Center, N. (2000). Biomedical Discovery Review with DNA Arrays. *Cell*, **102**, 9-15. 19

Zarandy, A., Werblin, F., Roska, T. & Chua, L.O. (1996). Spatial logic algorithms using basic morphological analogic CNN operations. *International Journal of Circuit Theory and Applications*, **24**, 283-300. 76

Zhang, F., Yoo, Y.M., Koh, L.M. & Kim, Y. (2007). Nonlinear diffusion in Laplacian pyramid domain for ultrasonic speckle reduction. *IEEE transactions on medical imaging*, **26**, 200-11. 85

Zhou, S., Kassauei, K., Cutler, D.J., Kennedy, G.C., Sidransky, D., Maitra, A. & Califano, J. (2006). An oligonucleotide microarray for high-throughput sequencing of the mitochondrial genome. *The Journal of molecular diagnostics : JMD*, **8**, 476-482. 22